

Düşük işlem yüküne sahip hareket kestirimi için tümlev imge temelli ikileştirme

Integral image based binarization for low-complexity motion estimation

Aysun TAŞYAPI ÇELEBİ^{1*}, Oğuzhan URHAN²

^{1,2}Elektronik ve Haberleşme Mühendisliği Bölümü, Mühendislik Fakültesi, Kocaeli Üniversitesi, Kocaeli, Türkiye.
aysun.tasyapi@kocaeli.edu.tr, urhano@kocaeli.edu.tr

Geliş Tarihi/Received: 27.09.2017, Kabul Tarihi/Accepted: 27.02.2018

doi: 10.5505/pajes.2018.48303

* Yazışılan yazar/Corresponding author

Araştırma Makalesi/Research Article

Öz

Günümüzde yüksek çözünürlüklü televizyonlar, kameralar, akıllı telefonların kullanımı ile birlikte yüksek çözünürlüklü video uygulamalarına talep duyulmaktadır. Bu cihazlardaki güç tüketimi ve sınırlı hafıza gibi kısıtlardan dolayı da düşük işlem yüküne sahip video kodlama yöntemlerine ihtiyaç artmaktadır. Video kodlama standartlarında halen en fazla işlem yükü hareket kestirimi kısmındadır. Bu çalışmada düşük işlem yüküne sahip, düşük bit derinliği gösterimi temelli bir hareket kestirimi yöntemi önerilmektedir. Bu yaklaşımda video çerçeveleri tümlev imge kullanılarak etkin bir şekilde ikileştirilmekte ve video çerçevelerinin iki bit ile gösterimi elde edilmektedir. Elde edilen ikili çerçeveler üzerinden geleneksel mutlak farklar toplamı (SAD) yerine donanıma daha uygun olan dışaran veya (EX-OR) operasyonu kullanılarak uyumlama işlemi yapılmaktadır. Hareket kestiriminde ikileştirme işlemi gerçekleştirirken tümlev imge kullanılması ilk kez bu çalışmada önerilmektedir. Önerilen yöntem, literatürde mevcut olan 1-bit dönüşüm (1BT) temelli yaklaşımlara kıyasla hareket kestirim doğruluğunu geliştirirken iki-bit dönüşüm temelli yaklaşımların başarısı ile hemen hemen aynı seviyede olmaktadır. Bunun yanında özellikle ikileştirme aşamasında bu yöntemlerin işlem yükünü azaltmaktadır.

Anahtar kelimeler: Tümlev imge, İkileştirme, Düşük bit derinliği, Hareket kestirimi

Abstract

Today, high resolution video applications are demanded with the use of high resolution televisions, cameras, smart phones. The requirement for low processing load video coding methods increase due to constraints such as power consumption and limited memory in these devices. In video coding standards, most of the processing load still originates from the motion estimation part. In this study, a low bit-depth representation based motion estimation method that has low computational load is proposed. In this approach, video frames are binarized efficiently by using integral image and the representation of video frames in terms of two bits is performed. Matching operation is carried out on these binary image frames using hardware-friendly EX-OR operation instead of conventional SAD (Sum of Absolute Difference). This study is the first attempt of using the integral image for the binarization process in the motion estimation. While the proposed method improves the motion estimation accuracy compared to the 1BT based approaches available in the literature, it provides similar motion estimation performance with the two-bit depth based approaches. Additionally, it reduces the processing load of these methods, especially during the binarization phase.

Keywords: Integral image, Binarization, Low bit-depth, Motion estimation

1 Giriş

Günümüzde teknolojinin hızla ilerlemesi ile beraber video kaydetme, video aktarımı, gerçek zamanlı konuşma gibi uygulamaların olduğu cihazların kullanımı önemli miktarda artış göstermiştir. Bu cihazlarda özellikle yüksek çözünürlüklü videoların kullanımının yaygınlaşmaya başlaması ile cihazların sınırlı kalıcı bellek, batarya, işlem gücü ve bant genişliğine sahip olması nedeniyle daha yüksek sıkıştırma başarımına sahip kodlama yöntemlerinin kullanılması kaçınılmaz hale gelmiştir. Ardışık video çerçevelerindeki zamansal artıklığı ortadan kaldıran hareket kestirimi (HK) aşaması halen video kodlayıcılardaki hesapsal olarak en yüksek karmaşıklığa sahip olan kısımdır. Standart bir video kodlayıcıda HK bloğu, kodlama süresinin yaklaşık %90'undan fazlasını harcamaktadır [1]. HK işleminin kodlayıcıya getirdiği yükün artmasıyla birlikte gerçek-zamanlı uygulamalar için daha düşük işlem yüküne sahip olan hareket kestirimi algoritmalarının kullanılması kaçınılmaz olmuştur.

Video kodlayıcılarda HK için geleneksel olarak mutlak farklar toplamı (SAD-sum of absolute difference) ölçütü kullanılarak blok uyumlama (block matching) yaklaşımı yöntemler kullanılmaktadır. Blok uyumlama yaklaşımında, video çerçeveleri birbirleriyle örtüşmeyen bloklara bölünmekte, sonrasında ise her bir blok referans çerçevede önceden

belirlenmiş bir arama penceresi içinde tüm aday noktalarda en iyi blok uyumunu verecek şekilde aranmaktadır. Bu yaklaşımda aranacak blok, arama penceresi içerisinde bütün olası noktalarda arama yaptığı için yüksek bir işlem yüküne sahiptir. SAD uyumlama ölçütü kullanan tam arama temelli bu yöntem işlem yükü ve güç kısıtlamalarına sahip mobil cihazlarda kullanılması pek de mümkün değildir. Bu sebeple literatürde tam arama yaklaşımına göre daha düşük işlem yüküne ve kaynak kullanımına sahip, donanımsal gerçeklemlere uygun farklı HK yaklaşımları önerilmiştir.

Bu yaklaşımlardan birincisi arama noktası sayısını azaltmaya dayalı seyrek arama yöntemleridir. Bu gruba dahil olan yaklaşımlar aranan bloğu referans çerçevenin bütün aday noktalarında aramak yerine sadece önceden belirlenmiş bazı seyrek arama noktalarında arama mantığını kullanmaktadır. Örneğin üç-adımlı arama (Three step-search 3SS) [2], yeni üç adımlı arama (new three-step search N3SS) [3], dört adımlı arama (four-step search 4SS) [4], elmas biçimli arama (diamond search) [5] ve altıgenel arama (hexagonal search) [6] bu gruba giren yaklaşımlardır.

İkinci grup yaklaşım da ise uyumlama ölçütü hesabında bütün pikseller kullanılmamaktadır. Uyumlama hesabının yapılması sırasında blok içerisinde seçilmiş bazı piksellerin kullanımı ile işlem yükü düşürülmektedir. Bu grupta bulunan [7]-[8]'deki

yaklaşımlar farklı alt-örnekleme oranları ve N-Queen farklı alt-örnekleme örüntülerini kullanmaktadır.

Diğer bir grup yaklaşımda ise kısmi hata araması (partial error search) [9], uyarlamalı arama bölgesi tespiti (adaptive search range) [10] ve ardışık eleme algoritması (successive elimination algorithm) [11] gibi aramayı erken sonlandıran yaklaşımlar yer almaktadır.

Son grup yer düşük işlem yüküne sahip uyumlama ölçütü kullanan video çerçevelerini düşük-bit derinliği ile temsil eden yaklaşımlardan oluşmaktadır. Bu yaklaşımlarda tam arama temelli HK'deki uyumlama ölçütü olarak kullanılan SAD yerine hesapsal olarak daha düşük karmaşıklığa sahip EX-OR işlemi temelli donanımsal ve paralel gerçeklemeye uygun bir uyumlama ölçütünün kullanılmasını hedeflenmektedir. Literatürde düşük bit çözünürlüğünü kullanan yöntemlerden bazıları 1-bit dönüşümü (1BT) [12], çarpmasız 1-bit dönüşümü (MF-1BT) [13], 2-bit dönüşümü (2BT) [14], kısıtlanmış 1-bit dönüşümü (C-1BT) [15] gibi çalışmalardır.

1BT yaklaşımı [12] ile 8-bit çözünürlüğündeki imge çerçeveleri bir-bit derinliğindeki imge çerçevelerine dönüştürülmektedir. Bu işlem, çoklu bant geçiren karakteristikte bir çekirdek (kernel) ile imgeler süzgeçlenerek yapılmaktadır. Süzgeçlenmiş imge çerçevesi ile orijinal imge çerçevesi pikselleri arasında bir karşılaştırma yapılır ve bu karşılaştırma sonucuna göre bir-bit derinliğinde imgeler elde edilmektedir. Giriş imgelerinin ikili (binary) hale getirilmesi sonrasında blok uyumlamasını hesaplamak için EX-OR işlemi kullanılarak uyumsuz nokta sayısı (number of non-matching points-NNMP) adı verilen bir uyumlama ölçütü kullanılmaktadır.

Çarpmasız bir-bit dönüşümü (MF-1BT) [13] temelli HK yaklaşımında ise 1BT yaklaşımının işlem yükünü azaltmak hedeflenmiştir. Bir-bit dönüşümde kullanılan çekirdekte 25 adet "1" bulunmakta ve normalizasyon işlemi sırasında kullanılan katsayısı 2'nin kuvveti şeklinde olmamaktadır. MF-1BT yaklaşımında ise elmas şeklinde bir çekirdek oluşturulmuş ve bu çekirdeğe 16 tane "1" yerleştirilerek normalizasyon katsayısı 2'nin kuvveti olacak şekilde ayarlanmıştır. Kullanılan çekirdekteki bu değişiklik haricinde MF-1BT yaklaşımı 1BT yaklaşımı ile aynıdır.

2BT yaklaşımı HK yönteminde [14] yerel blokların ortalama ve değişinti özellikleri dikkate alınarak ikilileştirme işlemi gerçekleştirilmektedir. [14]'teki yöntemde yerel ortalama ve değişinti değerlerini kullanarak yerel eşik değerleri oluşturmaktadır. Hesaplanan bu eşik değeri kullanılarak giriş imgelerinin iki bitlik gösterimi elde edilmektedir.

[15]'te önerilen kısıtlanmış bir-bit dönüşümü temelli yöntemde (C-1BT) ise 2BT yöntemindeki gibi iki bit-uzayı kullanılmaktadır. Ancak bu yöntemde ilk bit-uzayı MF-1BT'de olduğu gibi oluşturulurken, ikinci bit uzayı uyumlama katılacak pikselleri tespit etmek üzere kısıt maskesi (constraint mask-CM) kullanılarak elde edilmektedir.

[16]'da önerilen iyileştirilmiş 2BT yönteminde uyumlama ölçütünün hesabında yapılan değişiklik ile hareket kestirim başarımının artırılması hedeflenmektedir. [17]'de C-1BT yönteminde önerilen kısıt maskesinin iki bit kullanımını ifade edilmesi önerilmekte ve böylelikle toplamda üç bit uzayı kullanılarak HK başarımının artırılması sağlanmaktadır. [18]'de ikilileştirme aşamasında bulanık mantık kullanımı önerilmektedir.

Bu çalışmada, tümlev imge yaklaşımının kullanıldığı özgün bir ikilileştirme aşaması bulunduran düşük işlem yüküne sahip bir

HK yaklaşımı önerilmektedir. Önerilen yaklaşımda giriş imgelerinin iki-bitlik gösterimi elde edilerek her bit uzayı için NNMP değeri hesaplanmakta ve uyumlama işlemi gerçekleştirilmektedir. Önerilen yöntem iki-bit uzayı kullanılan literatürdeki çalışmalar ile hemen hemen aynı nesnel hareket kestirimi başarımını daha düşük işlem yükü ile sağlamaktadır.

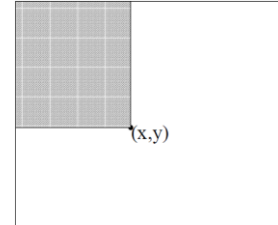
2 Tümlev imge temelli ikilileştirme

Literatürde birçok ikilileştirme yöntemi mevcuttur. Bir imgeyi ikili hale getirmek için global bir eşik değeri kullanarak (örneğin Otsu ile tespit edilmiş) eşikleme yapmak her durumda başarılı sonuçlar vermemektedir. Bunun yerine literatürde birçok yerel uyarlamalı eşikleme yöntemleri önerilmiştir [19]-[21]. Bu yöntemler her bir merkez piksel için $w \times w$ komşuluğunda ortalama değer ve standart sapmaya bakarak bir eşik değeri belirlemektedir. Yalnız bu yöntemler w 'ya bağlı olarak çok fazla iş yüküne sahiptir. Öte yandan [22]'de önerilen yöntem tümlev imge kullanarak gerçek zamanlı uyarlamalı eşikleme yapmaktadır. Bu çalışmada da tümlev imge kullanılarak hızlı bir şekilde ikili imgeler elde edilmektedir.

2.1 Tümlev imge

Tümlev imge [23],[24], bir imgede veya imgedeki bir dikdörtgen alandaki piksel değerlerinin toplamını hesaplamak için kullanılan hızlı ve etkili bir yöntemdir. Her bir pikselin tümlev imge değeri, ilgili pikselin solundaki ve üstündeki piksellerle sınırlanan alan içindeki piksellerin toplam değeridir.

Şekil 1'de tümlev imge hesabının temsili hale gösterilmektedir. Tümlev imgedeki x ve y noktalarındaki değer, giriş imgesinin sol yukarıdaki piksel değerlerinin toplamına eşittir ve bu değer Denklem (1)'de verilen formül ile hesaplanmaktadır.



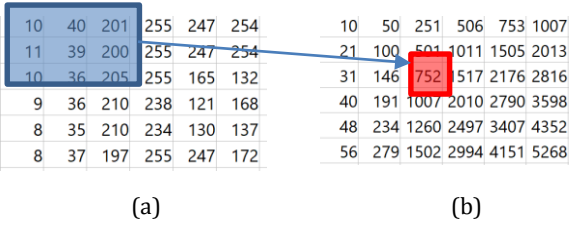
Şekil 1: Tümlev imge oluşturma [24].

$$I_T(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (1)$$

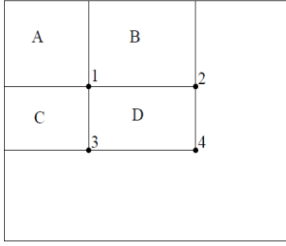
Burada, I giriş imgesini, I_T ise elde edilen tümlev imgeyi ifade eder.

Şekil 2'de gri seviyeli bir imgede küçük bir blok için için ışıklılık değerleri ve bu blok için oluşturulan tümlev imge gösterilmektedir. Giriş imgesi üzerinde ilk 3×3 'lük (mavi renk ile gösterilen) bloğun toplamı, denklem (1)'de de gösterildiği gibi tümlev imge üzerinde $I_T(3,3) = 752$ şeklinde oluşmaktadır.

Bir kez tümlev imge oluşturulduktan sonra giriş imgesi üzerindeki belirli bir bölgenin piksellerin toplamı hızlı ve kolay bir şekilde tümlev imge kullanılarak yapılabilmektedir. Tümlev imge üzerinde sadece 4 referans noktadaki değerlerin kullanılması, istenilen dikdörtgen alanın hesabının yapılması için yeterli olmaktadır. Örneğin Şekil 3'teki tümlev imgede D bölgesinin alanı, tümlev imge üzerindeki 1,2,3,4 No.lu noktalardaki değerler kullanılarak $D=4-3-2+1$ olacak şekilde hesaplanmaktadır.

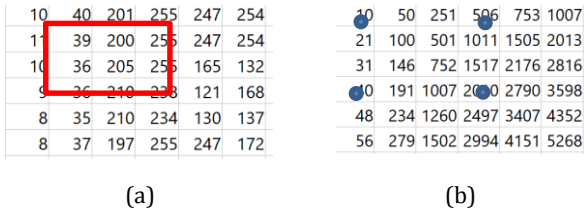


Şekil 2: (a): Giriş imgesinden bir bölge. (b): Oluşturulan tümlev imgeden bir bölge.



Şekil 3: Herhangi bir bölgenin alanının hesaplanmasında izlenen yol hesabı [24].

Şekil 3'teki gösterimin 8-bitlik bir imge üzerinden yorumu Şekil 4'te verilmektedir. Şekil 4'te orijinal imgede herhangi bir dikdörtgen alan içinde kalan piksellerin toplamının integral imge üzerinde nasıl hesaplandığı gösterilmektedir. Şekil 4(a)'da 8-bitlik orijinal imge üzerinde kırmızı dikdörtgen ile çevrilmiş alandaki ışıklılık değerlerinin toplamı Şekil 4(b)'te verilen tümlev imgedeki 4 referans nokta ile bulunmaktadır. Kırmızı dikdörtgen ile belirtilen piksellerin toplam işlemi tümlev imge üzerinden $2010-40-506+10=1474$ işlemi ile hesaplanmaktadır. Giriş imgesi üzerinde seçilen dikdörtgen alan büyüse bile (yani toplanılacak piksel sayısı artsa da) bu işlem tümlev imge üzerinde sadece 4 referans nokta üzerinden 2 çıkartma ve 1 toplama işlemi gerçekleştirilmektedir.



Şekil 4: (a): Giriş imgesinden bir bölge. (b): Oluşturulan tümlev imgeden bir bölge.

Tümlev imge kullanarak yerel ortalama değer hesaplamak ve hesaplanan ortalama değere bağlı olarak video çerçevelerini ikili hale çevirmek, işlem yükü açısından çok etkili bir yoldur. Çünkü bir kez tümlev imge hesaplandığı durumda Denklem (2)'de verilen yerel toplam $s(x, y)$ değerini herhangi bir w pencere boyutunda hesaplamak çok kolaylaşmaktadır. Giriş imgesi üzerinde yerel toplam değeri tümlev imge kullanılarak Denklem (2)'deki gibi hesaplanabilmektedir.

$$\begin{aligned}
 s(x, y) &= I_T(x + d - 1, y + d - 1) \\
 &\quad - I_T(x + d - 1, y - d - 1) \\
 &\quad - I_T(x - d - 1, y + d - 1) \\
 &\quad + I_T(x - d - 1, y - d - 1)
 \end{aligned} \quad (2)$$

$$d = \lfloor \frac{w}{2} \rfloor$$

Burada, $\lfloor \cdot \rfloor$ sembolü elde edilen bölümü üste yuvarla anlamında kullanılmaktadır. Bu denklem Şekil 4'te de gösterildiği gibi tümlev imgede dört noktadan faydalanılarak, giriş imgesinde herhangi bir pikselin etrafında $w \times w$ aralığındaki piksellerin ışıklılık değerlerinin toplamının elde edilmesini ifade eder. Örneğin Şekil 4'te verilen giriş imgesi üzerindeki $I(3,3)$ noktası etrafında 3×3 'lük alan için yerel toplam hesaplanırken $d = 2$ olacak şekilde seçilir ve Denklem (2) kullanılarak tümlev imge üzerinden bu toplam;

$$s(x, y) = I_T(4,4) - I_T(4,1) - I_T(1,4) + I_T(1,1) = 2010 - 40 - 506 + 10 \text{ şeklinde elde edilir.}$$

Yerel ortalama değeri (x, y) noktasında hesaplandıktan sonra, giriş imgesinde (x, y) noktası etrafındaki $w \times w$ aralığındaki piksellerin ortalama değeri olan yerel aritmetik ortalama $m(x, y)$ değeri, Denklem (3)'teki gibi hesaplanmaktadır. Bu şekilde yerel ortalama değeri, tümlev imge kullanılarak pencere boyutuna bağlı olmaksızın çok etkili ve hızlı bir şekilde hesaplanmış olmaktadır.

$$m(x, y) = \frac{s(x, y)}{w^2} \quad (3)$$

Hızlı bir şekilde hesaplanan yerel ortalama değeri kullanılarak her bir piksel için bir karşılaştırma yapılarak etkin bir şekilde ikilileştirme işlemi gerçekleştirilebilir.

2.2 Önerilen Yaklaşım

Önerilen yaklaşımda giriş çerçevesi için tümlev imge bir kez hesaplandıktan sonra w_1 pencere boyutuna bağlı olarak Denklem 3'deki gibi yerel ortalama değer hesaplanmaktadır. Hesaplanan ortalama değer kullanılarak da aşağıdaki denklemde gösterildiği gibi ilgili piksel için bir karşılaştırılma yapılarak ikilileştirme işlemi gerçekleştirilmektedir.

$$B_1(x, y) = \begin{cases} 1 & , I(x, y) - m_1(x, y) \geq T_1 \\ 0 & , \text{diğer} \end{cases} \quad (4)$$

Burada, B_1 birinci bit uzayını, I giriş imge çerçevesini, m_1 tümlev imgeden faydalanarak hızlı bir şekilde hesaplanan yerel ortalama değeri ve T_1 deneysel sonuçla belirlenmiş eşik değerini göstermektedir. Bir pikselin değeri etrafındaki w_1 pencere boyutundaki alan için hesaplanan yerel ortalama değerden T_1 kadar büyük ise bu piksel için kenar bilgisidir veya detay içeren bir pikseldir kararı verilmektedir. Bu karar ile de uyumlama aşamasında kullanılacak birinci bit uzayı oluşturulmaktadır. Burada, T_1 değerinin kullanılmasının sebebi gürültüye karşı olan hassasiyeti azaltılmaktadır. Karşılaştırma işlemi, pikselin ışıklılık değeri ile doğrudan yerel ortalama değeri kullanılarak yapılırsa, elde edilen ikili imgede birçok yerde piksel kenar gibi algılanıp ciddi miktarda gürültü oluşacaktır.

Elde edilen birinci bit uzayı ile komşuluk ilişkileri dikkate alınarak ince ayrıntılar ciddi şekilde çıkartılmaktadır. Bu bilgiye ek olarak imgedeki daha kaba ayrıntıları çıkartmak için ikinci bit uzayı oluşturulmaktadır. İkinci bit uzayı oluşturulurken w_1 pencere boyutundan daha küçük w_2 pencere boyutunda yerel ortalama değer hesaplanmaktadır. Denklem 5'de verildiği gibi w_1 ve w_2 pencere boyutundan elde edilen iki ortalama değer karşılaştırılması ile de ikinci bit uzayına karar verilmektedir. Buradaki motivasyon ilgili piksel için büyük pencere boyutunda hesaplanan ortalama değeri küçük bir pencere boyutu için hesaplanan ortalama değerinden belli bir eşik değeri kadar büyük ise bu piksel kesinlikle kenar bilgisi içermekte ve HK için detay içeren bir bilgiye sahip olmaktadır.

Burada T_2 gibi bir eşik kullanılmasının sebebi de birinci bit uzayı için anlatıldığı gibi gürültüye olan hassasiyeti azaltmak içindir. Birinci bit uzayı elde edilirken gürültü de kenar bilgisi şeklinde algılanabilirken ikinci bit uzayında bu şekilde ikinci bir kontrol yapılarak çok yüksek ihtimal kenar olan pikseller tespit edilmiş bulunmaktadır.

$$B_2(x, y) = \begin{cases} 1 & , \text{abs}(m_1(x, y) - m_2(x, y)) \geq T_2 \\ 0 & , \text{diğer} \end{cases} \quad (5)$$

Burada, B_2 ikinci bit uzayını, m_2 tümlev imgeden faydalanarak w_2 pencere boyutuna bağlı olarak hızlı bir şekilde hesaplanan yerel ortalama değeri ve T_2 deneysel sonuçla belirlenmiş eşik değerini göstermektedir.

Şekil 5'te bu yaklaşım kullanılarak elde edilen bit düzlemleri gösterilmektedir. Bu şekilden de görüldüğü gibi birinci bit düzlemi ince ayrıntıları gösterirken ikinci bit düzlemi belirgin özelliklerini çıkartmaktadır. Bu iki bit düzleminin birlikte kullanılmasıyla beraber HK başarımının artırılması hedeflenmektedir. Bu şekilde sonuçlar elde edilirken $T_1=5$ ve $T_2=11$ olarak seçilmiştir.



(a): Orijinal çerçeve.



(b): Önerilen yöntem ile edilmiş birinci bit-düzlemi.



(c): Önerilen yöntem ile edilmiş ikinci bit-düzlemi.

Şekil 5: Foreman dizisinin bir çerçevesi için önerilen yöntem ile elde edilen bit düzlemleri.

İkileştirilme işlemi gerçekleştirildikten sonra ikili imgeler üzerinde bir bloğun hareket vektörlerini bulmak için literatürde uyumsuz nokta sayısı (Non-Matching Points (NNMP)) isimli bir metrik hesaplanmakta ve denklemi aşağıdaki gibi verilmektedir.

$$NNMP(c_x, c_y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B^t(i, j) \oplus B^{t-1}(i+c_x, j+c_y)\} \quad (6)$$

Burada, $-s \leq c_x, c_y \leq s-1$ aralığında olup, (c_x, c_y) aday hareket vektörlerini göstermektedir. N ve s sırasıyla blok boyutunu ve arama aralığını göstermektedir. $B^t(i, j)$ ikili imgeyi temsil etmektedir. İkili imgelerde referans ve o anki çerçeve kendi içinde EX-OR'lanıp sonuçlar toplanarak NNMP değeri hesaplanmaktadır. Elde edilen minimum NNMP değerini veren (c_x, c_y) noktası o blok için hareket vektörü olarak belirlenmektedir.

Önerilen yaklaşımda tümlev imge kullanarak giriş imgesi 2-bit düzlemde temsil edildikten sonra, bir bloğun hareket vektörlerinin bulunması için aşağıda verilen denklem ile genişletilmiş uyumsuz nokta sayısı (Extended Non-Matching Points (ENNMP)) isimli bir metrik hesaplanmaktadır.

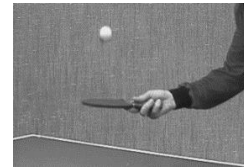
$$ENNMP(c_x, c_y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B_1^t(i, j) \oplus B_1^{t-1}(i+c_x, j+c_y)\} + \{B_2^t(i, j) \oplus B_2^{t-1}(i+c_x, j+c_y)\} \quad (7)$$

$B_1^t(i, j)$ birinci bit düzlemini ve $B_2^t(i, j)$ ise ikinci bit düzlemini temsil etmektedir. Standart NNMP kriteri altında yatan yaklaşımla elde edilen minimum ENNMP değerini veren (c_x, c_y) noktası o blok için hareket vektörü olarak belirlenmektedir.

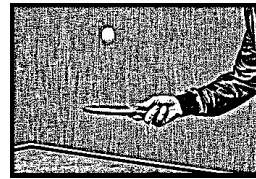
3 Deneysel sonuçlar

Önerilen yöntemin HK başarımı düşük çözünürlüklü, farklı karakteristiğe sahip video dizileri için açık çevrim (open loop) olarak değerlendirilmiştir. Bu amaçla orijinal çerçeve ile kestirilen çerçeve arasında PSNR (Peak Signal to Noise Ratio) değeri hesaplanmıştır.

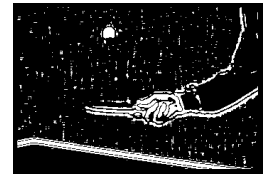
İkileştirme işlemi gerçekleştirilirken tümlev imge elde edildikten sonra birinci bit uzayını elde etmek için kullanılan w_1 pencere boyutu ve karşılaştırma için kullanılan T_1 değeri deneysel olarak belirlenmiş ve en iyi değerler sırasıyla 11 ve 5 olarak tespit edilmiştir. Aynı şekilde ikinci bit uzayı edilirken en iyi sonuçları verdiği için w_2 değeri 5 olarak T_2 değeri de 10 olarak belirlenmiştir. Şekil 6'da Tenis dizisinin bir çerçevesi için bu parametreler kullanılarak elde edilen bit uzayları gösterilmektedir.



(a): Orijinal çerçeve.



(b): Önerilen yöntem ile edilmiş birinci bit-düzlemi.



(c): Önerilen yöntem ile edilmiş ikinci bit-düzlemi.

Şekil 6: Tenis dizisinin bir çerçevesi için önerilen yöntem ile elde edilen bit düzlemleri.

Şekil 7'de Football dizisi için bir referans ve o anki çerçeve gösterilmektedir. O anki çerçeve için referans çerçeveden önerilen çalışma ile elde edilen hareket vektörleri oklar ile işaretlenmiştir.



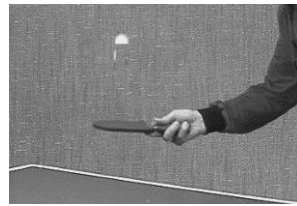
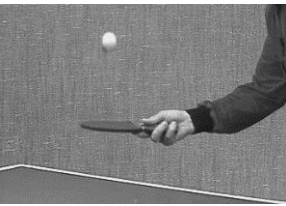
(a)



(b)

Şekil 7: (a): Football dizisinden bir referans çerçeve,
(b): O anki çerçeve ve bulunan hareket vektörleri.

Şekil 8'de önerilen HK yaklaşımı sonucunda bir önceki çerçeveden elde edilen hareket vektörleri kullanılarak geri çatılmış çerçeveler gösterilmektedir. Şekilden görüldüğü gibi kestirim sonucu elde edilen çerçevelerde bozulma çok az olmaktadır. Bu bozulma video kodlayıcının diğer bileşenleri tarafından kodlanmaktadır.



(a): Orijinal çerçeveler.

(b): Önerilen yöntem ile kestirilmiş çerçeveler.

Şekil 8: Foreman ve Tenis dizilerinin bir çerçevesi için önerilen yöntem ile elde edilen sonuçlar.

Kullanılan pencere boyutlarına bağlı olarak yerel ortalamalar hesaplanırken donanım için zorluğu bilinen kayan noktalı (floating point) çarpma işlemi yerine kuvvet indirgeme (strength reduction) yöntemi ile yaklaşık olarak sonucu verecek şekilde kaydırma işlemi gerçekleştirilmiştir. Birinci bit uzayı elde edilirken $w_1=11$ pencere boyutu için ortalama değer hesabında $1/121$ ile çarpma yerine $(1/128)$ değeri ile çarpma işlemine karşılık gelen 1 tane 7-bit kaydırma işlemi yapılmıştır. Aynı şekilde ikinci bit uzayı için $w_2=5$ değeri için ortalama

hesabında $1/25$ ile çarpma yerine $(1/32 + 1/128)$ değeri ile çarpma işlemi karşılık gelen 1 tane 7-bit ve 1 tane 5-bit kaydırma işlemi yapıp sonuçları toplanmıştır. Bu yaklaşımla kayan noktalı çarpma işlemlerinden kaçınılmış ve bütün işlemler tamsayı formda gerçekleştirilerek donanımsal uygulamalar için de etkin bir yöntem geliştirilmiştir.

Önerilen yaklaşımın başarımı Tablo 1'de bulunan düşük işlem yüküne sahip bir-bit ve iki-bitlik diğer yöntemler ile karşılaştırıldığında bir-bitlik yöntemler olan 1BT ve MF-1BT yaklaşımlarından daha iyi PSNR değeri verirken iki-bitlik yöntemlerden 2BT'nin başarımını geçerken C-1BT ile hemen hemen aynı başarımı vermektedir.

Geliştirilen yaklaşımın işlem yükü, diğer düşük-bit derinliğine sahip diğer yaklaşımlar ile beraber Tablo 2'de karşılaştırılmaktadır. Burada piksel başına düşen işlem sayısı dikkate alınmaktadır. Tabloya bakıldığında en az işlem yükü veren HK yaklaşımının önerilen yaklaşım olduğu görülmektedir. Önerilen yaklaşımın işlem yükü detaylı incelendiğinde tümlev imge oluştururken piksel başına 1 toplama işlemi, tümlev imgeden belirli bir alandaki piksellerin toplamını oluştururken bit-uzayı başına 1 toplama 2 çıkartma işleminden iki bit uzayı için 2 toplama 4 çıkartma yapılmaktadır. İlgili bloğun ortalama değerini hesaplamak için birinci bit uzayı için $(1/128)$ değeri ile çarpma işlemine karşılık gelen 1 tane kaydırma, ikinci bit uzayı için $(1/32 + 1/128)$ değeri ile çarpma işlemine karşılık gelen 2 tane kaydırma ve 1 tane toplama işlemi yapılmaktadır. İkili imgeleri oluşturmak için birinci bit uzayını elde ederken $I(x,y) - m_1(x,y) \geq T_1$ denkleminde 1 çıkartma, 1 karşılaştırma işlemi bunun yanında ikinci bit uzayını elde etmek için $(m_1(x,y) - m_2(x,y)) \geq T_2$ işleminden dolayı 1 çıkartma ve 1 karşılaştırma işlemi yapılmaktadır. Önerilen yaklaşımın işlem yükü toplamda incelendiğinde ise ikili imge elde etme aşamasında 4 toplama, 6 çıkartma, 3 kaydırma ve 2 karşılaştırma olmak üzere 15 işlem gerektirdiği görülmektedir.

Diğer düşük bit gösterim temelli yaklaşımlardan piksel başına bir-bit kullanan 1BT ve MF-1BT'nin HK başarımlarının düşük olmasının yanında ikilileştirme aşamasında kullandığı işlem sayısı da daha fazla olmaktadır. 2BT yöntemi piksel başına 2.8125 toplama, 1.0625 çarpma ve 0.0312 karşılaştırmaya ihtiyaç duymaktadır. Buradaki kayan nokta çarpma ise donanım açısından uygun olmayan bir işlem olup gerçekleştirilmesi zordur. C-1BT yaklaşımının ikili imge elde etme aşamasında ise toplam 20 işlem bulunmakta ve önerilen yaklaşım ile karşılaştırıldığında daha fazla işlem yüküne sahip olduğu görülmektedir. Uyumlama aşamasındaki işlem yükü ise önerilen yöntemde ve 2BT, C1-BT yaklaşımlarında benzer seviyededir. Sonuçlardan anlaşılacağı gibi önerilen yaklaşım literatürde donanıma uygun olan düşük bit derinliği kullanarak düşük işlem yüküne sahip olan HK yöntemleri arasında en az işlem yüküne sahip olurken HK başarımı olarak da 1BT, MF-1BT ve 2BT'ye göre daha iyi başarımlar gösterirken C-1BT ile yaklaşık aynı olmaktadır.

4 Sonuçlar

Bu çalışmada tümlev imge kullanılarak hızlı bir şekilde uyarlamalı eşik hesabı yapan ve giriş çerçevelerini iki-bit ile temsil ederek HK gerçekleştirilen özgün bir ikilileştirme yaklaşımı önerilmektedir. Önerilen yaklaşım mevcut düşük bit gösterimli yaklaşımlara göre işlem yükünü özellikle ikilileştirme aşamasında düşürmektedir.

Tablo 1: Düşük-bit derinliği kullanan HK yöntemlerinin ve önerilen çalışmanın başarımı (dB).

| HK Yöntemi | Video Dizileri (Çerçeve Boyutu, Dizi uzunluğu) | | | | | | 6 Video dizisinin ortalaması |
|------------------------|--|---------------------------------------|--|--|--------------------------------------|--------------------------------------|------------------------------|
| | 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) | |
| SAD (8-bit derinlikte) | 22.88 | 32.09 | 23.79 | 30.48 | 29.45 | 23.94 | 27.10 |
| 1BT [12] | 21.83 | 30.32 | 23.31 | 29.83 | 28.11 | 23.61 | 26.17 |
| MF-1BT [13] | 21.81 | 30.38 | 23.26 | 29.88 | 28.18 | 23.63 | 26.19 |
| 2BT [14] | 22.06 | 30.70 | 23.43 | 29.94 | 28.46 | 23.66 | 26.38 |
| C-1BT [15] | 22.10 | 30.86 | 23.38 | 29.98 | 28.71 | 23.69 | 26.45 |
| Önerilen Yöntem | 22.02 | 30.92 | 23.32 | 30.10 | 28.60 | 23.66 | 26.44 |

Tablo 2: Önerilen yaklaşımın diğer yöntemler ile işlem yükü karşılaştırılması.

| | | 1BT [12] | MF-1BT [13] | 2BT [14] | C-1BT [15] | Önerilen Yöntem |
|---------------|---------------|----------|-------------|----------|------------|-----------------|
| İkilileştirme | Toplama | 25 | 16 | 2.8125 | 16 | 4 |
| | Çarpma | 1 | - | 1.0625 | - | - |
| | Kayıdırma | - | 1 | - | 1 | 3 |
| | Çıkarma | - | - | 0.0312 | 1 | 6 |
| | Karşılaştırma | 1 | 1 | 3 | 2 | 2 |
| | Boolean | - | - | 1 | - | - |
| TOPLAM | | 27 | 18 | - | 20 | 15 |
| Uyumlama | Boolean | 1 | 1 | 3 | 3 | 2 |
| | Toplama | - | - | - | - | 1 |
| | TOPLAM | 1 | 1 | 3 | 3 | 3 |

Önerilen yaklaşım HK başarımı açısından incelendiğinde ise bir-bitlik yöntemlere göre çok iyi başarımlar gösterirken 2BT ve C-1BT gibi iki bit uzayı kullanan yaklaşımlar ile karşılaştırıldığında C-1BT aynı başarımları, 2BT'den ise daha iyi başarımlar göstermektedir. Bu durum önerilen yöntemin hareket kestirimi başarımı ile işlem yükü arasında bir denge kurduğunu göstermektedir. İleriki çalışmalarda önerilen bu yaklaşım için donanım mimarisi geliştirilmesi planlanmaktadır.

5 Teşekkür

Bu çalışma, Türkiye Bilimsel ve Teknoloji Araştırma Kurumu (TÜBİTAK) tarafından EEEAG/115E921 No.lu araştırma projesi kapsamında desteklenmiştir.

6 Kaynaklar

[1] Medhat A, Shalaby A, Sayed MS, Elsabrouty M. "A highly parallel SAD architecture for motion estimation in HEVC encoder". *IEEE Asia Pacific Conference Circuits Systems (APCCAS)*, Ishigaki, 17-20 November 2014.

[2] Koga T, Linuma K, Hirano A, Lijima Y, Ishiguro T. "Motion compensated interframe coding for video conferencing". *In National Telecommunication Conference*, C9.6.1-C9.6.5, 29 November -3 December 1981.

[3] Renxiang L, Bing Z, Ming LL. "A new three-step search algorithm for block motion estimation". *IEEE Transactions Circuits and Systems for Video Technology*, 4(4), 438-442, 1994.

[4] Lai-Man P, Wing-Chung M. "A novel four-step search algorithm for fast block motion estimation". *IEEE Transactions Circuits and Systems for Video Technology*, 6(3), 313-317, 1996.

[5] Zhu S, Ma KK. "A new diamond search algorithm for fast block-matching motion estimation". *IEEE Transactions. Image Processing*, 9(2), 287-290, 2000.

[6] Zhu, C, Lin, X, Chau, LP. "Hexagon-based search pattern for fast block motion estimation. *IEEE Transactions Circuits and Systems for Video Technology*, 12(5), 349-355, 2002.

[7] Yu Y, Zhou J, Chen CW. "A novel fast block motion estimation algorithm based on combined subsamplings on pixels and search candidates". *Journal Visual Communication Image Representation*, 12(1), 96-105, 2001.

[8] Wang CN, Yang SW, Liu CM, Chiang T. "A hierarchical decimation lattice based on n-queen with an application for motion estimation". *IEEE Signal Processing Letters*, 10(8), 228-231, 2003.

[9] Cheung CK, Lai-Man P. "Normalized partial distortion search algorithm for block motion estimation". *IEEE Transactions Circuits and Systems for Video Technology*, 10(3), 417-422, 2000.

[10] Kim I, Kim J, Jeong J, Jeon G. "Low-complexity block-based motion estimation algorithm using adaptive search range adjustment". *Optical Engineering*, 51(6), 2012.

[11] Li W, Salari E. "Successive elimination algorithm for motion estimation". *IEEE Transactions Image Processing*, 4(1), 105-107, 1995.

[12] Natarajan B, Bhaskaran V, Konstantinides K. "Low-complexity block-based motion estimation via one-bit transforms". *IEEE Transactions Circuits and Systems for Video Technology*, 7(4), 702-706, 1997.

[13] Ertürk S. "Multiplication-Free one-bit transform for low-complexity block-based motion estimation". *IEEE Signal Processing Letters*, 14(2), 109-112, 2007.

[14] Ertürk A, Ertürk S. "Two bit transform for binary block motion estimation". *IEEE Transactions Circuits and Systems for Video Technology*, 15(7), 938-946, 2005.

- [15] Urhan O, Ertürk S. "Constrained one-bit transform for low-complexity block motion estimation". *IEEE Transactions Circuits and Systems for Video Technology*, 17(4), 478-482, 2007.
- [16] Choi C, Jeong J. 'Enhanced two-bit transform based motion estimation via Extension of matching criterion". *IEEE Transactions Consumer Electronics*, 56(3), 1883-1889, 2010.
- [17] Güllü MK. "Weighted constrained one-bit transform based fast block motion estimation". *IEEE Transactions Consumer Electronics*, 57, 751-755, 2011.
- [18] Song CM, Guo Y, Wang XH, Liu D. "Fuzzy quantization based bit transform for low bit-resolution motion estimation". *Signal Processing: Image Communication*, 28(10), 1435-1447, 2013.
- [19] Sankur B, Sezgin M. "A survey over image thresholding techniques and quantitative performance evaluation". *Journal of Electronic Imaging*, 13(1), 146-165, 2004.
- [20] Sauvola J, Pietikainen M. "Adaptive document image binarization". *Pattern Recognition*, 33(2), 225-236, 2000.
- [21] Sauvola J, Seppanen T, Haapakoski S, Pietikainen M. "Adaptive document binarization". *4th International Conference. on Document Analysis and Recognition*, Ulm Germany, 18-20 August 1997.
- [22] Romen ST, Roy S, Imocha SO, Sinam T, Manglem SK. "A new local adaptive thresholding technique in binarization". *International Journal of Computer Science*, 6(2), 271-277, 2011.
- [23] Crow F. "Summed-area tables for texture mapping". In Proceedings of SIGGRAPH, July 1984.
- [24] Viola P, Jones MJ. "Robust real-time face detection". *International Journal of Computer Vision*, 57(2), 137-154, 2004.