

Kenar Bağımlı Ara Değerleme Yöntemi Kullanarak De-Interlacer İşleminin FPGA Tabanlı Gerçeklenmesi

Implementation of Edge Dependent Interpolation Based De-Interlacer on FPGA

Mehmet Ali Çavuşlu¹, Fuat Karakaya², Alişan Balkoca³

¹KOÇ Bilgi ve Savunma Teknolojileri A.Ş., Ankara, Türkiye
ali.cavuslu@kocsavunma.com.tr

²Elektrik-Elektronik Mühendisliği Bölümü, Niğde Üniversitesi, Niğde, Türkiye
fkarakaya@nigde.edu.tr

³Kağıtsokağı ArGe, Ankara, Türkiye
alisan@kagitsokagi.com

Özetçe— Örgülü tarama tekniği (interlacing) ile alıcı birime gönderilen veri miktarının azaltılmasıyla yüksek çözünürlüklü video sistemlerinin düşük maliyetle gerçekleştirilmesini sağlamaktadır. Örgülü tarama sistemlerinde görüntünün alıcı birimde oluşturulması önem arz etmektedir. Bu çalışmada örgülü tarama ile alıcı birime aktarılan görüntülerin tekrar elde edilmesi işlemi (de-interlacing) kenar bağımlı ara değerlendirme yöntemi kullanılarak FPGA tabanlı donanımsal gerçekleştirilmiştir. Uygulama VHDL donanım tanımlama dilinde kullanılarak Altera Cyclone-II FPGA'sı üzerinde donanımsal gerçekleştirilmiştir. Çalışmada tekrarlı verilerin tekrardan okunmaması sağlanarak zaman kaybının önüne geçilmiştir. Gerçekleme kullanılan FPGA'da %3'lük alan kaplamıştır.

Anahtar Kelimeler — FPGA , De-interlacer, Interlacing,

Abstract— Interlacing technique aims to lower the costs of high definition video systems by reducing the data amount sent to receiver unit. Regeneration of image at the receiver unit is an important point of interlacing method. In this study, regeneration (de-interlacing) of frames that are sent to receiver unit is implemented by using edge dependent interpolation method. The method is implemented using VHDL on Altera Cyclone-II FPGA. The method avoids reading of redundant data which yields to reduced operation time. Implementation occupies only %3 of the FPGA that is used in this study.

Keywords — FPGA , De-interlacer

I. GİRİŞ

İnsan gözü çok hızlı titreşimleri ve değişimleri algılayamamakta ve çoğunlukla yanılgıya düşmektedir. Televizyon sistemleri de bu yanılgıyı kullanarak, insan gözünün algılayabileceği frekansların üzerinde çalışarak sürekli görüntüler elde etmektedir.

PAL, NTSC gibi çoğu televizyon standartlarının ortak özelliği örgülü tarama tekniğine (interlacing) sahip olmalarıdır [1].

Interlacing, kısaca görüntünün bölünerek ardışık olarak gönderilmesi şeklinde tanımlanabilir. Görüntü tek satırlar ve çift satırlar olmak üzere iki alana ayrılır [1].

Bu alanlar arka arkaya gönderilerek bant genişliğinin yarısı kullanılmış olur. Ancak bu yöntem zaten “interlace” mantığında çalışan CRT (Cathod Ray Tube) sistemlerde problem yaratmazken “progressive” çalışan LCD gibi sistemlerde “Mouse teeth” denilen rahatsız edici kaymalara neden olur [3].

CRT monitörler yapısı gereği ilk önce çift satırları içeren alanı (even field) daha sonra da tek satırları içeren alanı (odd field) tarayarak görüntüyü oluşturduğu için interlacing'den fayda sağlar. Göz çok hızlı değişimleri algılayamadığı için iki alan birleşerek tek bir görüntü ortaya çıkar [4].

LCD gibi kademeli (progressive) çalışan yani, bütün kareyi tek seferde alıp ekrana yansıtan sistemlerde “interlaced video” görüntü bozulmalarına neden olur. Bunun sebebi iki alan arasında zaman farkı olmasıdır. LCD iki alanı da aynı anda göstereceği için görüntü iç içe geçmiş gibi görünür. LCD'nin de alanları belirli bir gecikmeyle göstermesi sağlanabilir ancak bu durumda da boş kalan yerlerde siyah alanlar olacak ve parlaklık yarıya düşecektir [5].

Bu çalışmada, iki tarama formatı arasındaki uyumsuzluğu azaltmak ve daha kabul edilebilir bir görüntü oluşturmak için kenar bağımlı ara değerlendirme yöntemi kullanılarak de-interlacer işlemi FPGA tabanlı olarak Altera Cyclone 2 FPGA'sı üzerinde donanımsal olarak gerçekleştirilmiştir.

II. DE-INTERLACER

Eski teknolojilere göre geliştirilmiş olan tarama sisteminin yeni teknolojilerde yarattığı problemlerin giderilmesi noktasında deinterlacing kavramı karşımıza çıkmaktadır. Deinterlacing kısaca interlaced bir kaynağın kademeli çalışan alıcıda uyumlu hale getirilme işlemidir. Deinterlacing işleminin matematiksel açıklaması Denklem (1)'de gösterilmiştir [1];

$$F_0(\bar{x}, n) = \begin{cases} F(\bar{x}, n), & \text{mod}(y, 2) = \text{mod}(n, 2) \\ F_i(\bar{x}, n), & \text{diğer,} \end{cases} \quad (1)$$

Denklem (1)'de \bar{x} uzaysal pozisyonu, $F(\bar{x}, n)$ giriş pikselini, $F_i(\bar{x}, n)$ ara değerlenen pikseli, $F_0(\bar{x}, n)$ çıkış pikselini ve n alan sayısını göstermektedir [1].

Tek veya çift satırları içeren alanlardan herhangi birisi seçildikten sonra, diğer alan seçilen alan üzerinden hesaplanarak çıkış fonksiyonuna verilir. Hesap yapmak üzere çift satırları içeren alanı seçtiğimizi düşünürsek, bir kare oluşturulurken çift satırlar üzerinde oynama yapılmadan, tek satırlar da gerekli hesaplamalar yapıldıktan sonra oluşturularak birleştirilir.

Bu işlemin gerçekleştirilmesini sağlayan algoritmalar doğrusal, doğrusal olmayan ve hareket dengelemeli olarak üç ana başlık altında toplanabilir. Bu algoritmaların en karışık olanları hareket dengeleme yöntemini kullananlardır. Ancak en iyi sonuçları da bu algoritmalar verir [1].

Bu çalışmada, doğrusal olmayan tekniklerden, kenar bağımlı ara değerlendirme yöntemi kullanılmıştır. Bu yöntem kolay olması ve donanım tabanlı gerçekleştirilmede maliyetinin düşük olması nedeniyle seçilmiştir. Aşağıda bu yöntem kısaca tanıtılacaktır.

A. Kenar Bağımlı Ara Değerleme Yöntemi

Kenar bağımlı ara değerlendirme yöntemi, kenar yönelimine karar verilecek referans piksel değeri için komşu piksellerin değerlerini kullanır (Şekil 1). Denklem (2) ve Şekil 1'den de görüleceği üzere; X referans piksel değerinin:

- Yatayda -1 birim ve dikeyde -1 birim uzağındaki piksel değeri A ,
- Dikeyde -1 birim uzağındaki piksel değeri B ,
- Yatayda +1 birim ve dikeyde -1 birim uzağındaki piksel değeri C ,
- Yatayda -1 birim ve dikeyde +1 birim uzağındaki piksel değeri D ,
- Dikeyde +1 birim uzağındaki piksel değeri E ,
- Yatayda +1 birim ve dikeyde +1 birim uzağındaki piksel değerini F ,
- Dikeyde -3 birim uzağındaki piksel değeri G ,
- Dikeyde -3 birim uzağındaki piksel değeri H ,

- olarak ifade edilmektedir.

Yeni piksel değerinin belirlenebilmesi için kenar yöneliminin tespit edilmesi gerekmektedir. Bu işlem ile üç doğrultudan işlem yapılır ve değişimin en az olduğu doğrultuya karar verilir. Denklem (3)'den de görüleceği üzere minimum farkın olduğu doğrultu seçilmektedir. Eğer $A-F$ doğrultusunda mutlak fark minimum ise çıkışa B ve F piksel değerlerinin toplamının ortalaması aktarılmaktadır (Denklem 4). $B-E$ doğrultusunda mutlak fark minimum ise çıkışa B ve E piksel değerlerinin toplamının ortalaması aktarılmaktadır (Denklem 5). $C-D$ doğrultusunda mutlak fark minimum ise çıkışa C ve D piksel değerlerinin toplamının ortalaması aktarılmaktadır (Denklem 6).

$$\begin{aligned} A &= F(\bar{x} - \bar{u}_x - \bar{u}_y, n) & B &= F(\bar{x} - \bar{u}_y, n) \\ C &= F(\bar{x} + \bar{u}_x - \bar{u}_y, n) & D &= F(\bar{x} - \bar{u}_x + \bar{u}_y, n) \\ E &= F(\bar{x} + \bar{u}_y, n) & F &= F(\bar{x} + \bar{u}_x + \bar{u}_y, n) \\ G &= F(\bar{x} - 3\bar{u}_y, n) & H &= F(\bar{x} - 3\bar{u}_x, n) \end{aligned} \quad (2)$$

$$X = \begin{cases} X_A, & (|A-F| < |C-D|) \wedge (|A-F| < |B-E|) \\ X_C, & (|C-D| < |A-F|) \wedge (|C-D| < |B-E|) \\ X_B, & b.d. \end{cases} \quad (3)$$

$$X_A = \frac{A+F}{2} \quad (4)$$

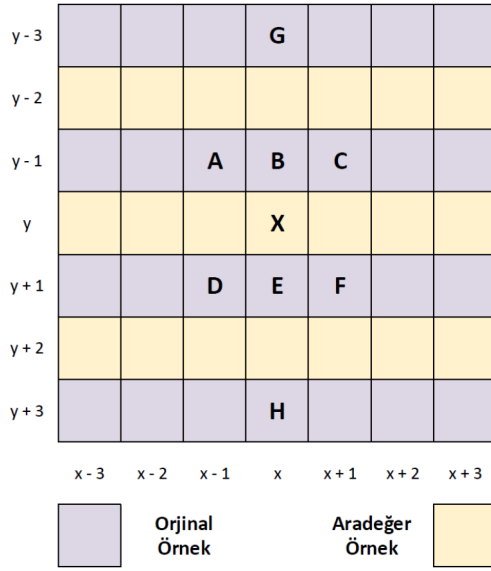
$$X_B = \frac{B+E}{2} \quad (5)$$

$$X_C = \frac{C+D}{2} \quad (6)$$

Şekil 2 ve Şekil 3'te renkli interlacing olmuş imgeler üzerinde kenar bağımlı ara değerlendirme yöntemi kullanılarak oluşturulmuş imgeler gösterilmektedir. Şekillerden de görüleceği üzere interlace olmuş imgelerden net görüntüler elde edilmiştir.

III. KENAR BAĞIMLI ARA DEĞERLEME YÖNTEMİ KULLANILARAK DE-INTERLACER'IN FPGA TABANLI GERÇEKLENMESİ

FPGA tabanlı kenar bağımlı ara değerlendirme yöntemi kullanılarak de-interlacer işleminin gerçekleştirilmesine ilişkin blok diyagram Şekil 4'de gösterilmiştir. **Veri Kontrol** işlem birimi **SRAM**'dan okunacak ve yazılacak olan verilerin adres bilgisinin kontrolünü, okunan verilerin **Kaydırmalı Saklayıcı** işlem birimine iletilmesini ve değeri hesaplanan yeni piksel değerini **SRAM**'da adersine yazma işlemlerini gerçeklemektedir



Şekil 1 Ara değerleme için seçilen komşu pikseller



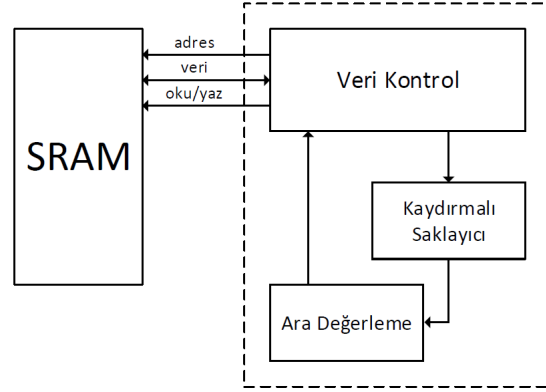
Şekil 2 Interlace olmuş imgenin kenar bağımlı ara değerleme yöntemi kullanılarak tekrar edilmesi - 1



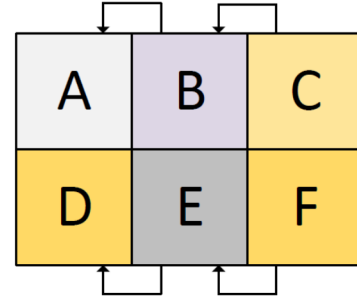
Şekil 3 Şekil 2 Interlace olmuş imgenin kenar bağımlı ara değerleme yöntemi kullanılarak oluşturulması - 2

Ara değerleme işlemi yapılacak olan pikselin değerinin belirlenebilmesi için Şekil 1’de gösterilen *A, B, C, D, E, F*

komşu piksellerinin *SRAM*’dan okunması gerekmektedir. Okunan bu değerler 2x3’lük bir matrise yazılmaktadır (Şekil 5). Matrisin her bir elemanın derinliği 8 bittir.



Şekil 4. FPGA tabanlı gerçekleştirme işlem blokları



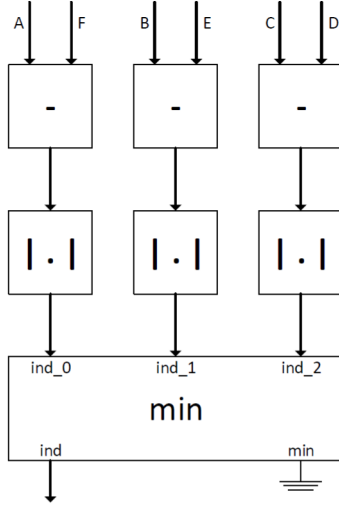
Şekil 5. FPGA tabanlı gerçekleştirme işlem blokları

Şekil 5’ten de görüleceği üzere her bir pikselin ara değerinin hesaplanabilmesi için 6 adet örneğe ihtiyaç vardır. Bir sonraki pikselin ara değerinin hesaplanması işleminde Şekil 5’te gösterilen *B* değeri *A* değerinin, *C* değeri *B* değerinin, *E* değeri *D* değerinin ve *F* değeri *E* değerinin yerini almaktadır. Bu durumda yeni ara değer işlemi için *SRAM*’dan *C* ve *F* değerlerinin okunması yeterli olacaktır. Bu durum göz önünde bulundurularak matris yapısı kaydırmalı saklayıcı olarak gerçekleştirilmiştir.

Matris bloğunun oluşturulması işleminden sonra minimum doğrultunun tespit edilmesi gerekmektedir. Bu işlemin gerçekleştirilmesine ilişkin detaylı blok diyagram Şekil 6’da gösterilmiştir. Şekil 6’dan da görüleceği üzere, ara değerleme yapılacak pikselin doğrultu noktalarındaki komşuları arasındaki mutlak fark hesaplanmaktadır. Mutlak değerlerin hesaplanması ile elde edilen 3 doğrultu mesafeleri arasındaki minimum mesafenin indis değerinin tespiti yapılmaktadır.

Minimum değerin tespit edilmesi ile birlikte yeni piksel değerinin hesaplanma işlemi Denklem (7)’de gösterilmiştir. Örneğin minimum doğrultumuzun *A-F* piksel değerleri arasında olduğunu kabul edelim. Bu durumda *ind* değerimiz 0 olacaktır. Bu durumda piksel

değerimiz $Matris(0, 0)$ ve $Matris(1, 2)$ değerlerinin yani A ve F değerlerinin ortalaması olacaktır.



Şekil 6. Ara değer hesabı için minimum doğrultu tespiti blok diyagramı

$$Piksel = \frac{Matris[0, ind] + Matris[1, 2 - ind]}{2} \quad (7)$$

IV. SONUÇLAR

Bu çalışmada kenar bağımlı ara değerleme yöntemi kullanılarak deinterlaced işleminin FPGA tabanlı gerçekleştirilmesinde Altera Cyclone 2 FPGA kullanılmıştır. VHDL dili kullanılarak geliştirilen uygulama Quartus yazılımı ile sentezlenerek test edilmiştir. FPGA üzerinde gerçekleştirilen sisteme ait sentezleme sonuçları Tablo 1’de gösterilmiştir.

	Kullanılan	Toplam	Yüzde
LEs	1014	33216	%3

Tablo 1. Ara değerleme yöntemi ile De-Interlaced’in FPGA tabanlı gerçekleştirilmesine ait sentez sonuçları

N sütunluk bir satırın kenar bağımlı ara değerleme yöntemi kullanılarak tekrar oluşturulması işlemi için hafıza biriminden $6N$ adet okuma işlemi yapılması gerekmektedir. Şekil 5’te gösterildiği gibi kaydırmalı saklayıcı türünde hafıza birimi kullanılması ile N sütunluk bir satırın kenar bağımlı ara değerleme yöntemi kullanılarak tekrar oluşturulması ile $2N$ adet okuma işlemi yapılmaktadır.

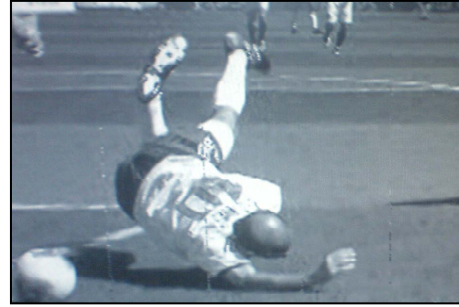
Şekil.7’de FPGA tabanlı gerçekleştirilmesinde kullanılacak olan interlace imge gösterilmektedir. Şekil 8’de ise Şekil 7’de verilen imgenin kenar bağımlı ara değerleme yöntemi kullanılarak FPGA tabanlı gerçekleştirilmesinden sonraki hal gösterilmektedir.

Şekil 8’den de görüleceği üzere FPGA tabanlı gerçekleştirilme sonucunda imgenin tekrar elde edilme işlemi

başarılı bir şekilde gerçekleştirilmiştir. Tablo 1’den verilen sentezleme sonuçları ise kullanılan FPGA’nın sadece %3’lük bir kısmı kullanılarak deinterlaced işleminin hızlı gerçekleştirilmesi sağlanmıştır.



Şekil 7. FPGA tabanlı gerçekleştirilmesinde kullanılan interlace imge



Şekil 8. Ara değerleme yöntemi kullanılarak De-Interlaced’in FPGA tabanlı gerçekleştirilmesi

KAYNAKÇA

- [1] Srinu.V, K.N.H.Srinivas, Addanki Purna Ramesh, Acceleration of Directional Median Filter Based Deinterlacing Algorithm (DMFD), *International Journal of Computer Science and Communication Networks*, 01(03) 344-353, 2011
- [2] Haan, G., Bellers, E. 1998. “Deinterlacing—An Overview”. *Proceedings of IEEE*, 86(9); 1839-1857.
- [3] Gwo Giun Lee, Ming-Jiun Wang, Hsin-Te Li, and He-Yuan Lin, “AMotion-Adaptive Deinterlacer via Hybrid Motion Detection and Edge-Pattern Recognition”, Hindawi Publishing Corporation *EURASIP Journal on Image and Video Processing* Volume 2008, doi:10.1155/2008/741290
- [4] <http://en.wikipedia.org/wiki/Deinterlacing>
- [5] Keller, S.H., Lauze, F. ; Nielsen, M., “Deinterlacing Using Variational Methods”, *IEEE Transactions on Image Processing*, Volume:17, Issue: 11, pp: 2015 – 2028, 2008