KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

BÜTÜNSEL HAREKET KESTİRİMİNİN İKİLİ GÖRÜNTÜLER ÜZERİNDEN TEK KOMUT ÇOKLU VERİ ALTYAPISI İLE GERÇEKLENMESİ

BOĞAÇ TURGUT

KOCAELİ 2015

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

BÜTÜNSEL HAREKET KESTİRİMİNİN İKİLİ GÖRÜNTÜLER ÜZERİNDEN TEK KOMUT ÇOKLU VERİ ALTYAPISI İLE GERÇEKLENMESİ

BOĞAÇ TURGUT

Doç.Dr. Oğuzhan URHAN
Danışman, Kocaeli Üniv. .....

Prof.Dr. Cabir VURAL
Jüri Üyesi, Marmara Üniv. .....

Doç.Dr. Kemal GÜLLÜ
Jüri Üyesi, Kocaeli Üniv. .....

Tezin Savunulduğu Tarih: 13.04.2015

ÖNSÖZ VE TEŞEKKÜR

Tez çalışmamda rehberliğini, değerli zamanını ve sabrını esirgemeyen, bilgi ve önerilerini her zaman paylaşan değerli danışman hocam Doç. Dr. Oğuzhan URHAN'a, bugünlere gelmemde emeği geçmiş tüm değerli hocalarıma teşekkürü bir borç bilirim.

Ayrıca yüksek lisans eğitimim süresince manevi destekleri ve sevgileri ile yanımda olan aileme, her düştüğümde beni elimden tutup kaldıran, sevgisiyle bana güç veren, her daim evimizde huzuru bana yaşatan hayat arkadaşım, sevgili eşim Safiye TURGUT'a ve bu zor yolculuğumda yanımda olan bütün arkadaşlarıma teşekkürlerimi sunarım.

Nisan - 2015 Boğaç TURGUT

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

t_n : n anındaki zaman
x : t_1 zamanındaki 2-boyutlu konumu
x′ : t_2 zamanındaki 2-boyutlu konumu
d_x : Yatay eksendeki değişim
d_y : Düşey eksendeki değişim
$d(x;t_1,t_2)$: Hareket vektörü
p : Hareket parametreleri
$M \times M$: Çerçeve boyutu
$N \times N$: Blok boyutu
s : Arama pencere boyutu
M : İki argüman arasındaki hata fonksiyonu
B^t : İkili şimdiki çerçeve
B^{t-1} : İkili önceki çerçeve
Im : İmge boyutu
$MSD()$: Farkların karelerinin ortalaması
$MAD()$: Mutlak farkların ortalaması
$NNMP()$: Uyumsuz nokta sayısı
$\oplus$: XOR operatörü
K : 1-BT süzgeç çekirdeği
α : Piksellerin ikili doğal değerleri
g_k : k ’inci gri bit düzlemi
LBP : Yerel ikili örüntü operatörü
RMSE : Karesel ortalama hata
x_k : MSE uyumlama ölçütüne göre çıkarılmış yatay hareket vektörü
y_k : MSE uyumlama ölçütüne göre çıkarılmış düşey hareket vektörü
$\hat{x}_k$: Önerilen uyumlama ölçütüne göre çıkarılmış yatay hareket vektörü
$\hat{y}_k$: Önerilen uyumlama ölçütüne göre çıkarılmış düşey hareket vektörü

Kısaltmalar

1-BT : One-Bit Transformation (Bir-Bit Dönüşümü)
2D : Two Dimensional (2-Boyutlu)
2DLOG : 2 Dimensional Logarithmic Search (2-Boyutlu Logaritmik Arama)
3D : Three Dimensional (3-Boyutlu)
3SS : Three Step Search (Üç Adımda Arama)
4SS : Four Step Search (Dört Adımda Arama)
ARM : Acorn RISC Machine (RISC tabanlı Bir İşlemci Mimarisi)
CPU : Central Processing Unit (Merkezi İşlem Birimi)

GPU : Graphics Processing Unit (Grafik İşleme Birimi)
LBP : Local Binary Pattern (Yerel İkili Örüntü)
MAD : Mean Absolute Difference (Mutlak Farkların Ortalaması)
MF-1BT : Multiplication-Free One-Bit Transform (Çarpmasız Bir-Bit Dönüşümü)
MIMD : Multiple Instruction Multiple Data (Çoklu Komut Çoklu Veri)
MISD : Multiple Instruction Single Data (Çoklu Komut Tek Veri)
MSD : Mean Squared Difference (Farkların Karelerinin Ortalaması)
N3SS : NewThree Step Search (Yeni Üç Adımda Arama)
NNMP : Number of Non-Matching Point (Uyumsuz Nokta Sayısı)
RMSE : Root Mean Square Error (Karesel Ortalama Hata)
SIMD : Single Instruction Multiple Data (Tek Komut Çoklu Veri)
SISD : Single Instruction Single Data (Tek Komut Tek Veri)

BÜTÜNSEL HAREKET KESTİRİMİNİN İKİLİ GÖRÜNTÜLER ÜZERİNDEN TEK KOMUT ÇOKLU VERİ ALTYAPISI İLE GERÇEKLENMESİ

ÖZET

Sayısal görüntü stabilizasyon uygulamaları video kaydetme yeteneğine sahip cihazlarının küçülmesi ve dolayısı ile mobilleşmesi nedeniyle birçok uygulamada önemli bir ihtiyaç olarak öne çıkmaktadır. Görüntü stabilizasyonu işlemleri sonunda istenen kamera hareketini bozmadan harici etkenler sebebi ile oluşan istemsiz titreşim ve bozunumları ortadan kaldırmak amaçlanmaktadır. Dijital görüntü stabilizasyonunda 2-boyutlu bütünsel hareket kestirim yöntemleri görüntü stabilizasyonu aşamasında daha az işlem karmaşıklığına sahip olması sebebi ile 3-boyutlu yöntemlere göre mobil işlemciler için daha uygundur.

Bu tez kapsamında, literatürde önerilen yerel ikili örüntü (LBP), MF-1BT ve Interlaced Gray-coded bit-plane tabanlı ikili bütünsel hareket kestirim yöntemlerinin SIMD (Single Instruction Multiple Data) mimarisinin mantıksal operatörlerdeki paralelleştirme avantajını kullanarak gerçeklenmesi ele alınmıştır. Yapılan deneyler önerilen gerçekleme yaklaşımının kullanılması durumunda ARM Cortex A8 tabanlı platformda NEON SIMD motorunun kullanılarak gerçek zamanlılık hedefinin üstüne çıkılabileceğini göstermiştir. SIMD motorunun kullanımı ile işlemci hesap gücü görece düşük olduğu mobil cihazlarda bile işlem karmaşıklığı yüksek olan tam arama temelli bir düşük bit derinlikli hareket kestirim algoritmasının başarılı bir şekilde saniyede 40-50 çerçeve çalıştırılabileceği gösterilmiştir.

Anahtar Kelimeler: Görüntü Stabilizasyonu, NEON, Tek Komut Çoklu Veri (SIMD), Yerel İkili Örüntü.

IMPLEMENTATION OF GLOBAL MOTION ESTIMATION OVER BINARY IMAGES WITH SIMD ARCHITECTURE

ABSTRACT

Image stabilization applications are getting a more important necessity for many applications with the minimization and mobilization of devices having video recording capability. Image stabilization aims to reduce undesired movement or vibration without corrupting the desired camera motion. In digital image stabilization, 2-D global motion estimation methods are suitable for mobile processors because of less computational complexity compared to 3-D methods.

In this thesis, several binary global motion estimation methods presented in literature such as Local Binary Pattern (LBP), MF1-BT and Interlaced Gray-coded bit-plane are implemented by making use of data parallelism of SIMD architecture. Experimental results shows that the proposed approach and NEON SIMD engine implementation can reach beyond the real time requirement on ARM Cortex A8 based embedded platform. It is shown that with the usage of SIMD engine, it becomes possible to execute low-bit depth based global motion estimation algorithm with full-search approach at 40-50 frame per second even in the mobile devices having relatively lower computational power.

Key Words: Image Stabilization, NEON, SIMD (Single Instruction Multiple Data), Local Binary Pattern.

GİRİŞ

Günümüzde görüntü kayıt cihazları gelişen teknoloji ve eğilimler sonucu küçülmüş ve dolayısı ile taşınabilir hale gelmiştir. Mobil uygulamaların bir sonucu olarak görüntü kayıt işlemi sırasında istenmeyen kamera hareketleri sebebi ile kayıt edilen görüntüde istenmeyen titreşimler meydana gelmektedir. İstenen kamera hareketlerini bozmadan istemsiz kamera hareketlerinin düzeltilmesi görüntü stabilizasyonu olarak adlandırılmaktadır. Bu işlem görüntü kaydı yapabilen mobil cihazlar için temel bir gereksinim haline gelmiştir.

Görüntü işleme temelli dijital görüntü stabilizasyonu genel olarak iki temel adımdan oluşmaktadır. Birinci adım hareket kestirimi, ikinci adım ise hareket düzeltimdir. Hareket kestirimi ardışıl çerçeveler arasında bütünselin hesaplanmasıdır. Bu aşamaların çıktıları hareket düzeltimi aşamasına girdi olarak kullanılmaktadır. Hareket stabilizasyonun işlem karmaşıklı en yüksek olan kısmı hareket kestirimi aşamasıdır. Hareket kestirimin hızlı şekilde yapılması stabilizasyon hızını arttırmaktadır. Bütünsel hareket kestirimi genel olarak çerçevenin bütününe etki eden hareket vektörlerinin kestirilmesi olarak tanımlanabilir. Bütünsel hareket kestirimi görüntü stabilizasyonunun yanı sıra video sıkıştırma ve sahne sınıflandırma gibi uygulamalarda da kullanılmaktadır [1]. Bunun yanı sıra yerel hareket kestirimi öncesi bütünsel hareket vektörlerinin bilinmesi, yerel nesne hareketlerinin üzerinde kamera hareketlerinin yarattığı bozunumu gidererek yerel hareket kestirim başarımını arttırmaktadır. Dolayısı ile bütünsel hareket kestirimi çıktıları kullanılarak yerel hareket kestirimi başarımının artırılması mümkündür. Bütünsel hareket kestirimi ilgili hareket modelinin farklılığına göre 2-boyutlu ve 3-boyutlu olarak gerçekleştirilebilir. 2-boyutlu hareket kestiriminin çıktılarını birçok 3-boyutlu hareket kestirimi yöntemi algoritmalarına girdi olarak kullanmaktadır. Ayrıca 2-boyutlu hareket kestiriminin başlı başına yeterli olduğu geniş bir uygulama alanının olması sebebi ile bu konu birçok araştırmada temel çalışma konusu olarak ele alınmıştır.

2-boyutlu bütünsel hareket kestiriminin yaygın olarak ihtiyaç duyulması ve işlemsel karmaşıklığının oldukça fazla olması sebebi ile literatürde bütünsel hareket kestirimini hızlandırmak için birçok farklı çalışma bulunmaktadır. İmgelerin ikili dönüşümü üzerinden uyumlama aşamasında ikili operatörlerin kullanıldığı yöntemler donanımsal tabanlı hızlandırma uygulamalarının önünü açmaktadır. Günümüzde artık mobil işlemciler üzerinde bile tek komut çoklu veri (Single Instruction Multiple Data-SIMD) motorları bulundurmaktadır. SIMD motorları veri miktarının fazla ve tekrarlı aynı komutların işletildiği uygulamalarda (video veya ses kodlama, sıkıştırma) başarı ile kullanılmaktadır. Bu motorların kullanımı ile sadece yazılımsal iyileştirmeler gerçekleştirilerek uygulamaların hızını arttırmak mümkündür. Bilindiği kadarı ile ikili bütünsel hareket kestiriminin donanımsal uygulamaların yanında yazılımsal olarak hızlandırma üzerine belirgin çalışmalar literatürde bulunmamaktadır. Bu tez kapsamında ikili bütünsel hareket kestirim algoritmalarının SIMD motorları ile gerçeklenebilmesine yönelik ilk çalışmalar yapılmıştır. Önerilen yaklaşım ARM tabanlı bir işlemci üzerinde SIMD motorları kullanılarak gerçeklenmiştir.

Tez kapsamında ilk olarak, ikili bütünsel hareket kestirimi için literatürdeki çalışmalar incelenmiştir. Bu kapsamda bütünsel hareket tanımı, uyumlama ölçütleri, ikili dönüşüm yöntemleri anlatılacaktır.

Çalışmanın ikinci kısmında, farklı kaynaklarda önerilen ikili bütünsel hareket kestirim algoritmalarının SIMD adımlarının nasıl gerçeklendiği anlatılacaktır. Bu aşamada ARM işlemci komutları kullanılacak ve veri uyumlaması sonrası gerekli komut setleri gösterilecektir.

Çalışmanın son bölümünde, ARM işlemci üzeride SIMD ile gerçeklenen farklı ikili bütünsel hareket kestirimi yöntemlerinin standart ve SIMD sonrası performansları değerlendirilerek sonuçlar ve yapılan öneriler anlatılacaktır.

1. İKİLİ GÖRÜNTÜLER ÜZERİNDE BÜTÜNSEL HAREKET KESTİRİMİ

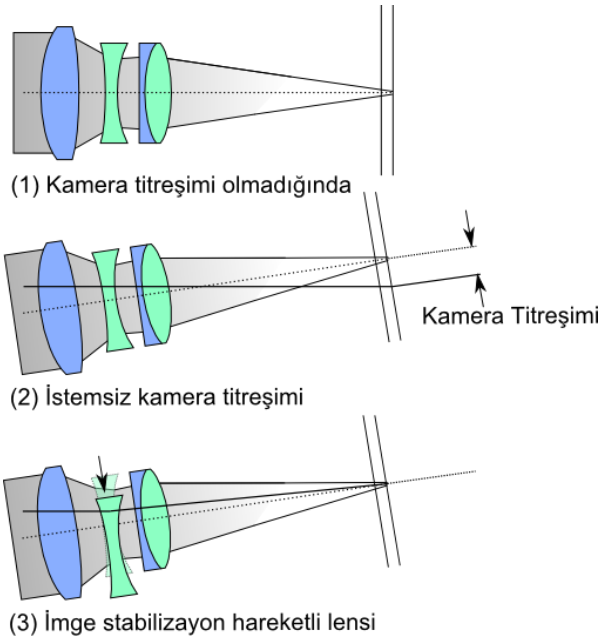
1.1. Giriş

Hareket bilgisi video işlemenin farklı alanlarında kullanılmak için önemli bir girdi oluşturmaktadır. Video sıkıştırma, onarım ve hareket stabilizasyonu gibi video işleme uygulamaları hareket bilgisinin kullanıldığı örnekler olarak verilebilir. Video sıkıştırmada ardışık çerçevelerdeki piksellerin hareketlerinin bilinmesi sıkıştırma algoritmasında çerçeveler için kullanılacak/aktarılacak bit oranını önemli bir ölçüde azaltmaktadır [2]. Hareket bilgisinin önemini gösteren son kullanıcıya yönelik diğer bir uygulama ise video stabilizasyonudur. Çerçeveler arasındaki hareket bilgisinin analizi sonucu istenmeyen titreşimleri yok edilerek videonun stabilizasyonu sağlanmaktadır. Kullanıcıya yönelik bu gibi bütün uygulamalar için hareket kestirimi girdi sağlamaktadır. Hareket kestirimi ile ilgili temel bilgilere [3-4]'ten ulaşılabilir. Genel olarak iki imge arasındaki gözlemlenen hareketi sahne içindeki hareket eden objeler yani yerel hareketler ve bütün sahne içindeki kameranın bütünsel hareketi olarak ikiye ayırabiliriz. Kısacası sahne içinde hareket eden objelerin bir önceki ve sonraki çerçevedeki konumlarının farklılığını kestirmeye yerel hareket kestirimi, kameranın hareketi dolayısı ile sahne değişimlerinin kestirimi ise bütünsel hareket kestirimi olarak tanımlanmaktadır.

[5]'deki çalışmada bütünsel hareket stabilizasyonu yöntemleri mekanik, optik ve dijital olmak üzere üç temel başlık altında sınıflandırılmıştır. Bu üç yöntemde istenmeyen kamera hareketlerinden oluşan titreşimi azaltmayı önermektedir. Mekanik stabilizasyonda direk olarak kamera açısal hız ölçme sensörleri ve motorun bağlı oluğu mekanik bir platforma sabitlenir. Titreşimden kaynaklı ölçülen açısal değişimlerin zıt yönünde motorlar hareket ettirilerek, titreşimin kameranın kayıt ettiği videoya yansımaması sağlanır. Şekil 1.1'de jiroskopik bir sabitleyici örneği görünmektedir [6].

Şekil 1.1. Mekanik jiroskopik sabitleyici

Optik sabitleyiciler görüntünün imge sensörü tarafından yakalanmadan önce optik olarak ışık demetini yönlendirme esasına dayalı çalışır. Görüntüleme cihazı lens takımının optik eksinin içine yerleştirilen, yatay ve düşey eksende kamera hareketin tersi yönünde hareket edebilen imge sabitleme lensi ışığın imge sensörünün üzerine bozunumsuz düşmesini sağlar. Şekil 1.2'de görüldüğü gibi imge sabitleme lensi istemsiz kamera titreşimi sonucu ters yönde hareket ederek ışık kümesinin sensöre bozunumsuz olarak ulaşmasını sağlamaktadır. Günümüzde farklı şirketlerin pazarda kullandığı birçok imge sabitleyici vardır. İmge stabilizasyonu ile ilgili açıklamalar için [7]'deki sunulan teknoloji özeti incelenebilir.



Şekil 1.2. Optik imge sabitleyici

Üçüncü yöntem ise sayısal stabilizasyon olarak tanımlanmaktadır. Dijital stabilizasyon genel olarak hareket kestirimi ve hareket düzeltimi adımlarından oluşmaktadır. İşlem karmaşıklığı en fazla olan kısım ise hareket kestirimidir. Hareket kestirimi konusunda [5]'deki literatür özeti incelenebilir. Kamera hareketlerinin incelendiği bütünsel hareket kestirimi konusunda farklı birçok çalışma yapılmıştır. İkili bütünsel hareket kestiriminin detaylarına geçilmeden önce bütünsel hareket kestirimi ile ilgili genel bir giriş ve literatür özeti verilecektir.

1.2. Bütünsel Hareket

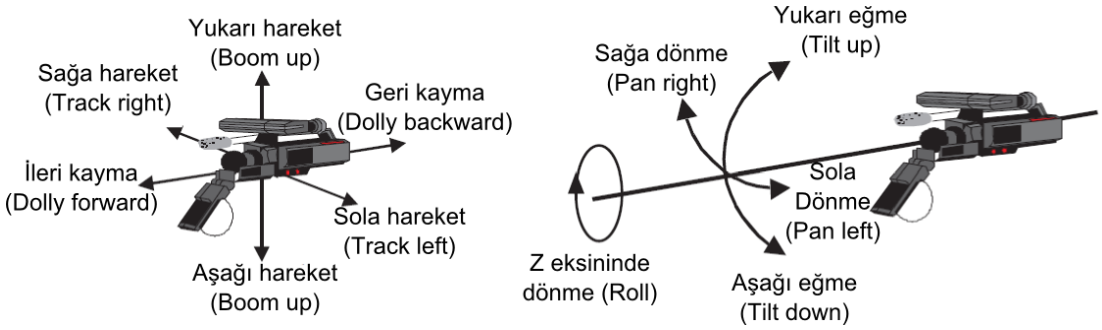
Bütünsel hareket bir imge dizisindeki parametrik olarak modellenebilen kamera kaynaklı hareket olarak tanımlanabilir [1]. Bu parametrelerin kestirimine ise bütünsel hareket kestirimi denmektedir. Bütünsel hareket kestiriminde hareket parametre seti blok pikseller veya her bir piksel değişimi yerine bütün çerçevedeki hareketi tanımlamaktadır. Bütünsel kamera hareketleri ayrıca yerel hareketlere baskın şekilde etkili olduğu için birçok stabilizasyon uygulamasında kullanılmaktadır. Bütünsel hareketin yönünü göstermek için bütünsel hareket vektörleri kullanılmaktadır. Bütünsel hareket vektörü kestirilen bütünsel hareket parametrelerinden hesaplanan hareket vektörü olarak ifade edilir [1]. Şimdiki çerçevedeki pikseli (x_t, y_t) ile sonraki çerçevedeki aynı pikselin bütünsel hareket parametrelinin kestirilmesi ile hesaplanan konumu ise (x'_t, y'_t) olarak gösterilsin. Bu durumda bütünsel hareket vektörü $(\hat{x}_t, \hat{y}_t)$ Eşitlik (1.1)'deki gibi gösterilebilir [1].

$$\begin{cases} \hat{x}_t = x'_t - x_t \\ \hat{y}_t = y'_t - y_t \end{cases} \tag{1.1}$$

Bütünsel hareket vektörleri değerlendirilerek bütünsel hareket alanı belirlenebilir.

1.2.1. Mobil cihazlarda bütünsel hareket

Şekil 1.3'de genel kamera 3 boyutlu düzlemde yapabileceği istemli veya istemsiz hareketleri görünmektedir [8].



Şekil 1.3. Genel kamera hareketleri

Mobil bir kameranın kullanıcı kaynaklı bütünsel hareketleri için aşağıdaki tanımlar [5]'den alınmıştır;

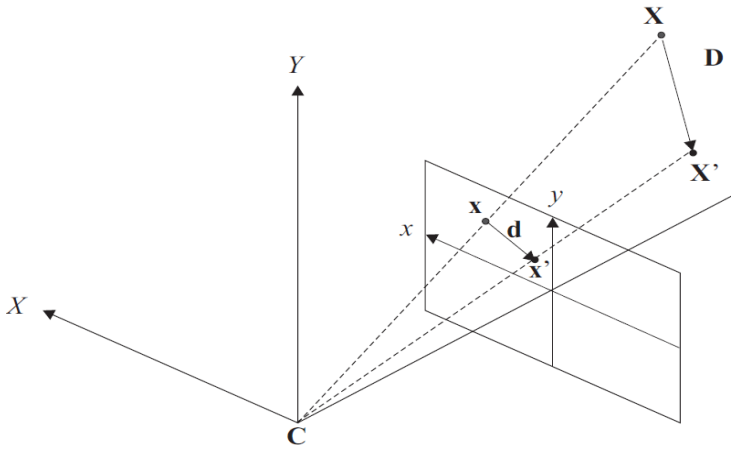
- Track: X yönündeki yatay hareket.
- Boom: Y yönündeki dikey hareket.
- Dolly: Kameranın optik ekseni yönündeki hareket.
- Pan: Kameranın dikey ekseni etrafında dönme hareketi.
- Tilt: Kameranın düşey ekseni etrafında dönme hareketi.
- Roll: : Kameranın optik ekseni etrafında dönme hareketi.
- Zoom: Kameranın odak uzaklığının değişmesi sonucu oluşan hareket.

Bu kamera hareketleri istenmeyen ve istemli kamera hareketleri olarak ikiye ayrılmıştır [1]. Track, Boom, Tilt ve Pan istenmeyen kamera titreşimlerinden en sıklıkla karşılaşılan kamera hareketleridir. Bu tez çalışmasında 2-boyutlu yaklaşımlar kullanılarak yatay ve düşey (öteleme "translation": "Track" ve "Boom") yöndeki bütünsel hareketlerin kestirimi üzerinde durulacaktır.

1.2.2. Bütünsel hareket modeli

Hareket kestirimi yaklaşımları kullanılan hareket modeline göre 2-boyutlu (2D) ve 3-boyutlu (3D) olmak üzere iki genel başlık altında toplanabilir. 3-boyutlu hareket kestirimi nesnenin veya kameranın 3-boyutlu uzayda hareketinin tanımlanmasına olanak verir. Özellikle robotik ve araç takip sistemleri gibi bilgisayarlı imge işleme sistemlerinde kullanılmaktadır [8]. Diğer uygulama alanlarından birisi ise obje tabanlı video sıkıştırma uygulamalarıdır [9]. 2-boyutlu hareket kestirimi ise kamera veya nesne hareketlerinin 2-boyutlu bir hareket alanına yansımasını ele almaktadır. [10]'da hareket vektörü alanı 3-boyutlu hareketi kestirmek için kullanılmaktadır. 2-boyutlu hareket kestirimi daha önce bahsedildiği gibi birçok uygulama alanın olması

yanı sıra 3-boyutlu hareket kestirimi için ön adım olarak 3-boyutlu kestirim algoritmalarına girdi oluşturmaktadır. Bir obje t_1 anındaki $\boldsymbol{X} = [X, Y, Z]^T$ konumundan $t_2 = t_1 + d_t$ anında ki $\boldsymbol{X}' = [X', Y', Z']^T = [X + D_X, Y + D_Y, Z + D_Z]$ konumuna hareket ettiğinde, imgenin 2-boyutlu yansıması $x = [x, y]^T$ konumdan $x = [x', y']^T = [x + d_x, y + d_y]^T$ konumuna değişir. 3-boyutlu yer değişimi yani 3-boyutlu hareket vektörü $\boldsymbol{D}(\boldsymbol{X}; t_1, t_2) = \boldsymbol{X}' - \boldsymbol{X} = [D_X, D_Y, D_Z]^T$ şeklinde tanımlanır. 2-boyutlu yer değişimi yani 2-boyutlu hareket vektörü ise $\boldsymbol{d}(\boldsymbol{x}; t_1, t_2) = \boldsymbol{x}' - \boldsymbol{x} = [d_x, d_y]^T$ şeklinde tanımlanır [8]. Şekil 1.4'de hareket eden bir objenin imge düzlemine yansımasının 2-boyutlu ve 3-boyutlu hareket ilişkisi gösterilmiştir. 2-boyutlu ve 3-boyutlu hareket ilişkisin detayları için [8] incelenebilir.



Şekil 1.4. 2-boyutlu ve 3-boyutlu hareket vektör ilişkisi

Kamera hareketleri bütün pikseller üzerinde değişime sebep olmakta ve bütünsel hareket modeli ile tanımlanmaktadır. Bütünsel hareket yukarıda anlatıldığı gibi 2-boyutlu ve 3-boyutlu olarak tanımlanabilmektedir. 2-boyutlu bütünsel hareketlerin kestirimi hareket alanın hangi parametreler ile gösterilmesi problemini içermektedir. Her uygulama ve farklı hareket modellerine ihtiyaç duyar. Örnek olarak futbol yayınlarında sıklıkla yakınlaştırma ve uzaklaştırma yapılır iken araç kamerasında sabit sahne üzerindeki yatay ve düşey eksendeki titreşimlerin giderilmesi gerekmektedir [11]. Genel olarak sahnede birden fazla hareket eden obje olsa bile dominant bütünsel hareketlerin kestirimi büyük öneme sahiptir. İki imge arasındaki bağımlı bütünsel hareket farklı parametrelere sahip (Translation), (Similarity/Scaled

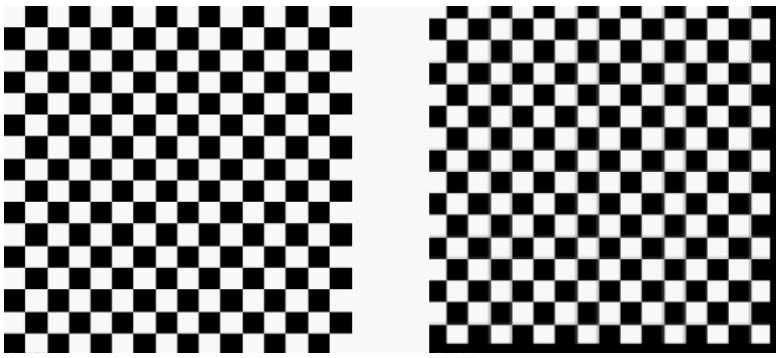
Rotation), (Affine) ve (Projective) gibi hareket modelleri ile tanımlanabilir [5]. Bütünsel hareket parametreleri Eşitlik (1.2) ile gösterilebilir [12]. Farklı parametre öbekleri farklı tipte hareketi göstermektedir.

$$\begin{bmatrix} x \\ y \end{bmatrix}_t = p\left\{ \begin{bmatrix} x \\ y \end{bmatrix}_{t-1}, \text{Parametreler} \right\} \tag{1.2}$$

Bütünsel hareket modeli olarak öteleme (translation) seçildiğinde hareket parametreleri yatay ve düşey eksenlerdeki tanımlayacak şekilde iki adettir. Eşitlik (1.3) ile ilgili bütünsel hareket parametreleri gösterilebilir.

$$\begin{bmatrix} x \\ y \end{bmatrix}_t = \begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + \begin{bmatrix} a \\ b \end{bmatrix} \tag{1.3}$$

Bu parametre öbeği ile imge arasındaki hareket (a=2,1, b=4,7) Şekil 1.5'de gösterilmektedir [12]. Tez kapsamında 2-boyutlu bütünsel hareket kestirimi dönüşümü üzerinden öteleme parametrelerinin kestirilmesi ele alınacaktır.

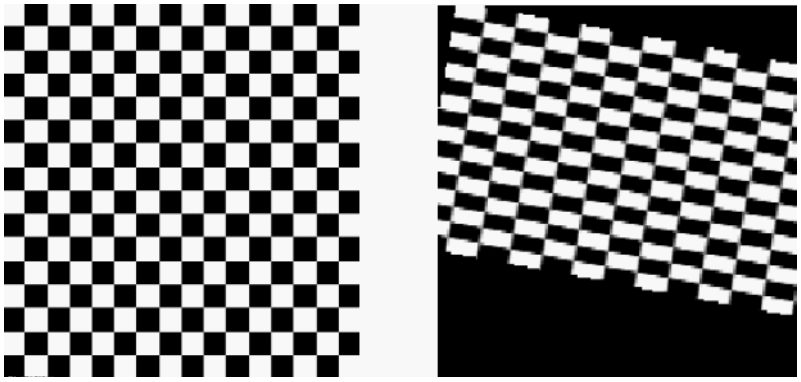


Şekil 1.5. Öteleme (translation) hareket modeli

Bütünsel hareket modeli olarak "Similarity / Scaled Rotation" seçildiğinde hareket parametreleri 5 adettir. Eşitlik (1.4) ile ilgili bütünsel hareket parametreleri gösterilebilir.

$$\begin{bmatrix} x \\ y \end{bmatrix}_t = \begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + \begin{bmatrix} a \\ b \end{bmatrix} \tag{1.4}$$

Bu parametre öbeği ile imge arasındaki hareket (a=5, b=5, c=0,7, d=1,5, $\theta = 10$) Şekil 1.6'de gösterilmektedir [12].

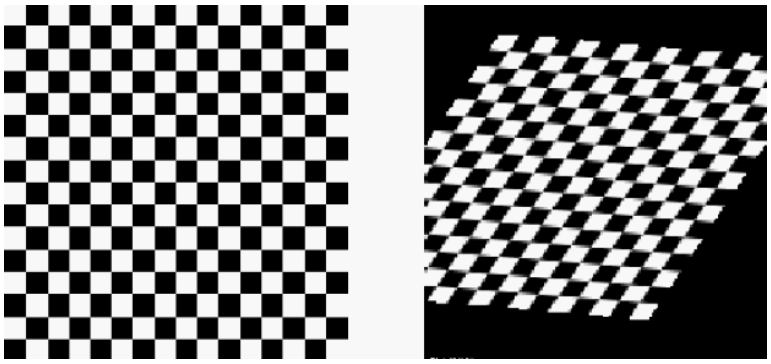


Şekil 1.6. (Similarity / scaled rotation) hareket modeli

Bütünsel hareket modeli olarak ilgin model (affine) seçildiğinde hareket parametreleri 6 adettir. Eşitlik (1.5) ile ilgili bütünsel hareket parametreleri gösterilebilir.

$$\begin{bmatrix} x \\ y \end{bmatrix}_t = \begin{bmatrix} c & e \\ f & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + \begin{bmatrix} a \\ b \end{bmatrix} \tag{1.5}$$

Bu parametre öbeği ile imge arasındaki hareket (a=-45, b=15, c=1, d=1,3, e=0,7,f=-0,1) Şekil 1.7'de gösterilmektedir [12]. Bu model ile paralel çizgiler paralel olarak korunmaktadır. Bu modelin kullanılarak bütünsel hareketlerin kestirildiği [13-14] örnek olarak incelenebilir.

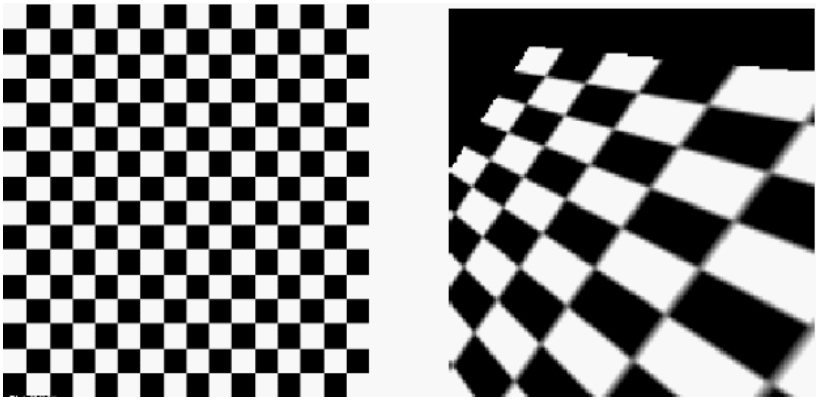


Şekil 1.7. (Affine) hareket modeli

Bütünsel hareket modeli olarak "projective" seçildiğinde hareket parametreleri 8 adettir. Eşitlik (1.6) ile ilgili bütünsel hareket parametreleri gösterilebilir.

$$\begin{bmatrix} x \\ y \end{bmatrix}_t = \frac{\begin{bmatrix} c & e \\ f & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + \begin{bmatrix} a \\ b \end{bmatrix}}{\begin{bmatrix} g & h \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + 1} \tag{1.6}$$

Bu parametre öbeği ile imge arasındaki hareket (a=-45, b=15, c=1, d=1,3, e=0,7, f=-0,1, g=0,01, h=0,0062) Şekil 1.8'de gösterilmektedir [12]. Bu modelin kullanımına örnek olarak [15] incelenebilir.



Şekil 1.8. (Projective) hareket modeli

1.3. Bütünsel Hareket Kestirimi

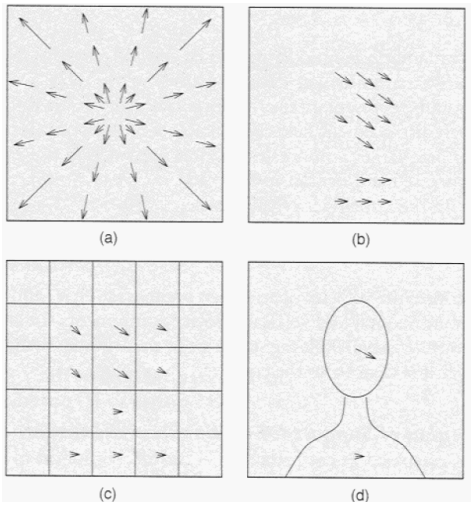
Bütünsel hareket kestirimi belirlenen hareket modeli üzerindeki parametrelerin kestirimi olarak tanımlanabilir. Bütünsel hareket kestirim yöntemlerini doğrudan (direct) ve dolaylı (indirect) olarak iki ana başlık altında toplanır. Doğrudan yöntemde bütünsel hareket parametreleri kestirimi verilen hareket parametre öbeğindeki kestirim hatasını azaltarak yapılmaktadır. Dolaylı yöntemde ise piksel veya blok bazındaki hareket yöntemleri bulunarak kestirilen bütünsel hareket alanına en iyi uyumlanan bütünsel hareket modeli bulunur [8]. Doğrudan kestirim yöntemleri işlemsel karmaşıklığın tekrarlı işlemlerin fazlalığı sebebi ile dolaylı yöntemler ile karşılaştırıldığında daha fazladır. Ayrıca doğrudan yöntemlerdeki başarım oranı daha düşüktür. Bu sebep ile dolaylı kestirim yöntemleri farklı popüler kamera kestirim yöntemlerinde kullanılmaktadır [16].

Doğrudan bütünsel hareket kestirim yönteminde iki ardışık çerçevedeki pikseller üzerindeki bütünsel hareket parametreleri eğim (gradient) arama veya diğer tekrarlı

yöntemlerle kestirilmektedir [16-19]. Doğrudan yöntemin detaylı örnekleri için [17-19] ve [8] incelenebilir.

Dolaylı bütünsel hareket kestirim yönteminde ilk olarak blok veya tüm pikseller için hareket vektörü çıkartılır. Bu yöntem yerel hareketlere karşı duyarlı olması sebebi ile gerçek bütünsel hareket vektörünü tanımlamada hataya sebep olabilir. Bu nedenle ikinci aşamada yerel hareketlerin sebep olduğu aykırı hareket vektörlerinin bütünsel hareket vektörleri üzerindeki bozunumu tekrarlı yöntemler ile en aza indirilmeye çalışılır. Eğer yerel hareket aykırı (outlier) vektörler olarak tanımlanabilir ise bütünsel hareket başarımı önemli ölçüde artacaktır. Aykırı hareket vektörlerinin elenmesi konusunda literatür özeti ve değişken ağırlıklı medyan filtre tabanlı aykırı (outlier) hareket vektörlerinin elenmesi temelli bir yöntem [20]'de sunulmuştur.

Dolaylı bütünsel hareket kestiriminde hareket vektörleri çıkartılırken hareket alanı önemli bir kavramdır. Bütünsel, piksel, blok veya obje/bölge tabanlı hareket alanları Şekil 1.9'da gösterilmektedir [9]. Piksel veya blok hareketleri üzerinden bütünsel hareket parametrelerinin kestirimi yerel hareketlere karşı duyarlı olmasına rağmen bütünsel hareketin dominant yapısı sonucu etkin şekilde kullanılmaktadır.



Şekil 1.9. Hareket alanı a) bütünsel, b) piksel, c) blok, d) nesne

Bütünsel hareket kestiriminde hareket vektörlerinin çıkartılması için kullanılan temel yöntemler optik akış, piksel özyineleme ve blok uyumlandırma olarak üç başlık altında ele alınabilir [21].

Optik akış görüntü dizisindeki her piksel için ayrı bir hareket vektörü hesaplanmasıdır. İlgili piksel için yer değiştirme hesaplanırken önceki çerçevedeki yakın komşulukları ile karşılaştırılarak kendisine en yakın değer sahip piksel bulunur. İki piksel arasındaki değişim hareket vektörünü vermektedir. Bütün pikseller için yapılan arama dolaylı yöntemler içinde kestirim başarımının yüksek olmasının yanı sıra işlemsel karmaşıklık açısından oldukça fazladır [21].

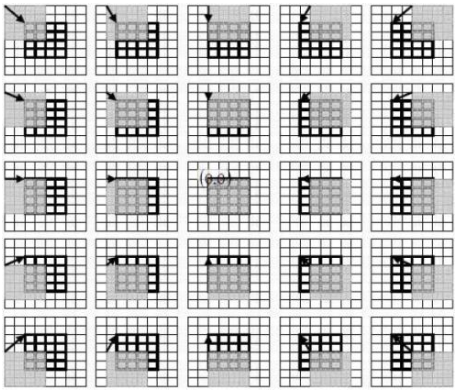
Piksel özyineleme yönteminde her piksel için hareket vektörü, ardışıl çerçevelerde belirli bir bölgedeki farklılıkların lineer olmayan fonksiyonu üzerinden özyinelemeli olarak kestirilir. Bölge bir grup piksel olabileceği gibi sadece tek bir pikselde olabilir. Piksel özyinelemeli yöntemler yalın olmalarına rağmen kestirim hataları büyük ve başarımları düşüktür [21].

Blok uyumlandırma yöntemi önceki ve şimdiki çerçeveler birbiri ile örtüşmeyen bloklara ayrılarak, her bir blok için arama bölgesi içerisinde blok uyulmama ölçütüne göre aranması ile hareket vektörlerinin çıkarılmasıdır. Blok uyumlandırma yönetimi en yaygın MPEG-1, MPEG-2, H.264/AVC ve HEVC gibi video kodlayıcılarda hareket kestirimi aşamasında kullanılmaktadır.

Blok bazlı hareket kestirimi yöntemleri arasında en iyi blok uyulmama başarımını veren yöntem tam arama yöntemidir. Bu yöntemde seçilen blok arama penceresi içinde tüm piksellerin aynı yönde yer değişimine uğrağını varsayılmakta ve tüm pikseller için uyulmama ölçütüne göre blok için tek bir vektör hesaplanmaktadır. Bu nedenle bu yöntemin işlem karmaşıklığı yüksek fakat donanım tabanlı uygulamalar için oldukça uygun bir yapıya sahiptir.

Aranacak blok boyutu $N \times N$ piksel ve arama penceresi için yatay ve düşey eksende $\mp s$ piksel olsun. Adım aralığı bir piksel olarak seçilir ise her blok $(2s+1)^2$ adet uyumlana bileceği blok ile karşılaştırılmaktadır. Her bir blok kaydırma işleminde N^2 adet piksel için uyulmama ölçütü hesabı yapılır. Bu durumda çerçeve içindeki her bir bloğa hareket vektörü atanmadan önce $(2s+1)^2 N^2$ işlem yapılması gerekmektedir. Eğer imge boyutu MxM olarak alınır ise tam arama için gereken blok sayısı $(M/N)^2$ olacaktır. Böylece çerçevedeki tüm bloklar için hareket vektörü

hesaplanması için gereken işlem sayısı N blok boyutundan bağımsız olarak $(2s+1)^2 M^2$ şeklinde hesaplanır [8]. Şekil 1.10'da tam arama blok uyumlandırma yöntemi için hareket vektörlerinin atanması işlemi gösterilmiştir [22].



Şekil 1.10. Tam arama blok uyumlama yöntemi

Literatürde tam arama yönteminin işlem yükünün fazla olması sebebi ile hızlı hareket kestirimi algoritmaları önerilmiştir. Hızlı arama yöntemlerinin arama penceresindeki aday nokta sayısını azaltmaya yönelik arama örüntüsüne sahiptirler. Bu neden ile tam aramaya göre başlarım oranı daha düşüktür. Bu yöntemlere başlıca üç adımda arama (3SS) [23], 2-boyutlu logaritmik arama (2DLOG) [24], yeni üç adımlı arama (N3SS) [25] ve dört adımda aramadır (4SS) [26].

Genellikle yerel hareketin kestirimi için kullanılan bu yaklaşımların blok boyutunun aranacak çerçevenin önemli bir bölümünü kapsayacak şekilde ele alınması ile bütünsel hareketi kestirmek için de kullanılması mümkündür. Bu yaklaşımı kullanan yöntemler blok uyumlama temelli bütünsel hareket kestirimi yöntemleri olarak isimlendirilmektedir.

Blok uyumlama temelli hareket kestiriminin hesapsal karmaşıklığın azaltılması için bir diğer yöntem ise imgelerin düşük bit çözünürlüğünde ifade edilmesidir. Böylece düşük işlem yüküne sahip blok uyulmama ölçütü kullanımı ile yapılacak hesaplamaların daha hızlı yapılabilmesi amaçlanmıştır. Bölüm 1.4'te ikili bütünsel hareket kestirimi başlığında düşük bit çözünürlüğü kullanan yöntemler incelenecektir. Düşük bit çözünürlüğünün kullanan hareket kestirim yöntemleri etkin donanımsal gerçeklemelerin (ASIC ve FGPA gibi) yanı sıra grafik işleme

ünitesi (GPU) veya SIMD motorları gibi hızlandırıcı yongaların etkin kullanımına uygundur.

1.4. Blok Uyumlama Ölçütleri

Blok uyumlama temelli bütünsel hareket kestirim yönteminde farklı blok uyumlama ölçütleri kullanılmaktadır. Blok uyumlamalarda maksimum benzerliği bulmak yerine, uyumlama hatasını bulmak tercih edilir. Genel olarak t_n ve t_{n-1} anındaki iki çerçeve arasındaki hata Eşitlik (1.7)'deki gibi yazılabilir [27]. Blok uyumlama ölçütleri [27] üzerinden anlatılacaktır. Bu konudaki daha detaylı açıklamalar için [27] ve [28] incelenebilir.

$$D(m,n)=\frac{1}{Im}\sum_{i=0}^{p}\sum_{j=0}^{q}M\left(B^{t}(i,j),B^{t-1}(i+m,j+n)\right) \tag{1.7}$$

Im: İmge boyutu ($Im=p\times q$), M fonksiyonu iki giriş imgesi arasındaki hatayı belirten metrik, $D(m,n)$ uyumlama ölçütü, B^{t} şimdiki çerçeve, B^{t-1} ise bir önceki çerçeveyi göstermektedir.

1.4.1. Farkların karelerinin ortalaması (mean squared difference, MSD)

MSD başarımı en yüksek blok uyumlama ölçütlerinden biridir. Ayrıca (mean squared error MSE) olarak da adlandırılır. Aşağıdaki şekilde tanımlanır.

$$MSD(m,n)=\frac{1}{pq}\sum_{i=-p/2}^{p/2}\sum_{j=-q/2}^{q/2}\left(B^{t}(i,j)-B^{t-1}(i+m,j+n)\right)^{2} \tag{1.8}$$

Burada (m,n) değeri arama penceresinin tanımına göre değişmektedir. Eğer arama penceresi s sınırında kaydırılır ise $-s\le m,n\le s$ ile gösterilir.

1.4.2. Mutlak farkların ortalaması (mean absolute difference, MAD)

MAD, mutlak hataların ortalaması (mean absolute error, MAE) olarak da adlandırılır. Aşağıdaki şekilde tanımlanır. Bu metriğin MSD'den farklı kare alma işlemi gerektirmemesi nedeniyle göreceli olarak daha düşük yüküne sahip olmasıdır.

$$MAD(m,n)=\frac{1}{pq}\sum_{i=-p/2}^{p/2}\sum_{j=-q/2}^{q/2}\left|B^{t}(i,j)-B^{t-1}(i+m,j+n)\right| \quad (1.9)$$

Eğer arama penceresi s sınırında kaydırılır ise $-s \le m,n \le s$ ile gösterilir.

1.4.3. Uyumsuz nokta sayısı (Number of non-matching point, NNMP)

NNMP, bit çözünürlüğü düşürülmüş imgelerde özel-VEYA (Exclusive-OR) temelli bir hesaplama yapısı kullanarak uyumlamanın hızlı hesaplanmasını sağlamak üzere önerilmiş bir metriktir. Ayrıca blok uyumlamanın donanımsal olarak gerçeklenmesini kolaylaştırmaktadır. Aşağıdaki şekilde tanımlanır.

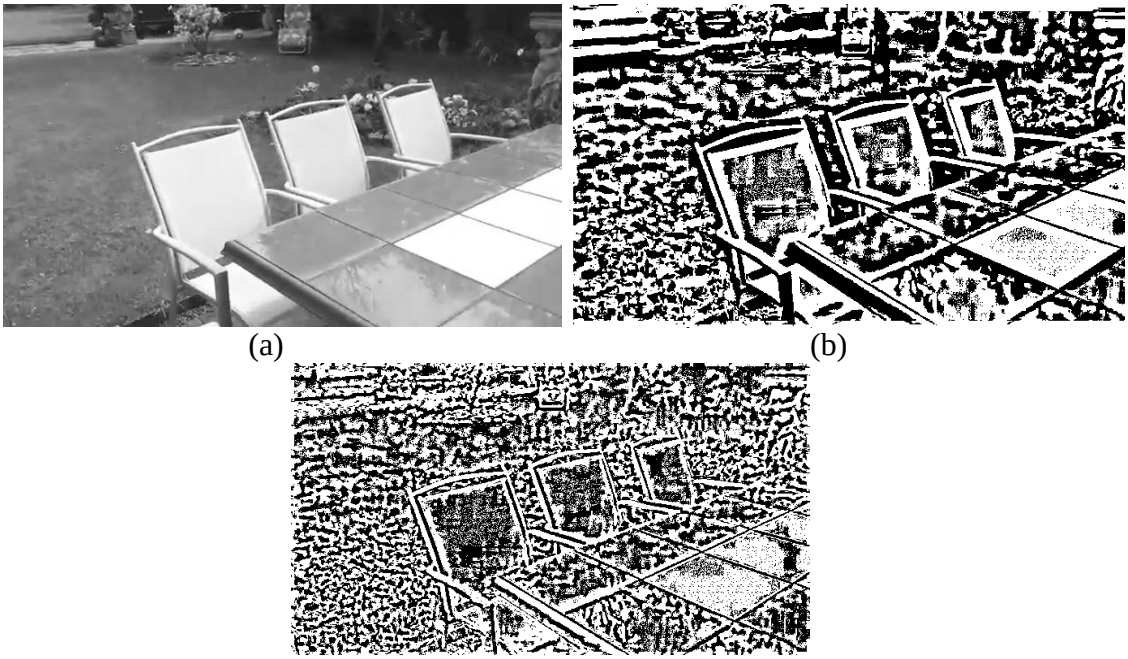
$$NNMP(m,n)=\sum_{i=-p/2}^{p/2}\sum_{j=-q/2}^{q/2}\left[B^{t}(i,j)\oplus B^{t-1}(i+m,j+n)\right] \quad (1.10)$$

Eğer arama penceresi s sınırında kaydırılır ise $-s \le m,n \le s$ ile gösterilir.

1.5. Düşük Bit Çözünürlüklü İmge Dönüşümleri

Literatürde tam arama temelli 2-boyutlu bütünsel hareket kestirim yöntemlerinin hesapsal karmaşıklığını azaltmak için farklı yaklaşımlar önerilmiştir. Görüntü çerçevelerinin doğal kodlanmış ve Gri-kodlanmış bit uzaylarının uyumlama aşamasında mantıksal (logical) işlemler kullanılarak ele alınması ile özellikle donanımsal uygulamalarda önemli avantajlar sağlanabileceği [28,29]'da gösterilmiştir. [30]'daki çalışmada imge çoklu bant geçiren bir filtreden geçirilip orijinal imge ile karşılaştırılması sonucu bir bitlik imgeler (1BT) oluşturulmaktadır. Çarpmasız bir-bit dönüşümü (Multiplication-free one-bit transform - MF-1BT) temelli hareket kestirim yaklaşımı çekirdek süzgecindeki sıfır olmayan değerlerin sayısının ikinin üstü bir sayıya getirerek normalizasyon işleminin hızlı ve donanımsal olarak daha kolay uygulanmasına sağlayan bir yöntem önermiştir [31]. [32]'deki çalışmada ikinci bir-bit düzlemi eklenerek uyumlamaya sadece güvenilir ikili pikseller katılıp başarım arttırılıştır. [30]'da önerilen yöntem bütünsel hareket kestirimine uygulanıp imge çerçevesinin kenarlarına yerleştirilmiş alt imgeler kullanılarak hesaplama yükünün azaltılması önerilmiştir [33]. [32]'de önerilen yöntemin bütünsel hareket kestirimine uyarlanıp [33]'deki çalışmada kullanılan

kenar yerleşimli alt imgelerin bütünsel hareket başarımını düşürmesi sebebi ile tek ve merkez yerleşimli alt imgeler üzerinden yapılması [34]'de önerilmiştir. [35]'de yakın zamanda önerilen bir yöntem ise yerel ikili örüntü temelli bir yaklaşım ile elde edilen ikili görüntüler üzerinden NNMP uyumlama ölçütü kullanılarak bütünsel hareket kestiriminin yapılmasını önermektedir. Şekil 1.11'de farklı bir bit dönüşümü çerçeve örnekleri görüntülenebilir.



(a) (b)

(c)

Şekil 1.11. 1-bit derinlikli imge çerçeve örnekleri, a) orijinal imge, b) alt imge tabanlı yöntemi ile oluşturulmuş ikili imge, c) yerel ikili örüntü tabanlı yöntem ile oluşturulmuş ikili imge

1.5.1. Bir-bit dönüşümü (1BT)

1BT yönteminde 8 bitlik imge çerçeveleri Eşitlik (1.11)'de verilen çekirdek kullanılarak çoklu bant geçiren süzgeçten geçirilir [33].

$$K(i,j)=\begin{cases}1/25, & \text{eğer } i,j \epsilon [1,4,8,12,16] \\ 0, & \text{aksi halde}\end{cases} \tag{1.11}$$

Giriş imgesi çoklu bant geçiren süzgeçten geçirildikten sonra Eşitlik(1.12)'de belirtilen karşılaştırma ifadesi ile ikili imgeye dönüştürülür.

$$B(i,j)=\begin{cases}1, & I(i,j)\geq I_F(i,j)\\ 0, & \text{aksi halde}\end{cases} \tag{1.12}$$

Eşitlikteki $I_F(i,j)$ belirtilen çoklu bant-geçiren süzgeç ile filtrelenmiş imgeyi göstermektedir.

1.5.2. Çarpmasız bir-bit dönüşümü (MF-1BT)

1BT yöntemi filtre çekirdek ifadesinde 25 normalizasyon değerinin olması sebebi ile filtreleme işlemi sadece tam sayı aritmetiği kullanılarak yapılamamaktadır. Öte yanda, noktalı aritmetik hesaplamalar donanımsal uygulamalar için göreceli olarak daha yüksek işlem yüküne sahiptir. Bu yüzden [31]'de 1BT'de kullanılan çekirdek güncellenerek normalizasyon katsayısı 2'nin kuvveti şeklinde belirtilmiştir. Eşitlik (1.13)'de MF-1BT'de kullanılan çekirdek süzgeci verilmiştir [31].

$$K=\frac{1}{16}\begin{bmatrix}
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0
\end{bmatrix} \tag{1.13}$$

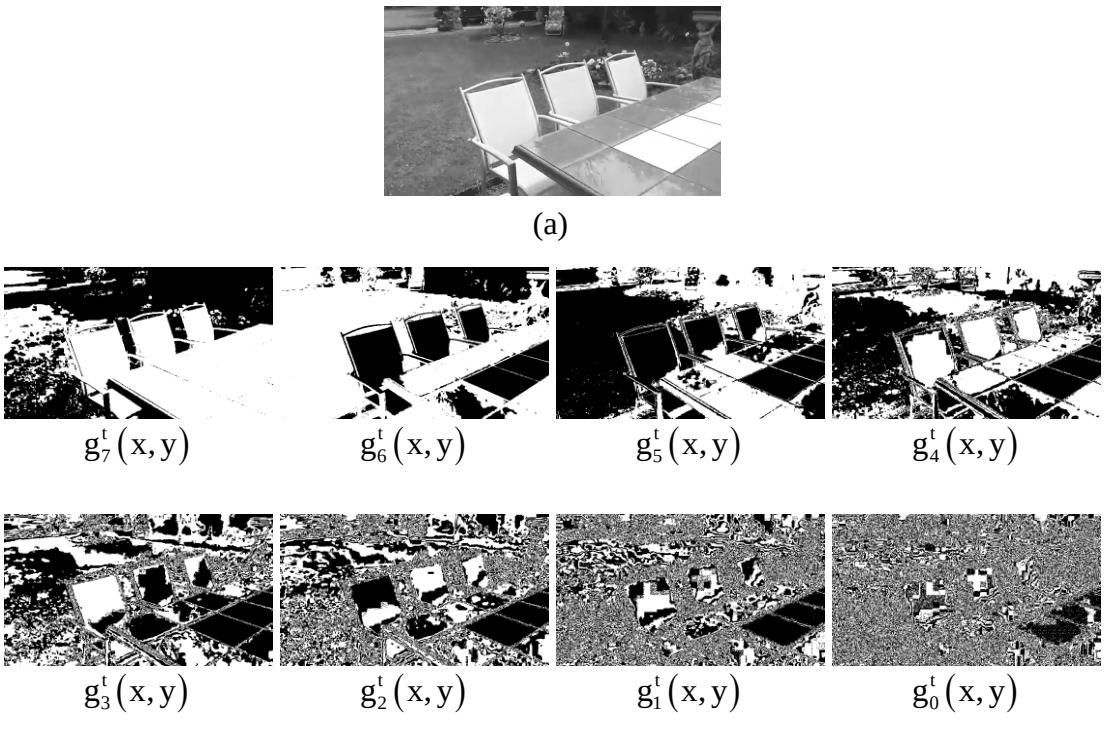
Böylelikle normalizasyon işlemi tam sayı aritmetiği kullanılarak öteleme işlemi ile yapılıp hızlı donanımsal uygulamaların geliştirilmesi sağlanmıştır.

1.5.3. Geçmeli gri-kodlamalı bit-düzlemi temelli bir-bit dönüşümü (interlaced gray-coded bit-plane based one-bit transform – IGCBP-1BT)

[36]'daki çalışmada ikili dönüşüm maliyetini azalmak için gri-kodlamalı bit düzlemlerini taramalı olarak kullanan bir bit dönüşümü önerilmiştir. Piksel değerinin K-bit gri kodu Eşitlik (1.14) ile hesaplanmaktadır [36].

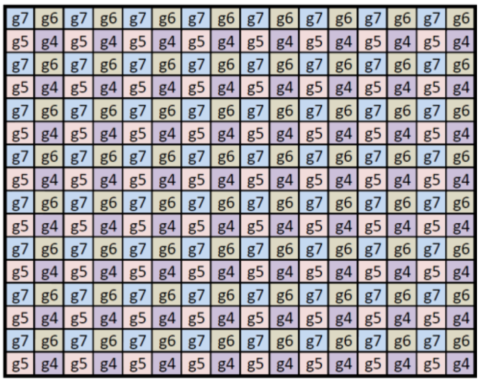
$$\begin{aligned} g_{K-1} &= \alpha_{K-1} \\ g_k &= \alpha_k \oplus \alpha_{k-1}, 0 \le k \le K-2 \end{aligned} \tag{1.14}$$

Burada α piksellerin ikili doğal değerlerini göstermektedir. Şekil1.12'de örnek gri-kodlanmış bit-düzlemleri görünmektedir [29].



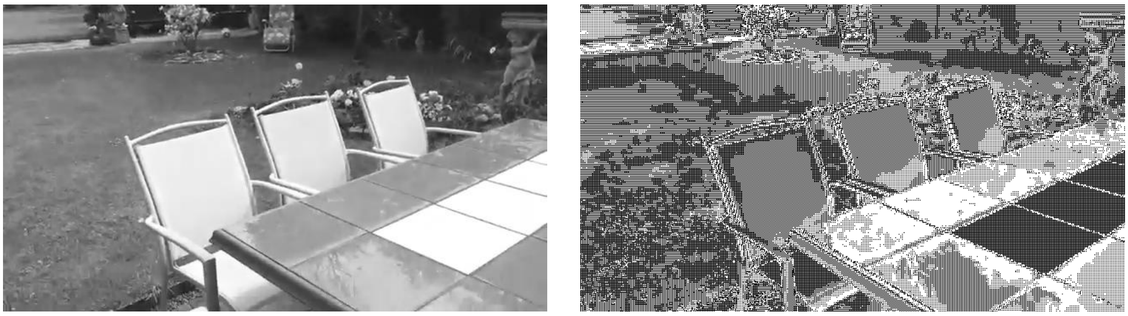
Şekil 1.12. Gri-kodlamalı bit-düzlemleri, a) orijinal imge, b) 8 adet gri-kodlanmış bit düzlemleri

Çerçevelere Gri-kod dönüşümü yapıldıktan sonra gri-kodlanmış bit düzlemlerinden en önemli 4 tanesi kullanılarak farklı bit düzlemleri üzerinden tek bir bit düzlemi kombinasyonu oluşturularak bir bit dönüşümü tamamlanır (Interlaced Gray-coded bit-plane) [36]. Şekil 1.13'de bit-düzlemi seçim yöntemi gösterilmektedir.



Şekil 1.13. Bit-düzlemi seçim yöntemi

Şekil 1.14’de IGCBP-1BT dönüşümün uygulandığı “table” imgesi görünmektedir



(a) (b)

Şekil 1.14. a) Orijinal imge, b) IGCBP-1BT dönüşümü sonrası

1.5.4. Yerel ikili örüntü (local binary pattern -LBP) temelli dönüşüm

[35]'te ise LBP yaklaşımı ile imgelerin ikili hale dönüştürülmesi sonrası bütünsel hareket kestirimi yapılması önerilmektedir. Bu yaklaşımda bütün pikseller R yarıçaplı komşulukları ile karşılaştırılır. Komşulukları tanımlamak önceden tanımlı LBP operatörü kullanılır. LBP yönteminde giriş imgesindeki her piksel değeri P adet eşit uzaklıklı R yarıçapındaki komşulukları ile karşılaştırılır. (i, j) noktasında verilen piksel için LBP değeri Eşitlik (1.15) ile gösterilebilir [35].

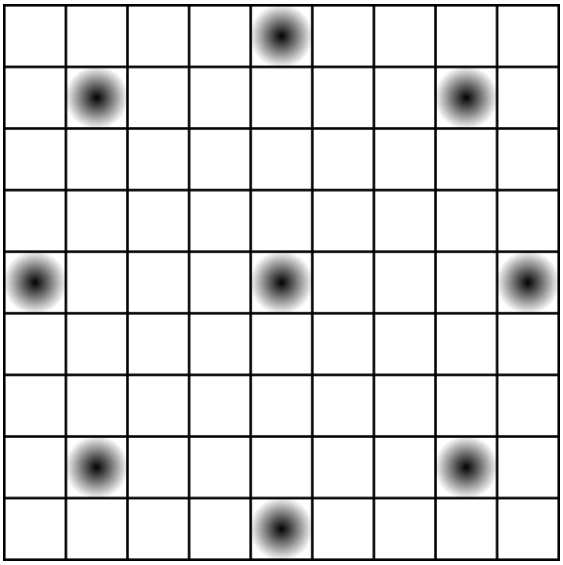
$$LBP_{(P,R)}(i,j) = \sum_{p=0}^{P-1} sgn(I(p) - I(i,j)) \times 2^{p}$$
$$sgn(x)\begin{cases} 1 & x \geq 0 \\ 0 & \text{aksi halde} \end{cases} \tag{1.15}$$

Eşitlik (1.15)'de p ve 2^p (i,j) konumuna eşit uzaklıktaki pikselleri ve binom faktörünü göstermektedir. LBP ikili dönüşümü matematiksel ifadesi Eşitlik (1.16)'de görülebilir [35].

$$B(i,j)\begin{cases} 1 & \text{eğer} \sum_{p=0}^{P-1} \text{sgn}\left(I(p)-I(i,j)\right) \geq P/2 \\ 0 & \text{aksi halde} \end{cases} \tag{1.16}$$

Burada x x 'den büyük olmayan en büyük tam sayı olarak kullanılmıştır.

Şekil 1.12'de $LBP_{(8,4)}$ yani 8 örnek ve 4 yarıçaplı operatör tanımlanarak kullanımı örneklenmiştir. Merkez piksel değeri 4 yarıçaplı daire içinde 8 nokta ile karşılaştırılır. Eğer bu karşılaştırmalardan en az 5 tanesinde merkez pikselin değeri diğer komşuluklardan büyük ise merkez pikseli değeri 1-bit derinlikli imgede "1" değerine eşitlenir. Eğer bu koşul geçerli değil ise ikili imgede bu piksel değeri "0" olarak ayarlanır.



Şekil 1.15. $LBP_{(8,4)}$ (8 örnek, 4 yarıçaplı operatör)

.

2. İKİLİ BÜTÜNSEL HAREKET KESTİRMİNİN ARM İŞLEMCİ ÜZERİNDE SIMD İLE GERÇEKLENMESİ

2.1. Giriş

Video sıkıştırma, stabilizasyon gibi bir çok uygulamada kullanılan ikili bütünsel hareket kestiriminde gerçek zamanlı hesaplama yapılması önem taşımaktadır. Bu nedenden dolayı bir önceki bölümdeki verilen literatür özetinde de açıklandığı gibi hareket kestirimini hızlandırmak için birçok önermede bulunulmuştur. Günümüzde kişisel bilgisayarlarımız hızlanmasına rağmen mobilitenin artması ile güç tüketiminin ve işlemci gücünün verimli şekilde kullanılması gerekmektedir.

İkili bütünsel hareket kestirim yöntemlerinin önerilmesi donanımsal uygulamaların daha kolay geliştirilmesinin önünü açmıştır. İkili hareket kestirim yöntemlerinin donanımsal olarak etkin bir şekilde gerçeklenmesini sağlayacak mimariler önerilmiştir [37,38]. Bilindiği kadarı ile ikili bütünsel hareket kestirim algoritmalarının yazılımsal olarak etkin şekilde gerçeklenmesini sağlayacak yaklaşımlar literatürde bulunmamaktadır. Günümüz işlemcilerinin birçoğu dâhili şekilde tek komut çoklu veri SIMD motorları ile piyasaya sunulmaktadır. Böylece çok fazla veri içeren ve tekrarlı hesaplamaların gerektiği grafik veya fizik hesaplamaları için alt yapı sağlamaktadır.

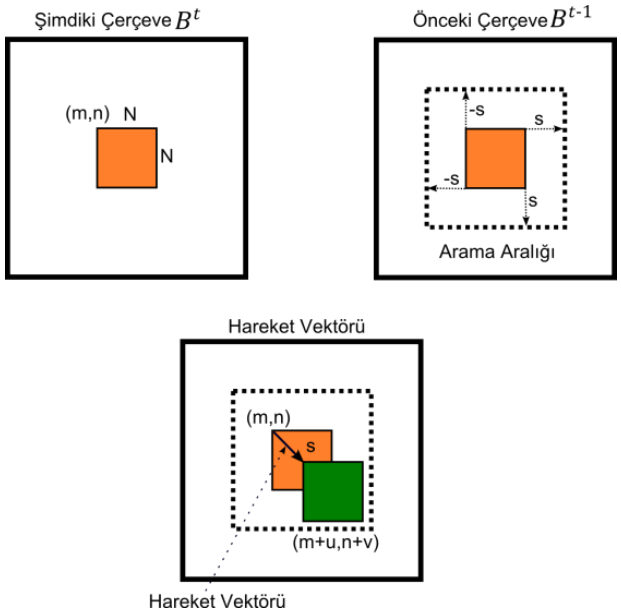
Bu tez kapsamında ikili bütünsel hareket kestirim yaklaşımlarının SIMD motorunu kullanarak yazılım tabanlı hızlandırılması için bir yöntem önerilmiştir. Çalışma kapsamında MF-1BT, LBP, IGCBP-1BT temelli ikili bütünsel hareket kestiriminin gerçek zamanlı performansı ve başarımı ele alınmıştır.

2.2. Uygulanan Yöntem

Yerel ikili örüntü yaklaşımı kullanılarak elde edilen ikili görüntüler üzerinden bütünsel hareket kestiriminin yapılması önerilmektedir [35]. Bu tez kapsamında [35]'de önerilen blok uyumlama yaklaşımının ikili imgeler üzerinde çalışmaya elverişliliğinin avantajları kullanılarak SIMD ile gerçeklemesi ile gerçek zamanlı

performansı değerlendirilmiştir. Blok uyumlamaya girdi olarak verilecek ikili çerçeveler için sadece LBP temelli yöntem değil MF-1BT ve IGCBP-1BT temelli farklı ikili dönüşümler de uygulanmıştır. Böylece ikili dönüşüm maliyetleri de deneysel sonuçların içine dâhil edilmişidir. Bölüm 1.5'de farklı ikili dönüşümler özetlenmiştir. Bu kısımda blok uyumlama yaklaşımı ve performans için kullanılan metrikler açıklanacaktır. Yöntem, öteleme (translation) hareket modeline göre yatay ve düşey hareketleri incelemekte ve parametre olarak 2 değişkeni çözümleyerek hareket vektörünü tanımlamaktadır.

Ön şart olarak ilgili çerçeve piksel değerleri ikili imge dönüşüm adımlarından uygun olanı ile ikili imgelere dönüştürülür. Blok uyumlama ölçütü olarak bölüm 1.4.5'de açıklanan NNMP kriteri kullanılarak bütünsel hareket kestirimine tabi tutulur.



Şekil 2.1. Tam arama blok uyumlama yaklaşımı

Eşitlik (1.10)'da (m,n) $N\times N$'lik seçilen imge blokunun s sınırında kaydırma sonrası aday yer değişimlerini göstermektedir. Aranacak blok merkezde seçilen alt piksellerin bütünsel hareketin başarımında etkili olması ve işlem yükü azaltması sebebi ile çerçevenin merkezinde seçilir [35]. Şimdiki çerçeve B^t ve bir önceki çerçeve B^{t-1} olsun. Şimdiki çerçevenin merkezinden alınan $N\times N$ boyutundaki blok önceki çerçevedeki s sınırında piksel-piksel kaydırılarak (adım aralığı "1") her kaydırma işleminde bütün blok piksel değerleri XOR operatöründen geçirilerek tam arama blok yaklaşımı kullanılır. Uyumsuz noktaların en az olduğu yani NNMP

ölçütü en düşük hesaplandığı $(m+u, n+v)$ noktası ilgili blok için hareket vektörü olarak atanır.

Bütünsel hareket kestirim yaklaşımlarının başarımını değerlendirmek için [35]'de karesel ortalama hata RMSE (Root Mean Square Error) yaklaşımı kullanılmaktadır. Kullanılan başarın ölçütü Eşitlik (2.2)'deki gibidir.

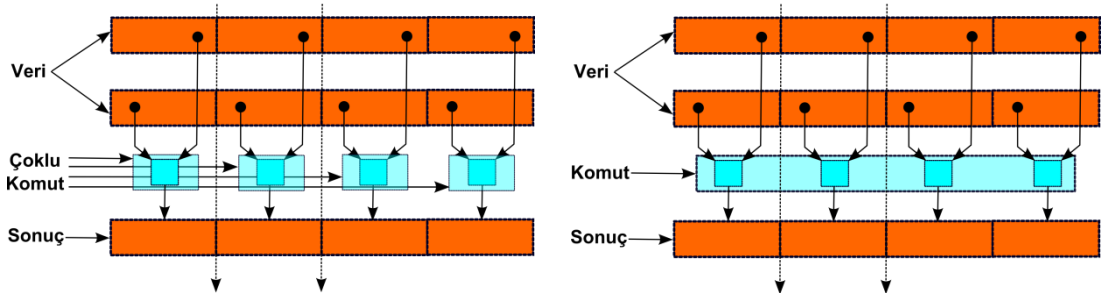
$$\mathrm{RMSE} = \frac{1}{N}\sqrt{\sum_{k=1}^{N}\left(x_k - \hat{x}_k\right)^2 + \left(y_k - \hat{y}_k\right)^2} \qquad -s \le m, n \le s \tag{2.2}$$

(x_k, y_k) MSE uyumlama ölçütüne göre tam arama yapılarak çıkarılmış hareket vektörlerini göstermektedir. $(\hat{x}_k, \hat{y}_k)$ ise önerilen yöntem sonucu çıkartılan hareket vektörlerini göstermektedir. Uygulanan yöntem sonucu çıkan hareket vektörleri ile MSE ölçütüne göre çıkartılmış hareket vektörlerinin ortalama karesel hatası hesaplanmaktadır. Amaç yöntemin kestirim başarımını MSE uyumlama ölçütü ile çıkarılmış vektörlere olan farkı üzerinden tanımlamaya dayalıdır. Deneysel sonuçlardaki başarım değerleri RMSE ölçüne göre yapılacaktır.

2.3. Tek Komut Çoklu Veri (SIMD)

İşlemci mimarileri genel olarak komut bazında dört ana başlık; tek komut tek data (Single Instruction Single Data - SISD), tek komut çoklu data (Single Instruction Multiple Data - SIMD), çoklu komut tek data (Multiple Instruction Single Data - SISD) ve çoklu komut çoklu data (Multiple Instruction Multiple Data - SIMD) olarak sınıflandırılır. Genel olarak günümüz işlemcileri işlem paralelliği (task parallelism) ile daha verimli çalışacak şekilde tasarlanmaktadır. Çoklu çekirdek mimarisi aynı anda birden fazla işlem yapmaya olanak sağlamaktadır. Bu dört ana başlığa çoklu işlemcilerin yaygınlaşması ile çoklu programlama alanı da eklenebilir. Fakat özellikle yoğun miktarda veri işlemek ihtiyacı olan video işleme gibi uygulamalar veri paralelliği (data parallelism) mimarisine ihtiyaç duymaktadır. SIMD işlemcilerin tek komut üzerinden çoklu veriyi paralel ve etkin şekilde işlemesine olanak sağlamaktadır. GPU'lar genel olarak veri paralelliği mimarisine uygun şekilde tasarlanmakta ve SIMD motorları içermektedir. Bunun yanı sıra artık günümüz

işlemcilerinde SIMD motorları bulunmaktadır. Mobilleşmenin getirisi ile düşük güç tüketimine sahip birçok mobil işlemci bile dâhili SIMD motorlarını barındırmaktadır. Genel olarak kullanım alanları çok büyük verinin tekrarlı bir şekilde aynı işleme tabi tutulacağı durumlar için idealdir. Örnek uygulama alanları grafik, kodlama, fizik hesaplarının yapılmasında kullanılmaktadır. Standart bir işlemci çoklu veriyi işlemek için skaler bir işleme tabi tutmaktadır. Aynı miktarda veri, SIMD operatörleri ve veri genişliği fazla olan özel saklayıcılar (register) kullanarak sadece tek bir komut ile işlenebilmektedir. Örneğin ARM işlemcilerde kullanılan NEON SIMD motoru 128 bitlik saklayıcılar sunmaktadır. Böylece 32-bitlik 4 adet veriyi tek bir komut ile işleyebilmektedir. Aynı miktar 32-bitlik veriyi işleyebilmek için normal işlemcide dört komut gerekir iken SIMD kullanımı ile aynı miktar veri bir komut ile işlenebilmektedir. Diğer önemli bir nokta ise standart bir komutun işleyeceği verinin boyutu azaldığında ortaya çıkmaktadır. SIMD 128 bitlik saklayıcılar farklı vektör tipleri ile çalışabilmektedir. Örneğin eğer 1 byte'lık veriler ile çalışılacak ise 128 bit vektör saklayıcısına 8 adet 1 bytelık veri yazılabilmektedir. Bu durumda normal işlemcide 8 adet 1 byte'lık veri işlemek için 8 skaler komuta ihtiyaç duyulur iken SIMD bunu tek komut üzerinden yapabilmektedir. Şekil 2.2'de görüldüğü gibi SIMD adımı 4 komutta işlenebilen bir skaler işlemi tek komut vasıtası ile tamamlayabilmektedir. Farklı firmaların SIMD mimarilerinde saklayıcı uzunlukları farklı olabildiğinden Şekil 2.2'de veri uzunlukları gösterilmemiştir.

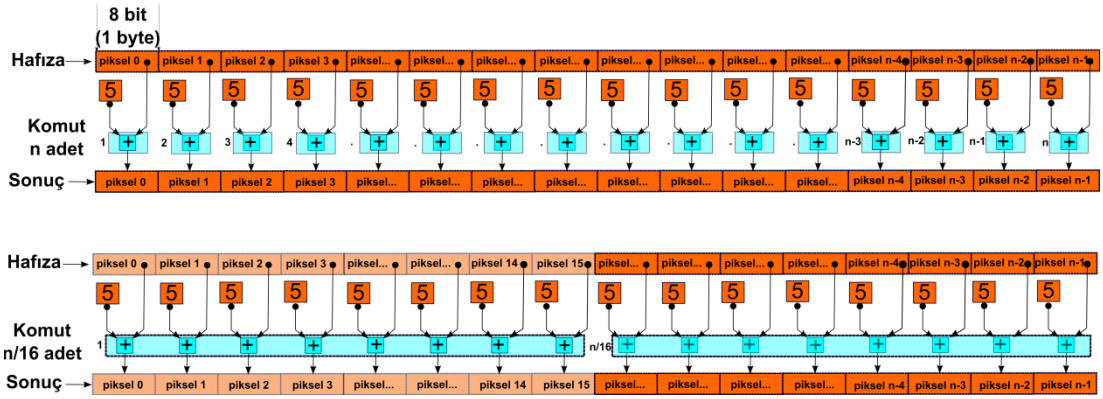


(a) (b)

Şekil 2.2. a) Skaler işlemi b) tek komut çoklu veri işlemi

Örnek olarak gri tonları içeren 8-bit derinlikli bir imgenin ışıklılık değerlerinin değiştirilmesi ele alınsın. Işıklılık değerini değiştirmek için piksel değerleri hafızadan okunarak sabit bir değer ile ekleme veya çıkarma yapılacak şekilde aynı işlem defalarca tekrarlanacaktır. Örneğimizde 5 sabit değerinin her piksele ekleneceğini

düşünelim. Boyutları $p \times q = n$ olan bir imge hafızada n byte yer kaplamaktadır. Her bir piksel 5 değeri eklemek için toplamda n adet komutu 1 byte'lık veri üzerinde çalıştırmak gerekir. Eğer işlem 128 bitlik SIMD saklayıcıları üzerinden yapılacak olur ise 1 byte'lık piksel değerlerinden 16 tanesi 1 SIMD vektör saklayıcısına yerleştirilerek tek komut ile 16 piksel değeri tek seferde hesaplanabilmektedir. Böylece aynı miktarda veriyi işlemek için gereken komut miktarı $n/16$ adet olacak şekilde azalacaktır. Şekil 2.3'de n adet piksele sahip imgeye sabit 5 değerinin standart ve SIMD yöntemi ile eklenmesi gösterilmiştir.



Şekil 2.3. SIMD ve standart komut n piksele sahip imge üzerinde karşılaştırması

Burada SIMD iki temel başarım unsuru sunmaktadır. Birincisi normal işlemci tasarımındaki gibi pikseller tek tek hafızadan okunmak yerine 16 adet piksel değerinin tek komut ile SIMD saklayıcılarına yüklenmesi sağlanır. İkinci avantajı ise tek bir komut ile yüklenen 16 adet piksele sabit ışıklılık değerini tek komut ile yazmasıdır. Normal bir işlemcide piksel başına tekrarlanacak komut 16 adet piksele tek bir kerede uygulanarak gereken işlem gücünü önemli ölçüde azaltacaktır. SIMD operatörlerini verimli şekilde kullanabilmek için veri yapısının vektör işlemeye uygun şekilde olması gerekmektedir. SIMD'nin en büyük dezavantajlarının başında veri yapısının uyumlandırılması ve algoritmanın sadece aynı ve tekrarlı komutlara ihtiyaç duymasıdır. Örneğin veri vektöründeki n adet pikselin bazıları için toplama bazıları için çıkarma işlemi tek komut üzerinden yapılamamaktadır. Günümüz derleyicileri bazı algoritmalar için SIMD komutlarını otomatik üretebilmesine karşılık başarım oldukça düşüktür. Vektörizasyon konusu başlı başına bilgisayar

bilimlerinin bir çalışma alanıdır. Bu sebep ile uygulamada kullanılacak algoritmadaki veri yapıları ve komutları etkin kullanım açısından kişi tarafından tasarlanmalıdır.

2.4. SIMD Temelli Bütünsel Hareket Kestirimi

[35]'de verilen performans bilgileri ve SIMD iyileştirmeleri öncesi uygulanan standart yöntemden elde edilen deneysel sonuçlar en fazla işlem gücüne gereksinim olan alanın NNMP uyumlama ölçütü hesabı olduğunu göstermiştir. Bu sebep ile SIMD komutları uygulama aşamasında NNMP değeri hesaplamalarında kullanılmıştır. NNMP hesabında şimdiki çerçeve B^t'den alınan $N \times N$ boyutundaki blok ve bir önceki çerçevedeki (B^{t-1}) s sınırında kaydırma işleminde bütün blok piksel değerleri XOR operatöründen geçirilmektedir. Dolayısı ile sabit bir veri dizisi ve sabit bir XOR operasyonu yapılmaktadır. Bu yöntem donanımsal tasarımı kolaylaştırmak için önerilmiş olmakla birlikte SIMD için de oldukça uygundur.

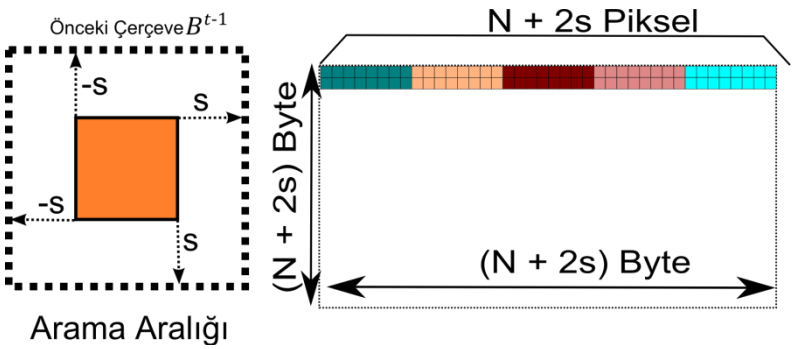
Uygulamada öncelikle şimdiki ve önceki çerçevelerin ikili dönüşümü yapılmalıdır. Bu tez çalışmasında ikili dönüşümler SIMD iyileştirilmesine sokulmamıştır. Farklı dönüşümler (LBP, MF-1BT, Interlaced Gray-coded Bit-Plane) kullanılarak ikili dönüşümlerin hesapsal maliyetleri deneysel sonuçlarda verilmiştir.

2.4.1. Veri yapılarının SIMD öncesi uyumlandırılması ve yüklenmesi.

SIMD verilerin işlenebilmesi için mimari özelinde faklı boyutlarda vektör saklayıcıları sunmaktadır. Örnek olarak 64 bit ve 128 birlik saklayıcılar düşünülebilir. Saklayıcılar verinin yapısına göre farklı şekilde vektörlere bölünebilir (8bit, 16bit, 32bit, 64bit, 128bit gibi). Birçok işlemci SIMD kütüphanesi farklı vektör ve yükleme yapılarını desteklemektedir.

İkili imge dönüşümü sonucu alınan çerçevelerde yazılımsal işlem karmaşıklığını azaltmak için 1 bitlik veriler için 1byte'lık bir alan kullanılmıştır. Dolayısı ile ikili imge dönüşümü sonucu elimizde çerçeve uzunlukları boyutunda bir byte dizisi bulunmaktadır. Verileri bu şekilde SIMD işlemine sokmak performansı azaltacağından öncelikle 1-bitlik piksel değerleri için veri uyulmaması yapılmaktadır. Aksi halde SIMD'de bir komut ile 128 bit işlem yapılabilecekken, bir bit veri için 8 bitlik hafıza harcanacağından $128 \div 8 = 16$ adet veri işlenebilecektir. Şekil 2.4'de

NNMP hesaplamada şimdiki çerçeveden alınan $N \times N$ boyutundaki blok önceki çerçevede s sınırında kaydırılır iken SIMD'nin verimli kullanılması için gereken veri uyulmaması yapılmaz ise $(N+2s) \times (N+2s)$ byte olacak bir veri dizisi üzerinde çalışılacağı görülmektedir.

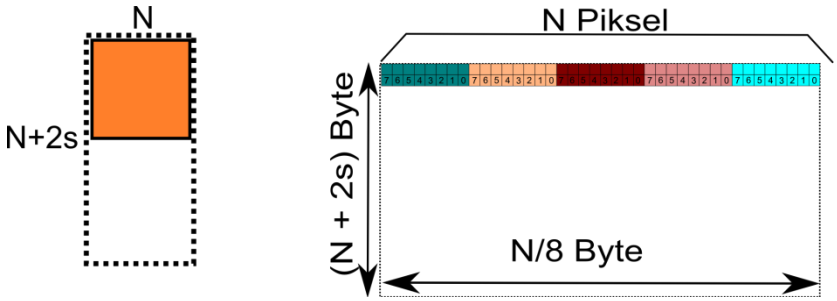


Şekil 2.4. Uyumsuz nokta sayısı hesaplama

Bu veri dizisi üzerinde şimdiki çerçeveden alınan $N \times N$ boyutundaki blok $2s+1$ kez yatayda ve aynı miktarda düşeyde 1 adım aralıklı piksel kaydırma ile XOR işlemi yapılacaktır. SIMD için 1 bitlik piksel değeri daha uygun bir yapıya getirilmelidir. Bu neden ile yatay ve düşeydeki kaydırma adımları göz önüne alınarak yatay eksendeki piksellerin 8 tanesi gruplanıp 1 byte olacak şekilde piksel değerlerinin yerleştirilmesi önerilmiştir. Böylece kullanılan veri dizisinde 8 kat azalma sağlanmıştır. Veri uyumlamasının blok karşılaştırma işlemini SIMD'ye en uygun şekilde yapabilmek için yatay ve düşey kaydırma işlemlerdeki önermeler aşağıdaki adımlardan oluşmaktadır.

Birinci adımda; şimdiki çerçeveden alınan blok için 1 byte yerleştirme işlemi yapılır. Böylece blok $N \times N/8$ bytelık bir dizi ile temsil edilebilir. Bu adımın bir kere yapılması yeterlidir.

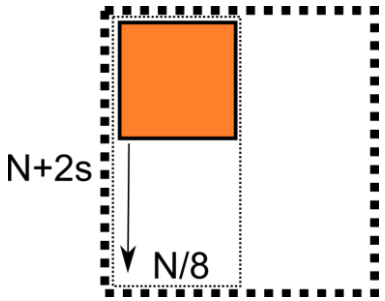
İkinci adımda; düşey kaydırma işlemin yapılması 1 byte veri yerleştirme adımlarını azaltacağı için ilk olarak düşey kaydırma yapılacaktır.



Şekil 2.5. Çerçeve piksel değerleri için veri uyumlama

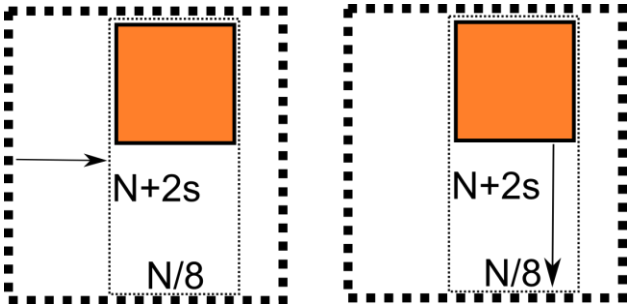
Şekil 2.5'de görüldüğü gibi düşey kaydırmada önceki çerçevedeki $(N+2s)\times N$ piksel değeri için 1 byte yerleştirme yapılır. Böylece önceki çerçevedeki pikseller $(N+2s)\times N/8$ byte'lık bir dizi ile temsil edilebilir.

Üçüncü adımda; ilk blok için XOR işlemleri üzerinden NNMP hesaplandıktan sonra Düşey kaydırma işlemi kolayca 1 byte'lik dizinin sonraki elemanını seçilerek yapılabilir. Şekil 2.6'da düşey kaydırma işlemi görünmektedir. Böylelikle SIMD saklayıcılarına byte bazında veri yüklenebilir.



Şekil 2.6. Önceki çerçevede düşey kaydırma

Dördüncü adımda; yatay düzlemde piksellerin 1 byte yerleştirme işlemi bloğun *s* sınırında yatay kaydırma işlemini yazılımsal olarak zorlaştırmıştır. Çünkü bir piksel bir bit ile gösterilmekte, adım aralığı 1 olarak seçildiğinde kaydırma işlemi 1 byte yerleştirilmiş verilerin tekrar hesaplanmasını gerektirmektedir. Bu sebep ile her bir piksellik kaydırma işlemi sonucu ikinci adımdaki piksel yerleştirme işlemleri tekrarlanarak SIMD byte vektörlerine uygun veri yapısı tekrar sağlanmakta ve üçüncü adımdaki düşey kaydırma işlemi Şekil 2.7'deki gibi tekrarlanmaktadır.



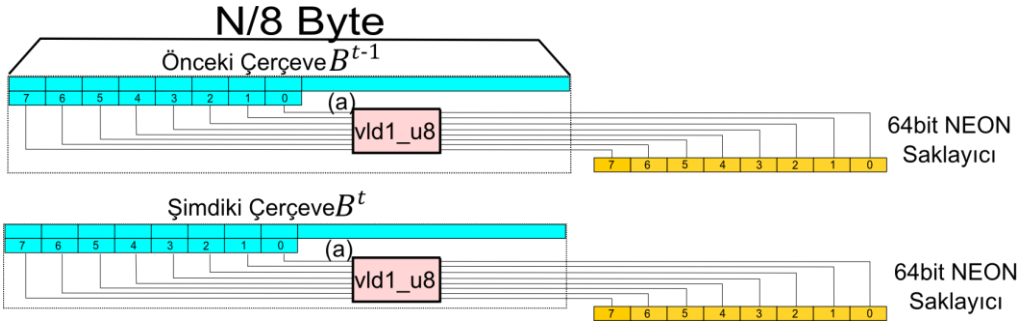
Şekil 2.7. Önceki çerçevede yatay kaydırma

Böylelikle bir önceki ve şimdiki çerçevelerde ilgili piksel değerleri SIMD byte vektörlerine uygun şekilde yüklenmek üzere byte dizilerine çevrilmektedir. Bu adımın yapılması bir SIMD komutu ile daha fazla veri işlenmesi sağlamaktadır.

2.4.2. NNMP hesabının SIMD ile gerçeklenmesi

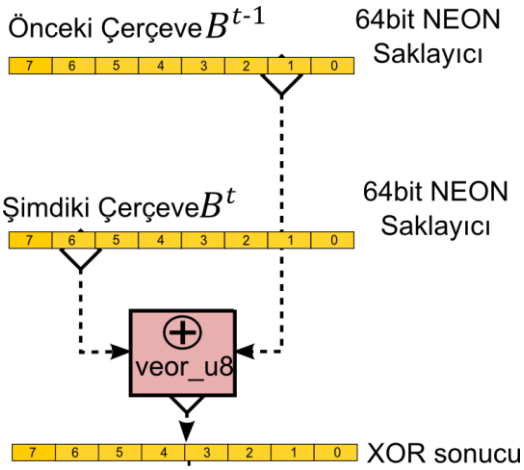
NNMP hesabı yapılırken şimdiki çerçeveden alınan ve 1 byte veri uyumlaması yapılmış $N \times N/8$ boyutundaki blok, önceki çerçevedeki arama bölgesinde her kaydırma işleminden önce tüm piksel değerleri için XOR hesabı yapılmaktadır. Bu kısımda byte dizisi içindeki piksel değerlerinin nasıl SIMD vektörlerine yerleştirdiği ve bir blok için hesaplamanın nasıl yapıldığı anlatılacaktır. Anlatımda kullanılan SIMD komutları deneyler aşamasında kullanılan ARM mimarisinin NEON SIMD motoru kütüphanesinden örneklenmiştir.

Birinci adımda; bir blok için yatayda $N/8$ adet byte bulunmaktadır. Şekil 2.8'de görüldüğü gibi 64 bitlik SIMD saklayıcılarına yatay piksel değerleri byte dizisi olarak yüklenir. Yükleme için "vld1_u8" SIMD komutu kullanılmaktadır. Bu komuttaki "vld" kısaltması vektör yükleme, "1" vektör elemanlarını yükleme sırasındaki boşluk (interleave pattern), "u8" ise vektörün "unsigned integer" yanı byte vektörü olacağını belirtmektedir.



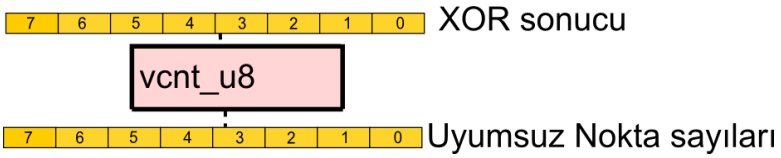
Şekil 2.8. SIMD saklayıcı veri yükleme

İkinci adımda; bit veri yani 64 piksel değerini içeren SIMD byte vektör saklayıcıları tek bir komut ile XOR operatöründen geçirilerek her piksel için uyumlama değeri hesaplanır. Böylece 64 bitlik SIMD saklayıcısı kullanarak $(N/8)/8)$ adet işlem ile yataydaki tüm pikseller işlenmiş olur. 64 bitlik SIMD saklayıcılarına XOR adımı Şekil 2.9'da gösterilmektedir. XOR sonucu geçici olarak bir SIMD byte vektör saklayıcıına yazılır. Çerçevelerin ikili olarak tanımlanması ve uyumlama ölçütünün mantıksal XOR işlemlerini sunması sayesinde tek mantıksal SIMD komutu ile bütün piksel değerleri için hesaplama yapılabilmektedir.



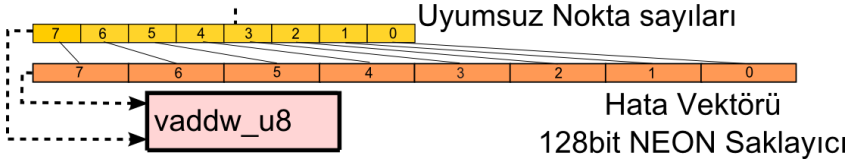
Şekil 2.9. SIMD XOR işlemi

Üçüncü adımda; elde edilen XOR sonucu bit bazında elde edilmiştir. NNMP hata vektörü hesaplaması için XOR sonucu "1" yanı uyumsuz noktaların sayısının toplanması gerekmektedir. Şekil 2.10'da byte dizisi içinde her vektör elamanına ayrı uygulayıp "1" olan bitleri sayan "vcnt_u8" komutu kullanılmıştır.



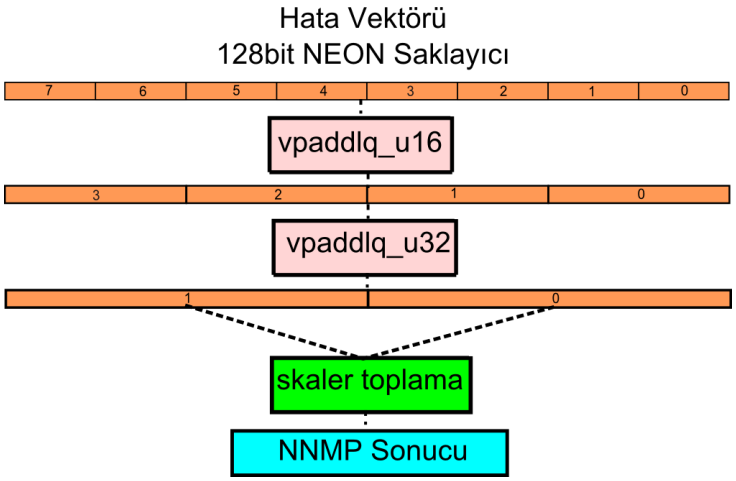
Şekil 2.10. SIMD uyumsuz nokta hesabı

Dördüncü aşamada; bir önceki aşamada elde edilen uyumsuz nokta sayıları 128 bitlik hata vektörüne atanmaktadır. 128 bitlik hata vektörü hatalı nokta sayısını tutabilmek ve vektörsel olarak hızlı işlem yapabilmek için 2byte'lık vektör şeklinde tanımlanmıştır. Şekil 2.11'de hata vektörü hesabı görüntülenebilir.



Şekil 2.11. SIMD hata vektörü hesabı

Beşinci ve son adımda; ilk dört adım bütün $N \times N$ blok piksel değerleri için hesaplanana kadar tekrarlı bir şekilde devam eder. Bütün blok bittikten sonra elimizde 128-bit 8 elemandan oluşan bir hata vektörü elde edilir. Son NNMP hata değeri hesaplanması için bu vektördeki bütün 8 değerin toplanması gerekmektedir. Şekil 2.12'de normalde 8 toplama komutu ile yapılacak işlem ise SIMD yardımı ile 3 adımda tamamlanır.

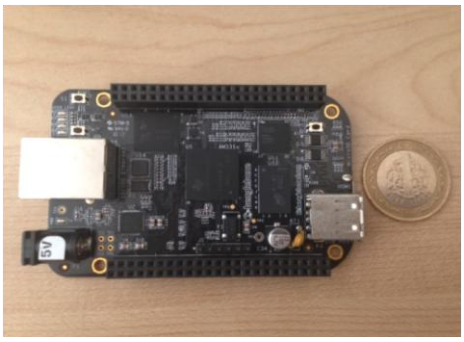


Şekil 2.12. SIMD uyumsuz nokta sayısı hesabı

3. DENEYSEL SONUÇLAR

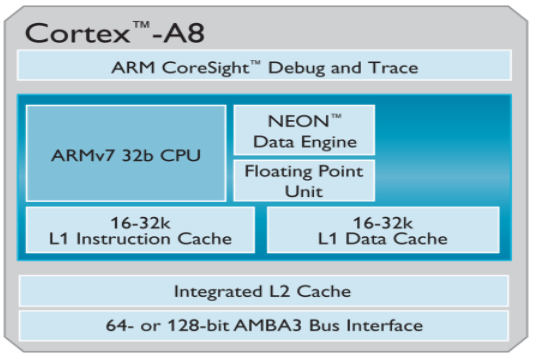
Tez çalışmasının bu kısmında sunulan ikili bütünsel hareket kestirimin SIMD alt yapısı kullanımı yöntemi ARM mimarisini kullanılan bir platform üzerinde gerçeklenmiştir. SIMD ve standart uygulamaların performansı, başarımı ve farklı ikili imge dönüşümlerinin (LBP, MF-1BT, IGCBP-1BT) maliyetleri verilecektir.

Deneylerde Şekil 3.1'de görülen Beagleboard topluluğunun BeagleBone Black geliştirme ortamı kullanılmıştır. Cihaz üzerinde AM335x 1GHz ARM® Cortex-A8 işlemci vardır.



Şekil 3.1. Beaglebone black

ARM işlemciler üzerinde kullanılan genel amaçlı 128 bitlik SIMD motoru NEON olarak adlandırılmaktadır. Kullanılan saklayıcılar aynı veri tipinden (8-bit, 16-bit, 32-bit, 64-bit) vektörler olarak tanımlanmaktadır. Şekil 3.2' ARM Cortex A8 mimarisi görünmektedir [39].



Şekil 3.2. ARM cortex-a8 mimarisi

Bu çalışmada deneysel sonuçlar elde edilirken $N=128$ olarak seçilerek imge merkezinden alınan 128×128 piksellik blok bir önceki çerçevede aranmaktadır. Arama sınırı $s=16$ olarak belirlenmiştir. Bu durumda önerilen SIMD iyileştirmesi öncesinde arama aralığı boyunca yani $(2\times16+1)\times(2\times16+1)=1089$ aday nokta için 128×128 adet XOR işlemi yapılmaktadır. SIMD yapılan bu tekrarlı ve aynı komut olan XOR işlemini 2.4.1 bölümünde anlatılan veri yerleştirme ve kaydırma sonrasında azaltmaktadır. SIMD veri uyumlaması için 33 adet yatay düzlemde kaydırma ve dolayısı ile 1 byte yerleştirme işlemi yapılmaktadır. Bu yerleştirme işleminin yapılan deneyler sonucu fazla işlem gücü gerektirmediği gözlemlenmiştir. Byte yerleştirme sonucunda 64 bitlik SIMD saklayıcılarının kullanılması sonucu 128×128 adet XOR işlemi $128\times\left((128/8)/8\right)$ yani 128×2 adete düşmektedir. Teorikte SIMD sonrası NNMP hesabının 64 kat hızlanması gerekirken deneysel sonuçlarda bu seviyeye çıkılamamıştır. Bu farklılığın nedenleri olarak SIMD vektör saklayıcılarına yükleme işlemin ekstra bir maliyet olması ve SIMD komutlarının bazı durumlarda birden fazla çevirim (cycle) gerektirmesi verilebilir.

İkili bütünsel hareket kestirimin başarımı Bölüm 2.2'de değinildiği gibi RMSE değerlerinden ölçülmektedir. SIMD yaklaşımı sadece mevcut yöntemleri hızlandırmak için önerilmiştir. RMSE değerinde herhangi bir değişime sebep olmamaktadır. Deney sonuçlarda aynı RMSE değerlerinin hesaplandığı kontrol edilmiştir. Farklı bütünsel hareket algoritmalarının başarımlarının karşılaştırmaları için [35]'deki çalışma incelenebilir.

Deneylerde farklı çerçeve boyutu ve çerçeve sayısına sahip "Car", "Walking", "Zoom", "Footpath", "Baseball", "Man", "Garden" ve "Table" dizileri kullanılmıştır. Seçilen diziler standart ve SIMD yöntemleri ile gerçek zamanlı ARM işlemcisi üzerinde C yazılımı tabanlı gerçeklenmiştir. Ayrıca farklı ikili imge dönüşümü temelli bütünsel hareket kestirim yöntemleri de uygulamaya dâhil edilmiştir. Böylece ikili imge dönüşümünün maliyetleri deneysel sonuçlar ile gösterilmiştir.

Tablo 3.1'de LBP, MF_1BT ve IGCBP-1BT dönüşümlerinin standart ve SIMD uygulaması sonucu performans değerleri verilmiştir.

Tablo 3.1. SIMD ve standart uygulamanın farklı hareket kestirim yöntemleri üzerinde performans değerleri

		LBP		MF_1BT		IGCBP-1BT	
		Standart	**SIMD**	**Standart**	**SIMD**	**Standart**	**SIMD**
Car 512x640 Çerçeve Sayısı:[98]	**RMSE**	0,08		0,08		0,10	
	İkili Dönüşüm Süresi (sn.)	0,78	0,78	0,84	0,84	0,27	0,27
	Uyumlama Süresi (sn.)	13,68	0,85	14,12	0,85	13,66	0,83
	Toplam Süre (sn.)	15,01	2,18	15,51	2,24	14,48	1,65
	Saniyede İşlenen Çerçeve Sayısı	6,53	44,95	6,32	43,75	6,77	59,39
Walking 430x854 Çerçeve Sayısı:[100]	**RMSE**	0,08		0,07		0,11	
	İkili Dönüşüm Süresi (sn.)	0,81	0,81	0,82	0,82	0,24	0,24
	Uyumlama Süresi (sn.)	14,50	0,91	14,92	0,96	14,19	0,90
	Toplam Süre (sn.)	15,90	2,31	16,33	2,37	15,02	1,73
	Saniyede İşlenen Çerçeve Sayısı	6,29	43,29	6,12	42,19	6,66	57,80
Zoom 430x854 Çerçeve Sayısı:[100]	**RMSE**	0,33		0,45		0,18	
	İkili Dönüşüm Süresi (sn.)	0,80	0,80	0,88	0,88	0,25	0,25
	Uyumlama Süresi (sn.)	14,61	0,93	14,83	0,90	14,30	0,95
	Toplam Süre (sn.)	16,00	2,32	16,30	2,37	15,14	1,79
	Saniyede İşlenen Çerçeve Sayısı	6,25	43,10	6,13	42,19	6,61	55,87
footpath 480x864 Çerçeve Sayısı:[100]	**RMSE**	0,06		0,06		0,10	
	İkili Dönüşüm Süresi (sn.)	0,83	0,83	0,92	0,92	0,30	0,30
	Uyumlama Süresi (sn.)	14,68	0,94	14,98	0,86	14,68	0,78
	Toplam Süre (sn.)	16,23	2,49	16,62	2,50	15,70	1,80
	Saniyede İşlenen Çerçeve Sayısı	6,16	40,16	6,02	40,00	6,37	55,56
baseball 410x844 Çerçeve Sayısı:[99]	**RMSE**	0,05		0,05		0,08	
	İkili Dönüşüm Süresi (sn.)	0,73	0,73	0,83	0,83	0,27	0,27
	Uyumlama Süresi (sn.)	14,88	0,88	14,97	0,84	14,73	0,80
	Toplam Süre (sn.)	16,22	2,22	16,41	2,28	15,61	1,68
	Saniyede İşlenen Çerçeve Sayısı	6,10	44,59	6,03	43,42	6,34	58,93
man 410x844 Çerçeve Sayısı:[199]	**RMSE**	0,11		0,10		0,12	
	İkili Dönüşüm Süresi (sn.)	1,53	1,53	1,66	1,66	0,55	0,55
	Uyumlama Süresi (sn.)	29,79	1,59	30,88	1,55	28,31	1,49
	Toplam Süre (sn.)	32,57	4,37	33,79	4,46	30,11	3,29
	Saniyede İşlenen Çerçeve Sayısı	6,11	45,54	5,89	44,62	6,61	60,49
garden 360x640 Çerçeve Sayısı:[199]	**RMSE**	0,02		0,21		0,19	
	İkili Dönüşüm Süresi (sn.)	1,33	1,33	1,49	1,49	0,44	0,44
	Uyumlama Süresi (sn.)	30,54	1,51	29,79	1,45	27,88	1,48
	Toplam Süre (sn.)	32,80	3,77	32,21	3,87	29,25	2,85
	Saniyede İşlenen Çerçeve Sayısı	6,07	52,79	6,18	51,42	6,80	69,82
table 360x640 Çerçeve Sayısı:[199]	**RMSE**	0.14		0.14		0.14	
	İkili Dönüşüm Süresi (sn.)	1.32	1.32	1.43	1.43	0.46	0.46
	Uyumlama Süresi (sn.)	30.72	1.46	31.44	1.66	26.84	1.43
	Toplam Süre (sn.)	32.97	3.71	33.80	4.02	28.23	2.82
	Saniyede İşlenen Çerçeve Sayısı	6.04	53.64	5.89	49.50	7.05	70.57
Tüm Çerçeveler için Toplam Sonuçlar Çerçeve Sayısı:[1094]	**Topl. İkili Dönüşüm Süresi (sn.)**	8,13	8,13	8,87	9,87	2,78	2,78
	Toplam Uyumlama Süresi (sn.)	163,40	9,07	165,93	9,07	154,59	8,66
	Toplam Süre (sn.)	177,70	23,37	180,97	24,11	163,54	17,61
	Saniyede İşlenen Ortalama Çerçeve Sayısı	6,16	46,81	6,05	45,38	6,69	62,12

İlk olarak LBP bütünsel hareket kestirim yöntemi standart olarak gerçeklenmiştir. Tablo 3.1'de "table" örneği üzerinden elde edilen sonuçlar incelendiğinde 199 çerçeve için toplam sürenin 32,97 saniye olduğu ölçülmüştür. Uygulama içindeki işlemler analiz edildiğinde bütün blok piksel değerlerinin XOR'a işleminden geçirildiği NNMP uyumlama süresi 30,72 saniye yani toplam sürenin %93'ünü kapsadığı gözlemlenmiştir. Tabloda belirtilen uyumlama süresi bir çerçeve için değil dizideki bütün çerçeveleri içermektedir. Bunun yanında ikili dönüşüm süresi 1,32 saniye olarak ölçülmüş ve uyumlama süresi ile kıyaslandığında toplam sürenin sadece %4'ü olduğu gözlemlenmiştir. Standart yöntem ile çerçeve hızı yaklaşık 6 fps olarak ölçülmüştür. Bu sebep ile SIMD yaklaşımı NNMP adımına uygulanmıştır.

SIMD dönüşümü sonrası bütün dizi için toplam hareket kestirim zamanı 3,72 saniye düşmüş ve uyumlama süresi 1,46 olarak büyük ölçüde hızlandırılmıştır. SIMD iyileştirmesinin yaklaşık olarak 21 kat uyumlama süresini hızlandırdığı gözlemlenmiştir. SIMD hızlandırma önermesi sonrası saniyede 40-50 çerçeve hızına çıkıldığı görülmektedir. Bunun yanı sıra uyumlama süresi ve ikili dönüşüm sürelerinin toplam süre bazında yaklaşık olarak aynı zamanı aldığı gözlemlenmiştir. Uyumlama süresi toplam sürenin %39 olur iken ikili dönüşüm süresi %35 olarak ölçülmüştür. Bu durumda standart uygulamada önemsiz gibi görünen ikili dönüşüm süresi SIMD yönteminin gerçeklenmesi sonunda toplam sürenin büyük bir kısmını kapsamaktadır. Farklı ikili dönüşümler gerçeklenerek ikili dönüşüm maliyetinin etkileri araştırılmıştır.

MF1-BT yönteminin ikili dönüşüm süresinde iyileştirme yapmadığı gözlemlenmiştir. Interlaced Gray-coded bit-plane dönüşümünün ikili dönüşüm süresini 3 kat azalttığı deneysel sonuçlar ile gözlemlenmiştir. İkili dönüşüm süresi LBP yönteminde 1,32 saniye ve MF-1BT yönteminde 1,43 saniye ölçülür iken Interlaced Gray-coded bit-plane yönteminde 0,46 olarak ölçülmüştür.

Diğer bütün çerçeve dizileri ele alındığında SIMD yaklaşımının uyumlama süresini yaklaşık 20 kat arttırdığı ve yaklaşık ikili dönüşüm iyileştirmesi yapılmadan önce saniyede 40 çerçeve sayısına erişildiği gösterilmiştir. Interlaced Gray-coded bit-plane seçimi sonrası ikili dönüşüm maliyeti azalmış ve yaklaşık saniyede 50 çerçeve oranına erişildiği gözlemlenmiştir.

SONUÇLAR VE ÖNERİLER

Bu tez kapsamında LBP, MF-1BT ve IGCBP-1BT ikili bütünsel hareket kestirimi yöntemlerinin SIMD komutlarının kullanımı ile ARM tabanlı bir gömülü bir platformda gerçek zamanlı olarak gerçeklenmiştir. Deneysel sonuçlar bütünsel hareket kestiriminin performans iyileştirmelerinin sadece donanımsal tabanlı önermeler ile değil SIMD motoru içeren düşük güce sahip mobil işlemciler üzerinde yazılımsal olarak da gerçeklenebileceğini göstermiştir. SIMD iyileştirmeleri sonucunda ikili dönüşüm maliyetlerinin toplam maliyette öne çıktığını ve IGCBP-1BT yönteminin bu maliyeti en aza indirdiği gösterilmiştir. SIMD motorunun kullanımı ile işlemci hesap gücü görece düşük olduğu mobil cihazlarda bile işlem karmaşıklığı yüksek olan tam arama temelli bir düşük bit derinlikli hareket kestirim algoritmasının başarılı bir şekilde saniyede 40-50 çerçeve hızında çalıştırılabileceği gösterilmiştir.

KAYNAKLAR

[1] Qian X., Global Motion Estimation and Its Applications, Video Compression, Dr. Amal Punchihewa (Ed.), *InTech*, DOI: 10.5772/34166.

[2] ISO/IEC JTC1/SC29/WG11 14496-2, Mpeg-4 coding of moving pictures and audio, *ISO/IEC (International Organization for Standardization/International Electrotechnical Commission*), 1998.

[3] Aggarwal J. K., Nandhahumar N., On the computation of motion from sequences of images - a review. *Proceedings of The IEEE*, 1988, **76**,917-935.

[4] Stiller C., Konrad J., Estimating motion in image sequences., *IEEE Signal Processing Magazine*, 1999, **16**,70-91.

[5] Rawat P., Singhai J., Review of Motion Estimation and Video Stabilization techniques For hand held mobile video, *Signal & Image Processing: An International Journal*, 2011, **2**(2), 159.

[6] http://www.gyroscope.com/d.asp?product=GYROSTABILIZER (Ziyaret tarihi: 20 Ocak 2015).

[7] http://www.digikey.it/Web%20Export/Supplier%20Content/invensense1428/pdf/invensense-image-stabilization-technology.pdf?redirected=1 (Ziyaret tarihi: 20 Ocak 2015).

[8] Wang Y., Ostermann J., Zhang Y.-Q., *Video Processing and Communication* 1st ed, Prentice Hall, New Jersey, 2002.

[9] ITU-T Recs H.264, advanced video coding for generic audiovisual services. *International Telecommunication Union*, Geneva, 2003.

[10] Tian T., Tomasi C., Heeger D., Comparison of approaches to egomotion computation, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, USA, 18-20 June 1998.

[11] Crawford A., Denman H., Kelly F., Pitie F., Kokaram A., Gradient based dominant motion estimation with integral projections for real time video stabilisation, *IEEEInternational Conference on Image Processing*, Singapore, 24-27 October 2004.

[12] https://classes.soe.ucsc.edu/ee264/Fall11/LecturePDF/14MotionEstimationI.pdf (Ziyaret tarihi: 20 Ocak 2015).

[13] Kokaram A., Delacourt P., A new global motion estimation algorithm and its application to retrieval in sports events, *IEEE International Workshop on Multimedia Signal Processing*, Cannes, France, 3-5 October 2001.

[14] Smolic A., Hoeynck M., Ohm J. R., Low-complexity global motion estimation from pframe motion vectors for mpeg-7 applications, *Proc. Of IEEE International Conference on Image Proceesing*, Vancouver, BC, Canada, 10-13 Sepember 2000.

[15] Dufaux F., Konrad J., Efficient, robust, and fast global motion estimation for video coding, *IEEE Trans. Image Process*., 2000, **9**(3), 497–501.

[16] Chung Y., He Z., Reliability Analysis for Global Motion Estimation, *IEEE Signal Processing Letters*, 2009, **16**(11), 980–997.

[17] Dufaux F., Konrad J., Efficient, robust, and fast global motion estimation for video coding, *IEEE Trans. Image Process*., 2000, **9**(3), 497–501.

[18] Keller Y., Averbuch A., Fast gradient methods based on global motion estimation for video compression, *IEEE Trans. Circuits Syst.Video Technol.*, 2003, **13**(4), 300–309.

[19] Barron J. L., Fleet D. J., Beauchemin S. S., Performance of optical flow techniques, *Int. J. Comput. Vis.*, 1994, **12**(1), 43–77.

[20] Okade M., A Novel Motion Vector Outlier Removal Technique based on Adaptive Weighted Vector Median Filtering for Global Motion Estimation, *IEEE Annual India Conference (INDICON)*, Mumbai, India, 13-15 December 2013.

[21] Çimtay Y., Doğrusal Olmayan Hareket Kestirimi İle Video Çerçeve Ara Değerlemesi, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eşkişehir, 2012, 320765.

[22] Dante A., Brookes M., Precise real-time outlier removal from motion vector fields for 3d reconstruction, *IEEE The International Conference on Image Processing ICIP*, Barcelona, Spain, 14-17 September 2003.

[23] Koga T., Iinuma K., Hirano A., Iijima Y., Ishiguro T., Motion- Compensated Interframe Coding for Video Conferencing, *Proceedings of the National Telecommunications Conference*, New Orleans, USA, 1-5 November 1981.

[24] Jain J. R., Jain A. K., Displacement Measurement and Its Application in Interframe Image Coding, *IEEE Trans. Communication*, 1981, **29**(12), 1799–1808.

[25] Li R., Zeng B., Liou M. L., A New Three-Step Search Algorithm for Block Motion Estimation, *IEEE Trans. on Circuits and Systems for Video Technology*, 1994, **4**(4), 438–442.

[26] Po L., Ma W., A novel four-step search algorithm for fast block motion estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 1996, **6**(3), 313-317.

[27] Boyacı F., Video dizilerinde kenar bulma algoritması ile öbek eşleme yöntemine dayalı hareket kestirimi,Yüksek Lisans Tezi, Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2010, 295611.

[28] Ko S. J., Lee S. H., Lee K. H., Digital image stabilizing algorithms based on bit-plane matching, *IEEE Trans. on Consumer Electron*., 1998, **44**(3), 617 - 622.

[29] Ko S. J., Lee S. H., Jeon S. W., Kang E. S., Fast digital image stabilizer based on Gray-coded bit-plane matching, *IEEE Trans. Consumer Electron.*, 1999, **45** (3), 598–603.

[30] Natarajan B., Bhaskaran V., Konstantinides K., Low-Complexity Block-Based Motion Estimation via One-Bit Transforms, *IEEE Trans. on Circuits and Systems for Video Technology*, 1997, **7**(4), 702–706.

[31] Ertürk S., Multiplication-Free One-Bit Transform for Low-Complexity Block-Based Motion Estimation, *IEEE Signal Processing Letters*, 2007, **14**(2), 109-112.

[32] Urhan O., Ertürk S., Constrained One-Bit Transform for Low Complexity Block Motion Estimation, *IEEE Trans. Circuit Syst. Video Technol*., 2007, **17**(4): 478-482.

[33] Yeni A.A., Erturk S., Fast digital image stabilization using one bit transform based sub-image motion estimation, *IEEE Trans Consumer Electron.*, 2005, **51** (3), 917–921.

[34] Urhan O., Erturk S., Single sub-image matching based low complexity motion estimation for digital image stabilization using constrained one-bit transform, *IEEE Trans. Consumer Electron.*, 2006, **52**(4), 1275–1279.

[35] Kır B., Kurt M., Urhan O., Local binary pattern based fast digital image stabilization, *IEEE Signal Processing Lett.*, 2015, **22**(3), 341-345.

[36] Kuo T. Y., Wang C. H., Fast local motion estimation and robust global motion decision for digital image stabilization, *Proc. Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, Harbin, China, 15-17 August 2008.

[37] Çelebi A., Urhan O., Hamzaoğlu I., Ertürk S., Efficient Hardware Implementations of Low Bit Depth Motion Estimation Algorithms, *IEEE Signal Process. Letts.*, 2009, **16**(6), 513-516.

[38] Akın A., Doğan Y., Hamzaoğlu I., High performance hardware architectures for one-bit transform based motion estimation, *IEEE Trans. Consumer Electron*., 2009, **55**(2), 1144–1152.

[39] http://www.arm.com/products/processors/cortex-a/cortex-a8.php (Ziyaret tarihi: 20 Ocak 2015).

KİŞİSEL YAYINLAR ve ESERLER

[1] **Turgut B.**, Urhan O., İkili Bütünsel Hareket Kestiriminin ARM İşlemci Üzerinde SIMD Altyapısı ile Gerçeklemesi, *Gömülü Sistemler ve Uygulamaları Sempozyumu (GömSis 2014*), İstanbul, Türkiye, 5-6 Aralık 2014.

[2] **Turgut B.**, Urhan O., İkili İmgeler Üzerinde Bütünsel Hareket Kestiriminin ARM İşlemci ile Gerçek Zamanlı Gerçeklemesi, *22. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU2014*), Trabzon, Türkiye, 23-25 Nisan 2014.

ÖZGEÇMİŞ

1981 yılında Erzurum'da doğan Boğaç Turgut, ilköğrenimini Atatürk İlköğretim Okulu'nda, orta öğrenimini Erzurum Anadolu Lisesi ve lise öğrenimini ise Erzurum İbrahim Hakkı Fen Lisesi'nde tamamladı. Lisans derecesini 2005 yılında Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü'nden aldı. Askeri hizmet sonrasında 2006 yılında başladığı özel çalışma hayatında Netaş, Tübitak UEKAE şirketlerinin ARGE bölümlerinde donanım ve yazılım geliştirici olarak çalıştı. 2011 yılından beri Siemens A.Ş ARGE bölümde Proje yöneticisi görevini sürdürmektedir.

2010 yılından beri Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine devam etmektedir.