

FPGA üzerinde görüntü işleme algoritmalarının gerçek zamanlı gerçekleştirilmesi

İsmail Koyuncu¹, Muhammed Furkan Taşdemir¹, Murat Alçın², Murat Tuna^{3*}, Ercan Coşgun⁴

¹Afyon Kocatepe Üni., Tekn. Fak. Ele.-Elektr. Müh. Böl., Ahmet Necdet Sezer Kampüsü, Afyonkarahisar

²Afyon Kocatepe Üni., Teknoloji Fak. Mekat. Müh. Böl., Ahmet Necdet Sezer Kampüsü, Afyonkarahisar

³Kırklareli Üniversitesi Teknik Bilimler MYO Elektrik Bölümü, Karahıdır Kampüsü, Kırklareli

⁴Kırklareli Üniversitesi Teknik Bilimler MYO Elektronik ve Otom. Böl., Karahıdır Kampüsü, Kırklareli

Geliş Tarihi (Received Date): 09.03.2021

Kabul Tarihi (Accepted Date): 07.09.2021

Öz

Sunulan bu çalışmada, köşe algılama algoritmalarından FAST ve Harris köşe algılama algoritmaları, kenar bulma algoritmalarından Sobel kenar bulma algoritması, morfolojik işlem algoritmalarından yayma ile aşındırma yöntemleri ve renk değiştirme algoritması FPGA çipleri üzerinde çalışmak üzere gerçek zamanlı olarak tasarlanmıştır. Tasarlanan gerçek zamanlı görüntü işleme algoritmalarında Xilinx Vivado Design Suite HLx kullanılmıştır. Gerçek zamanlı görüntüler HDMI aracılığıyla kameradan alınmıştır. Alınan gerçek zamanlı görüntü verilerinin FPGA üzerinde işlenebilmesi için VHDL kullanılmıştır. Yapılan tüm tasarımlar Xilinx Zybo Z7-20 kartı üzerinde gerçekleştirilmiştir. FPGA-tabanlı tasarımların sonuçlarından elde edilen görüntü verileri, HDMI aracılığıyla monitöre aktarılmıştır. Xilinx Zybo Z7-20 FPGA kartı üzerinde çalışmak üzere tasarımı yapılan gerçek zamanlı görüntü işleme algoritmalarından elde edilen sonuçlar sunulmuştur.

Anahtar Kelimeler: Gerçek zamanlı görüntü işleme, FPGA, Vivado HLx, VHDL.

*Murat TUNA, murat.tuna@klu.edu.tr, <https://orcid.org/0000-0003-3511-1336>

İsmail KOYUNCU, ismailkoyuncu@aku.edu.tr, <https://orcid.org/0000-0003-4725-4879>

M. Furkan TAŞDEMİR, furkantasdemir123@gmail.com, <https://orcid.org/0000-0002-3401-2929>

Murat ALÇIN, muratalcin@aku.edu.tr, <https://orcid.org/0000-0002-2874-7048>

Ercan COŞGUN, ercan.cosgun@klu.edu.tr, <https://orcid.org/0000-0003-4387-3699>

Real time realization of image processing algorithms on FPGA

Abstract

In this presented study, FAST and Harris corner detection algorithms, Sobel algorithm from edge detection algorithms, dilation and erosion methods from morphological processing algorithms, and color changing algorithm have been designed to work on FPGA chips in real time. Xilinx Vivado Design Suite HLx was used in the designed real time image processing algorithms. Real-time images were taken from the camera via HDMI. VHDL was used to process the real-time image data received on FPGA. All designs were implemented on the Xilinx Zybo Z7-20 card. Image data obtained from the results of FPGA-based designs were transferred to the monitor via HDMI. The results obtained from real-time image processing algorithms designed to work on Xilinx Zybo Z7-20 FPGA board have been presented.

Keywords: Real time image processing, FPGA, Vivado HLx, VHDL.

1. Giriş

Son yıllarda sayısal görüntü işleme uygulamaları alanında yoğun bir şekilde çalışmalar yapılmaktadır. Tıp [1], ölçme [2], kriptoloji [3], radar teknolojisi [4], robotik [5], derin öğrenme, gıda ve otomasyon bu çalışmalardan bazılarıdır [6]. Görüntüyü anlamlı alt bölgelere ayırıştırma, kullanılan özellik doğrultusunda alt bölgelerin tanımlanması, tanımlanan nesnelere etiketlenmesi ve etiketlenen nesnelere göre karar verme işlemlerinden oluşan görüntü işleme; normal bir görüntünün, işlenmiş veya değiştirilmiş bir sayısal görüntüye dönüşmesi için kat edilen yol ve yöntemlerdir. Sayısal görüntünün anlamlı alt bölgelere ayrılması işleminde genellikle kenar ve köşe belirleme teknikleri kullanılmaktadır. Kenarların belirlenmesi aşamasında piksel komşulukları oldukça önemlidir. Genel olarak eğim kenar algılama, sıfır geçiş (ikincil türev algılama), gaussian kenar algılama (Canny) ve vektör sıralama istatistiği ile gerçekleştirilmiş kenar algılama olmak üzere dört farklı kenar belirleme tekniği bulunmaktadır. Sobel operatörü de eğim kenar algılayıcılarından birisidir. Bu operatörde, sayısal görüntünün yataydaki türevleri ile ve dikeydeki türevleri çekirdek yapıları kullanılarak bulunmaktadır. Bu işlemde elde edilen değerler görüntü üzerindeki her noktanın eğim değerleri ile toplanarak kenar algılama işlemi gerçekleştirilmektedir [7]. İkincil türev algılayıcılarında ise, öncelikle birinci türev tepe noktaları belirlenmekte ve ardından ikinci türevde sıfır çaprazlama ile kenar belirleme işlemi gerçekleştirilmektedir [8].

Yoğun çalışma yapılan alanlardan birisi olan görüntü işleme, birçok mühendislik alanında başarıyla kullanılmaktadır. Görüntü işleme, girişleri ve çıkışları görüntüler olan işlemleri ve ayrıca bu görüntülerden öznelikleri çıkartan süreçleri kapsamaktadır. Bu süreçlere, bölümlenme, morfolojik işlemler, kenar bulma, köşe bulma ve filtreleme gibi işlemler örnek olarak verilebilir [9]. Bu işlemler için kullanılan resim veya görüntüler piksellerden oluşmaktadır. Bir VGA aracılığı ile aktarılan 640x480 piksel boyutundaki bir görüntü üzerinde işlem yapılabilmesi için 307,200 adet verinin hesaplanması gerekmektedir. Görüntü kalitesi veya boyutu arttığında işlem hacmi de büyümektedir. Bu işlemlerin gerçek zamanlı yapılabilmesi

için yüksek hızlı ve paralel işlem yapabilen sayısal platformlara ihtiyaç duyulmaktadır.

Günümüzde yaygın bir şekilde kullanılan sayısal sistem tasarımı araçlarından birisi olan FPGA (Field Programmable Gate Array-Alan Programlanabilir Kapı Dizileri) çipleri, paralel sinyal işleme, hızlı ilk üretim ve yüksek çalışma frekansı gibi özelliklerinden dolayı görüntü işleme uygulamalarında tercih edilen tüm-devre yapılarıdır. FPGA çipleri Konfigüre Edilebilir Mantıksal Bloklar (Configurable Logic Blocks (CLB)), Giriş/Çıkış Blokları (Input/Output Blocks (IOB)) ve Ara Bağlantılar (Programmable Interconnect) olmak üzere genel olarak üç kısımdan oluşmaktadırlar. FPGA çipleri ilk örnek aşamasında ASIC ((Application Specific Integrated Circuit-Uygulamaya Özel Tümüleşik Devre) ile FPGA çipleri karşılaştırıldığında, FPGA daha düşük maliyet ve daha hızlı ilk üretim gibi avantajlar sunmaktadır. DSP (Digital Signal Processor-Sayısal Sinyal İşlemci) ile FPGA çipleri karşılaştırıldığında ise paralel işlem yapabilme ve yüksek çalışma frekansı gibi üstünlüklerinden dolayı FPGA çipleri daha fazla ilgi çekmektedir. FPGA çipleri yüksek hız, düşük maliyet, tekrar tekrar programlanabilme ve yüksek frekans gibi önemli avantajlardan dolayı oldukça yüksek işlem gücü ile yüksek performans gerektiren sayısal görüntü işleme ve gerçek zamanlı uygulamalarda yoğun bir şekilde kullanılmaktadır [10].

Kumar ve arkadaşları tarafından FPGA-tabanlı olarak yapılan çalışmada, Sobel Operatörü kullanılarak kenar belirleme uygulaması 2 boyutlu olarak gerçekleştirilmiştir ve çalışmadan iyi sonuçlar alındığı ifade edilmiştir [11]. Mehra ve arkadaşları tarafından yapılan bir diğer çalışmada, FPGA çiplerinde çalışmak üzere 2 boyutlu Sobel Operatörü kullanılarak kenar algılama çalışması yapılmıştır. Tasarlanan FPGA-tabanlı Sobel operatörünün maksimum çalışma frekansı 148 MHz olarak sunulmuştur [12]. Literatüre sunulan Anusha ve arkadaşları tarafından sunulan bir diğer çalışmada FPGA çipi üzerinde 3 boyutlu Sobel Operatörü ile kenar tespit uygulaması gerçekleştirilmiştir [13]. Karaköse ve arkadaşları tarafından yapılan bir diğer çalışmada, FPGA-tabanlı morfolojik işlem algoritması geliştirilerek nesnelerin tespiti kolaylaştırılmıştır [14]. Aydoğdu ve arkadaşları bu çalışmalarında FPGA üzerine bağladıkları iki kamera yardımıyla köşe tespiti yapmışlardır [15]. Gacar tarafından yapılan bir diğer çalışmada, FPGA tabanlı görüntü işleme ara birimi üzerinde güvenlik kamerasından görüntülenen görüntüyü bilgisayar monitörü üstünde görüntülenmiştir [16]. Özçelik tarafından sunulan farklı bir çalışmada, FPGA kartı üzerinde morfolojik işlemler ve konvolüsyon işlemleri ile ten rengi ve kan hücreleri tanıma üzerine çalışılmıştır [17]. Çil tarafından literatüre sunulan bir diğer çalışmada, FPGA platformu üzerinde konvolüsyon işlemi gerçekleştirilmiştir. Ayrıca elde edilen sonuçlar Matlab sonuçları ile karşılaştırılmıştır [18]. Kızılkaya tarafından yapılan bir çalışmada, görüntülerin görüntü işleme yöntemlerinden histogram ve konvolüsyon işlemleri gerçekleştirilmiş ve çalışma sonuçları sunulmuştur [19]. Çelik tarafından yapılan çalışmada, FPGA üzerinde difüzyon ile sınırlı tanecik kümeleşme modeli kullanılarak fraktal bir şeklin görüntüsü elde edilmiştir. Elde edilen iki boyutlu görüntüler üç boyutlu tek bir görüntü haline getirilmiştir [20]. Özalp tarafından yapılan çalışmada, görüntülere konvolüsyon işlemleri uygulanmış ve elde edilen sonuçlar ekran da gösterilmiştir [21]. Altuncu tarafından yapılan çalışmada ise Zedboard FPGA kartı üzerinde kenar bulma ve filtreleme gibi görüntü işleme uygulamaları gerçekleştirilmiştir [22]. Hu jing ve arkadaşı futbol gibi sporlardan elde ettikleri görüntüleri FPGA üzerinde işleyerek sporcuyu analiz etmişlerdir [23]. M. Selvaganesh ve arkadaşları geleneksel olmayan bir yöntem ile düşük gecikmeli ve

yüksek hassasiyetli medyan filtre mimarisini FPGA çipinde uygulamışlardır [24]. Arshad ve arkadaşları FPGA çipi üzerinde kaotik tabanlı görüntü şifreleme çalışması yapmışlardır [25].

Sunulan bu çalışmada, literatürde sunulan çalışmalardan farklı olarak, ilk aşamada gerçek zamanlı görüntüler kameradan HDMI (High Definition Multimedia Interface (Yüksek Çözünürlüklü Çoklu-Ortam Arayüzü)) aracılığı ile alınmıştır. İkinci aşamada, köşe algılama algoritmalarından FAST (Features from Accelerated Segment Test (Hızlandırılmış Segment Testinden Özellikler)) ve Harris Köşe Algılama Algoritmaları, Kenar Bulma Algoritmalarından Sobel Kenar Bulma Algoritması, Morfolojik İşlem algoritmalarından Yayma ile Aşındırma Yöntemleri ve Renk Değiştirme Algoritması FPGA çipleri üzerinde çalışmak üzere gerçek zamanlı olarak tasarlanmıştır. Kameradan alınan gerçek zamanlı görüntü verilerinin FPGA üzerinde işlenebilmesi için kullanılan FAST ve Harris Köşe Algılama Algoritmaları, Sobel Kenar Bulma Algoritması, Yayma ile Aşındırma Yöntemleri ve Renk Değiştirme Algoritmaları VHDL dilinde kodlanmıştır. FPGA-tabanlı tasarımlar Xilinx Vivado Design Suite programı kullanılarak sentezlenmiş ve test edilmiştir. Üçüncü aşamada, tasarlanan görüntü işleme algoritmaları Zybo Z7-20 geliştirme kartına yüklenmiştir. Dördüncü aşamada, Zybo Z7-20 geliştirme kartından alınan gerçek zamanlı görüntü verileri HDMI aracılığıyla monitöre aktarılmıştır. Son aşamada ise tasarımlardan elde edilen FPGA çip istatistikleri ve performans sonuçları sunulmuştur.

2. Metot ve Yöntem

Bu bölümde, FPGA çipleri ve çip üzerinde gerçekleştirilen gerçek zamanlı görüntü işleme algoritmaları hakkında genel bilgiler verilmiştir.

2.1. FAST Köşe Algılama Algoritması

Köşe Algılama Algoritmaları, görüntüde istenilen veri türlerini tüm görüntüden ayıklamayı amaçlayan bir yaklaşımdır. Köşe, paralel olmayan iki kenarın kesişimidir. Kenar ise görüntü üzerinde ani piksel yoğunluğunun değişiminden anlaşılmaktadır. Köşe algılama algoritmaları, hareket algılama, cisim algılama, görüntü kaydetme, video izleme, görüntü bulanıklaştırma, stereo görüntü ve nesne tanımda sıklıkla kullanılmaktadır. FAST Köşe Algılama Algoritmasında üç ana temel nicelik üzerinde durulmaktadır. Tespit edilen pozisyonlar tutarlı olmalıdır. Gürültü değişimine duyarız ve aynı noktadan birden fazla görüntü elde edildiğinde hareket olmamalıdır. Köşeler, doğru pozisyonlara mümkün olduğu kadar yakın tespit edilmelidir. Köşe detektörü yeterince hızlı olmak zorundadır [26].

2.2. Harris Köşe Algılama Algoritması

Bu yöntemde dikkat edilmesi gereken üç önemli nokta vardır. Harris grafiğindeki iki eğri de küçükse, diğer bir deyişle, yerel otokorelasyon işlevi düzse, pencerenin olduğu bölgenin görüntüsü yaklaşık olarak sabit yoğunluktadır. Eğer bir eğri yüksek ve bir eğri de düşükse, diğer bir deyişle, yerel otokorelasyon fonksiyonu sırt şeklinde ise, o zaman sadece sırt (kenar) boyunca kaymalar E değerinde küçük bir değişikliğe neden olmaktadır. Bu durum bir kenarı ifade etmektedir. Her iki eğri de yüksekse, diğer bir deyişle yerel otokorelasyon fonksiyonu keskin bir şekilde zirveye ulaşırsa, o zaman herhangi bir yöndeki kayma E değerini de artırmaktadır. Bu durum bir köşeyi ifade etmektedir. Harris köşe algılama algoritması, elde edilen görüntülerde bulunan köşelerin

tespiti ve özniteliklerini çıkarmak için makine öğrenmesi [27], bulanık mantık [28] gibi bilgisayarlı görme algoritmalarında kullanılan bir köşe bulma algoritmasıdır. Harris köşe algılama algoritması görüntünün parlaklığı ve rotasyonu gibi değişen durumlar da kararlı davranıp aynı sonuçları verebilmektedir. Bu nedenle stereo eşleştirme ve görüntü veritabanı alanında daha sık kullanılmaktadır [29].

2.3. Aşındırma Yöntemi İle Morfolojik İşlem Algoritması

Aşındırma yöntemi bir morfolojik işlemdir. İkilik sisteme dönüştürülmüş olan imgedeki nesneyi küçültmeye ya da inceltmeye yaramaktadır. Yapılandırma ögesinin tamamen uyduğu şekildeki bütün pikseller tespit edilip değiştirilir ve sadece merkez noktadaki değer aynı kalmaktadır. Aşındırma yöntemi tam anlamıyla olmasa da bir bakıma yayma işleminin tersi gibi düşünülebilmektedir. İmge içerisindeki nesnelere küçültme, delikler genişletme ve birbirine bağlı olan nesnelere ayrılmasıdır.

2.4. Yayma Yöntemi İle Morfolojik İşlem Algoritması

Yayma yöntemi ikili sisteme, diğer bir deyişle, binary sisteme dönüştürülmüş olan görüntüdeki nesneyi büyütme veya kalınlaştırma işlemi yapan morfolojik işlemdir. Sayısal bir ifadeyi genişletmek, görsel yapısal elemanla kesiştiği bölümler kadar büyütme işlemidir. İşlenecek görselin her bir pikseli, yapısal elemanın merkez noktasına oturtularak genişletme işlemi yapılmaktadır.

2.5. Sobel Kenar Bulma Algoritması

Sobel Filtresi siyah beyaz görüntüye uygulanabilen bir kenar bulma algoritmasıdır. Sobel Filtresi temel olarak $[3 \times 3]$ 'lük bir çekirdek (kernel) matrislerden oluşmaktadır. Bu çekirdek matrisler görüntü içinde bulunan kenarları bulmak için kullanılmaktadır. Sayısal bir görüntü, bir fonksiyon olarak değerlendirildiğinde, bir nokta üzerindeki 1. türev değerinin, $[3 \times 3]$ 'lük komşu olan dört yönde elde edilebilir 1. türev değerlerinin vektör toplamları olmaktadır. Bu yaklaşımla Sobel kenar işleci oluşturulmuştur [30].

2.6. Renk Değiştirme Algoritması

Görüntü işleme üzerinde görüntülerin tanımlanması için kırmızı, yeşil ve mavi olmak üzere üç ana renk kullanılmaktadır. Görüntüler üzerinde değişiklik yapılabilmesi için bu üç ana renk üzerinde değişiklik yapılması gerekmektedir. Görüntü işleme literatüründe bu renklere RGB denilmekte ve kırmızı R (Red), yeşil G (Green), mavi ise B (Blue) ile kısaltılmaktadır. Renk değiştirme algoritmasında, görüntüden elde edilen RGB değerleri bir birim sola kaydırılarak yeni bir piksel değeri oluşturulmaktadır.

2.7. FPGA Çipleri

FPGA çipleri üretim anından itibaren kullanıcının kullanacağı doğrultuda istenilen sistem tasarımına göre yazılım ve donanım yapısı tekrar tekrar programlanıp kullanılabilme özelliği bulunan dijital tümdevrelerdir [31]. Bir FPGA üç ana bloktan oluşmaktadır. Konfigüre Edilebilir Lojik Bloklar (Configurable Logic Blocks (CLB)): Look-Up Table (LUT), Flip-Flop' lar ve multiplexer gibi çeşitli bölümlerden oluşan CLB, kullanıcı tarafından sistem tasarımı yapılabilen lojik devre için fonksiyonel elemanlar sağlamaktadır. Giriş-Çıkış Blokları (Input/Output Blocks (IOB)): IOB pinleri yapılması hedeflenen tasarıma göre giriş, çıkış veya giriş-çıkış olarak programlanabilir. IOB bloklarının görevi FPGA çipindeki iç sinyal hatları ve pinler arasında arabirim bağlantıları sağlar. Ara Bağlantılar (Programmable Interconnect): Ara bağlantı birimleri, mantıksal bloklar arasında ve mantıksal blokların IOB'ler ile bağlantılarının konfigüre edilmesinde kullanılırlar [32-33]. FPGA yongalarında kullanılan dil çoğunlukla VHDL

ve Verilog programlama dilleridir. Paralel çalışma, düşük güç tüketimi, hızlı ilk üretim ve yüksek kapasite gibi teknik özellikleriyle diğer sayısal tasarım platformlardan farklı olan FPGA çipleri; günümüzde osilatör tasarımı, yapay sinir ağları, havacılık, savunma sanayi, otomotiv, uzay bilimi, ses, video, görüntü işleme ve daha birçok alanda kullanılmaktadır.

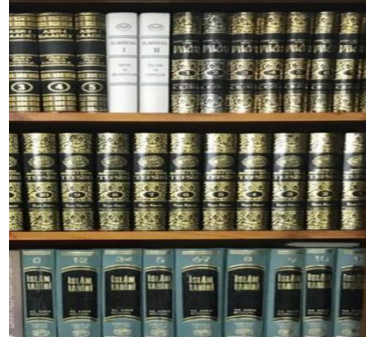
Bilgisayarın icadından itibaren 1950 yıllarından günümüze kadar tümleşik yapıların tasarımı hızla gelişen bir bilim ve teknoloji alanı olmuştur. Bu teknolojilerden birisi olan FPGA çipleri 1985 yılında Xilinx firması tarafından geliştirilmiştir. Tarihte programlanabilir olarak üretilen ilk mantık aygıtı PROM (Programmable Read Only Memory (Programlanabilir Salt Okunur Hafıza)) hafızasıdır. Bu terim, bilgi yüklenebilen kalıcı bir bellek anlamına gelmektedir. Farklı birçok PROM türü bir fabrikada toplu olarak (ASIC) ya da kullanıcı (FPGA) tarafından veya sahada programlanabilmektedir. Sahada programlanabilir olanlar FPGA çiplerinin evrimleştiği tiptir. Sahada programlanabilir PROM iki tipte olabilmektedir: EPROM (Erasable Programmable Read Only Memory (Silinebilir ve Programlanabilir Salt Okunur Hafıza)) ve EEPROM (Electrically Erasable Programmable Read-Only Memory (Elektronik olarak Silinebilir ve Programlanabilir Salt Okunur Hafıza)). En yaygın kullanılan EEPROM, kullanıcının belleğin içeriğini silmesine ve birçok kez yeniden programlamasına olanak tanımaktadır. FPGA çipleri tekrar tekrar programlanabilen paralel işlem kabiliyeti olan tümleşik devrelerdir. Bir dizi mantık bloğu ve yönlendirme kanalından oluşmaktadır. İki adet giriş ve çıkış pedi, bir satırın yüksekliğini veya bir sütunun genişliğini ifade etmektedir. Tüm yönlendirme kanalları aynı genişliğe sahip olmak zorundadır. İlk üretilen FPGA çipi Xilinx tarafından 1985 yılında üretilen XC2064 çipidir.

3. FPGA Tabanlı Görüntü İşleme Uygulamaları ve Sonuçları

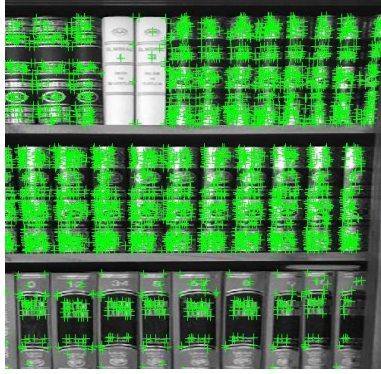
Bu bölümde, ikinci bölümde detaylı bir şekilde açıklanan görüntü işleme algoritmaları FPGA çipi üzerinde çalışmak üzere VHDL ile tasarlanmıştır. Bu algoritmalar Vivado HLS (High Level Synthesis) programı ile sentezlenmiş ve IP Core haline getirilmiştir. Vivado SDK uygulanmasından da faydalanılmıştır. Tasarım platformu olarak Xilinx Vivado 2017.4 kullanılmış ve tasarlanan sistem Zybo Z7-20 FPGA çipi için sentezlenmiştir.

3.1. FPGA tabanlı FAST köşe algılama görüntü işleme algoritması uygulaması

Şekil 1(a)'da, üzerinde herhangi bir algoritma uygulanmayan ışık düzeyi yüksek görüntü, Şekil 1(b)'de Matlab uygulamasından alınan sonuç ve Şekil 1(c)'de ise FPGA üzerinde yapılan çalışmadan elde edilen görüntü verilmiştir. Işık düzeyi yüksek görüntü sonuçlarına bakıldığında sonuçların neredeyse birbirine yakın olduğu görülmektedir. Üst kısımdaki beyaz kısımlara bakıldığında FPGA ile yapılan köşe tespitinin bir miktar da olsa daha etkili olduğu görülmektedir. Ayrıca Matlab uygulamasının köşe işaretlemelerinin Vivado uygulamasının köşe işaretlemelerine göre daha büyük boyutta olması göz yanılıgısına sebep olabilmektedir. Kıyaslama yapılırken buna dikkat edilmelidir. Ayrıca elde edilen sonuçlar Tablo 1'de verilmiştir.



a) Orijinal görüntü.



b) Matlab uygulamasından alınan görüntü.



c) FPGA uygulamasından alınan görüntü.

Şekil 1. Işık düzeyi yüksek bir ortamda görüntüler üzerinde köşe bulunması

3.2. FPGA tabanlı Harris köşe algılama görüntü işleme algoritması uygulaması

Şekil 2(a)'da, Harris köşe algılama algoritması uygulanmamış yalın haldeki orijinal görüntü verilmiştir. Şekil 2(b)'de Matlab üzerinde Harris algoritması ile rubik küplerin köşelerinin algılanması, Şekil 3(c)'de FPGA üzerinde Harris köşe algılama algoritması ile rubik küplerin köşelerinin algılanması gösterilmiştir ve elde edilen sonuçlar Tablo 1'de sunulmuştur.



a) Orijinal görüntü.



b) Matlab uygulamasından alınan görüntü.

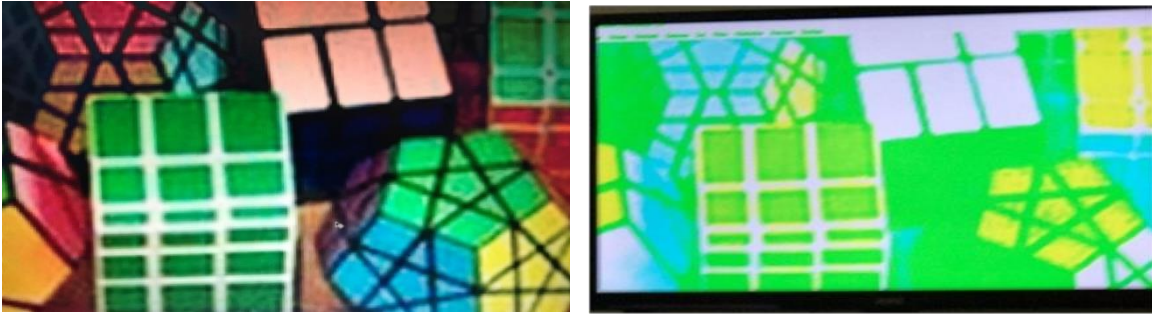
c) FPGA uygulamasından alınan görüntü.

Şekil 2. Yoğun renkler içeren görüntüler üzerinde köşe bulunması

Şekil 2(b) ve Şekil 2(c) kıyaslandığında FPGA üzerinde tespit edilen köşeler Matlab programında tespit edilen köşelere göre açıkça daha fazladır.

3.3. FPGA tabanlı aşındırma yöntemi ile morfolojik işlem uygulaması

Şekil 3(a)'da aşındırma algoritması uygulanmamış orijinal görüntü ve Şekil 3(b) 'de FPGA üzerinde aşındırma algoritması uygulanmış görüntü verilmiştir.



(a) Orijinal görüntü

(b) FPGA uygulamasından alınan görüntü

Şekil 3. Aşındırma yöntemi ile morfolojik işlem uygulaması

Şekil 3(a) ve Şekil 3(b)'ye bakıldığında aşındırma algoritması uygulanmış olan resmin içerisindeki dörtgenlerin arasındaki mesafenin aşınmadan dolayı bir miktar arttığı görülmüştür. Sonuçlar Tablo 1'de verilmiştir.

3.4. FPGA tabanlı yayma yöntemi ile morfolojik işlem uygulaması

Şekil 4(a)'da yayma algoritması uygulanmamış görüntü ve Şekil 4(b)'de FPGA üzerinde yayma algoritması uygulanmış görüntü verilmiştir.



(a) Orijinal görüntü.

(b) FPGA uygulamasından alınan görüntü.

Şekil 4. Yayma yöntemi ile morfolojik işlem uygulaması

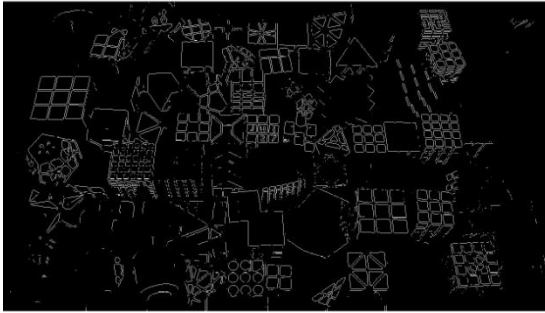
Şekil 4(a) ve Şekil 4(b)'ye bakıldığında yayma algoritması uygulanmış olan resmin içerisindeki dörtgenlerin arasındaki mesafenin yayılmadan dolayı bir azaldığı görülmüştür.

3.5. FPGA tabanlı Sobel görüntü işleme algoritması uygulanması

Çalışmanın bu kısmında, kenar belirleme uygulaması olan Sobel görüntü işleme algoritması, FPGA çipi üzerinde donanımsal olarak tasarlanmıştır. Tasarımdan elde edilen görüntüler, sırasıyla Şekil 5 (a)'da herhangi bir kenar bulma algoritması uygulanmamış orijinal görüntü, Şekil 5(b)'de Matlab üzerinde Sobel kenar algoritması uygulanmış görüntü ve Şekil 5(c)'de FPGA üzerinde Sobel kenar bulma algoritması uygulanmış görüntü olarak verilmiştir.



a) Orijinal görüntü.



b) Matlab uygulamasından alınan görüntü.



c) FPGA uygulamasından alınan görüntü.

Şekil 5. Matlab ve FPGA üzerinde sobel algoritmasının uygulanması.

Şekil 5(b) ve Şekil 5(c) kıyaslandığında FPGA üzerinde işlenmiş olan Şekil 5(c)'deki kenarların daha belirgin ve daha canlı olduğu görülmektedir. Matlab uygulamasında bulunan kenarların çokça keskin ve daha düzensiz olduğu fark edilmektedir. Bu durum ise görselin içerdiği cisimlerin anlaşılabilirliğini azaltmıştır. Sonuçlar Tablo 1'de gösterilmiştir.

3.6. FPGA tabanlı renk değiştirme yöntemi ile görüntü işleme algoritması uygulanması

Çalışmanın bu kısmında, renk değiştirme yöntemi algoritması, FPGA çipi üzerinde donanımsal olarak tasarlanmıştır. Tasarımdan elde edilen sonuçlar, Şekil 6(a) üzerinde renk değiştirme yöntemi uygulanmamış RGB görseli orijinal görüntü ve Şekil 6(b) FPGA üzerinde renk değiştirme yöntemi uygulanmış RGB görseli olarak verilmiştir. Sonuçlar Tablo 1'de gösterilmiştir.



(a) Orijinal görüntü.

(b) FPGA uygulamasından alınan görüntü.

Şekil 6. FPGA üzerinde renk değiştirme algoritmasının uygulanması.

3.7. FPGA tabanlı gerçekleştirilen görüntü işleme algoritmalarının test sonuçları

Tablo 3.1'de, yapılan çalışmalarda elde edilen sonuçların güç ve çip istatistikleri verilmiştir. Şekil 1'de Fast köşe algılama algoritmasının, Şekil 2'de Harris köşe algılama algoritmasının, Şekil 3'te Aşındırma yönteminin, Şekil 4'te Yayma yönteminin, Şekil 5'te Sobel algoritmasının, Şekil 6'da renk değiştirme görüntü işleme algoritmasının güç ve çip istatistikleri verilmiştir.

Tablo 1. Yapılan görüntü işleme algoritmalarının kullandığı güç ve hafıza istatistikleri.

Görüntü işleme algoritmaları	Güç ihtiyacı (W)	LUT (%)	FF (%)	BRAM (%)
Fast Köşe Algılama	2.145	31	23	30
Harris Köşe Algılama	2.651	71	42	48
Aşındırma Yöntemi	2.041	25	21	18
Yayma Yöntemi	2.036	24	20	16
Sobel Kenar Bulma	2.127	30	23	17
Renk Değiştirme	2.163	27	21	19

Bu verilere bakıldığında en az kaynak kullanan algoritmanın yayma yöntemi olduğu görülmektedir. Bu doğrultuda yayma yönteminin FPGA üzerinde daha hızlı çalışabileceği varsayılmaktadır. En fazla kaynak kullanan algoritmanın ise Harris köşe

algılaması olduğu görülmektedir. Bu sonuçlar ile bu uygulamanın daha yavaş ve yoğun çalışmadan ötürü FPGA çipinin fazla ısınmasından dolayı daha istikrarsız çalışması söz konusu olabilmektedir. Bu durumdan dolayı bu algoritmayı çalıştıracak olan FPGA çipi için iyi bir soğutucu tercih edilmelidir. Hafıza kullanımı sıralaması küçükten büyüğe doğru yapıldığında kenar bulma algoritmalarının daha yüksek çalışma alanına ihtiyaç duyduğu görülmektedir.

4. Sonuç ve öneriler

Sunulan bu çalışmada, Fast ve Harris köşe algılama algoritmaları, Sobel kenar bulma algoritması, morfolojik işlem algoritmalarından yayma ile aşındırma yöntemleri ve renk değiştirme algoritması FPGA çipleri üzerinde çalışmak üzere tasarlanmıştır. FPGA-tabanlı tasarlanan gerçek zamanlı görüntü işleme algoritmalarında Vivado Design Suite, Vivado High Level Synthesis ve Vivado SDK programları kullanılmıştır. Gerçek zamanlı görüntüler HDMI aracılığıyla kameradan alınmıştır. Alınan görüntü verilerinin FPGA üzerinde işlenebilmesi için VHDL kullanılmıştır. Yapılan tüm tasarımlar Xilinx Zybo Z7-20 geliştirme kartı üzerinde gerçekleştirilmiştir. FPGA-tabanlı tasarımların sonuçlarından elde edilen görüntü verileri HDMI aracılığıyla monitöre aktarılmıştır. Fast köşe algılama algoritması sonuçları, Matlab programının sonuçlarıyla karşılaştırılmıştır. Fast köşe algılama algoritmasının düşük ışıқта ve yüksek ışıқта bulunduğu köşeler Matlab programının düşük ışıқта ve yüksek ışıқта bulunduğu köşelerden daha fazla sayıda ve daha kesin olmaktadır. Harris köşe algılama algoritmasının küplerin üzerinde bulunduğu köşeler Matlab uygulamasının bulunduğu köşelerden daha fazla olmaktadır. Bu durum aynı görüntü üzerindeki ışık değişimlerinde daha belirgin olmakta ve Harris köşe bulma algoritması daha iyi sonuç vermektedir.

Bu çalışma değerlendirilirken Matlab uygulamasının köşe işaretçisinin Harris köşe bulma algoritmasının köşe işaretçisinden daha büyük olması sebebiyle göz yanılısına sebep olabilmektedir. Bu durum göz ardı edilmemelidir. Fakat FPGA-tabanlı Harris köşe algılama algoritmasının sonuçlarına bakıldığında bazı köşe olmayan noktaları işaretlediği görülmektedir. Bu durum yapılan tasarım için bir dezavantaj teşkil etmektedir. FPGA-tabanlı aşındırma algoritması uygulaması ve FPGA-tabanlı yayma algoritması uygulaması başarıyla gerçekleştirilmiştir. Sobel kenar bulma algoritmasında Matlab uygulaması sonuçları ile FPGA-tabanlı Sobel kenar bulma algoritması kıyaslandığında, FPGA-tabanlı çalışma kesin ve doğru biçimde sonuçlar göstermektedir ve kenarları belirten beyaz çizgilerin daha kararlı ve belirgin olduğu gözlemlenmiştir. FPGA-tabanlı Sobel kenar bulma algoritması görüntüde olan kenarları doğru ve tahribata uğratmadan tespit etmiştir. Bu algoritma literatürde uygulanan diğer algoritmalarından daha fazla piksel yoğunluğu işleyebilmektedir. Piksel yoğunluğu temel alındığında daha az BRAM kullanımı olduğu görülmektedir. FPGA-tabanlı renk değiştirme algoritması gerçek zamanlı olarak uygulanmış ve başarıyla gerçekleştirilmiştir.

Kaynaklar

- [1] Prakash, J., Dehghani, H., Pogue, B. W. ve Yalavarthy, K., Model-Resolution-Based Basis Pursuit Deconvolution Improves Diffuse Optical Tomographic Imaging, **IEEE Transaction on Medical Imaging**, 33, 4, 891-901, (2019).
- [2] Rana, K. B., Agrawal, G. D., Mathur, J. ve Puli, U., Measurement of void fraction in flow boiling of ZnO water nanofluids using image processing technique, **Nuclear Engineering and Design**, 270, 217-226, (2014).
- [3] Gürevin, B., Yıldız, M., Güleryüz, E., Kutlu, M. ve Sorgun, Ö., A Chaos Based Image Encryption On LabVIEW, **Chaos Theory and Applications**, 2, 2, 69-76, (2020).
- [4] Lu, W., Lifan, Z., Guoan, B., Chunru, W. ve Lei, Y., Enhanced ISAR Imaging by Exploiting the Continuity of the Target Scene, **IEEE Transactions on Geoscience and Remote Sensing**, 52, 9, 5736-5750, (2014).
- [5] Xiao, Y. ve Dong, S., Multilevel-Based Topology Design and Cell Patterning with Robotically Controlled Optical Tweezers, **IEEE Transactions on Control Systems Technology**, 23, 1, 176-185, (2015).
- [6] Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O. ve Acharya, U. R., Automated detection of COVID-19 cases using deep neural networks with X-ray images. **Computers in Biology and Medicine**, 121, 103792, (2020).
- [7] Yu, Y. H. ve Chang, C. C., A New Edge Detection Approach Based On Image Context Analysis, **Image And Vision Computing**, 24, 10, 1090-1102, (2006).
- [8] Gonzales, R. C. ve Woods, R. E., **Digital Image Processing**, 722, Prentice-Hall, New Jersey, (2002).
- [9] Harris, C. ve Stephens, M., A Combined Corner And Edge Detector, **Proceedings of the 4th Alvey Vision Conference**, 147-151, (1988).
- [10] Koyuncu, I., Özcerit, A. T. ve Pehlivan, I., Implementation of FPGA-based Real Time Novel Chaotic Oscillator, **Nonlinear Dynamics**, 75, 1-2, 49-59, (2014).
- [11] Kumar, S. ve Prabat, P., FPGA Implementation of Image Segmentation By Using Edge Detection Based On Sobel Edge Operator, **International Journal of Research in Engineering and Technology**, 2, 10, 198-203, (2013).
- [12] Mehra, R. ve Rupinder, V., Area Efficient FPGA Implementation of Sobel Edge Detector for Image Processing Applications, **International Journal of Computer Applications**, 56, 16, 7-11, (2012).
- [13] Nausheen, N., Seal, A., Khanna, P. ve Halder, S., A FPGA based implementation of Sobel edge detection, **Microprocessors and Microsystems**, 56, 84-91, (2018).
- [14] Karaköse, M., Baygın, M., Aydın, İ., Sarımaden, A. ve Akın, E., Endüstriyel Sistemlerde Arkaplan Çıkarımı Tabanlı Hareketli Nesne Tespiti ve Sayılması için Yeni Bir Yaklaşım, **Muş Alparslan Üniversitesi Fen Bilimleri Dergisi**, 4, 2, 373-381, (2016).
- [15] Aydoğdu, M. F., Demirci, M. F. ve Kasnakoğlu, C., Pipelining Harris Corner Detection with a Tiny FPGA for a Mobile Robot, **Proceeding of the IEEE International Conference on Robotics and Biomimetics**, Shenzhen, China, 2177-2184, (2013).
- [16] Gacar, A., FPGA Tabanlı Görüntü İşleme Arabirimi, Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, İzmir, (2009).
- [17] Özçelik, M. F., Görüntü İşleme Algoritmalarının FPGA Üzerinde Gerçeklenmesi, Yüksek Lisans Tezi, Gazi Üniversitesi, Bilişim Enstitüsü, Ankara, (2012).

- [18] Çil, M. M., Temel Görüntü İşleme Algoritmalarının FPGA Üzerinde Gerçeklenmesi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, (2015).
- [19] Kızılkaya, R., Implementation Of Image Processing Algorithms On FPGA Demonstration Board, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, İzmir, (2012).
- [20] Çelik, A. R., Görüntü İşleme Algoritmalarının FPGA Donanımı Üzerinde Gerçeklenmesi, Yüksek Lisans Tezi, Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Kahramanmaraş, (2013).
- [21] Özalp, R., Çip Üzerinde Sistem Mimarili FPGA Kullanarak Gerçek Zamanlı Görüntü İşleme Algoritmalarının Gerçekleştirilmesi, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ (2018).
- [22] Altuncu, M. A., Temel Görüntü İşleme Algoritmalarının Gerçek Zamanlı Olarak FPGA ile Gerçeklenmesi, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, (2015).
- [23] Jing, H. ve Xiaoqiong, X., Sports image detection based on FPGA hardware system and particle swarm algorithm, **Microprocessors and Microsystems**, 80, 103348, (2021).
- [24] Selvaganesh, M., Esakki-Vigneswaran, E. ve Vaishnavi. V., FPGA Implementation of Low Latency and Highly Accurate Median Filter Architecture for Image Processing Applications, **Inventive Systems and Control**, 805-816, Springer, Singapore, (2021).
- [25] Arshad, A., Shaukat S., Ali, A., Eleyan, A., Shah, S. A. ve Ahmad, J., An Essential Framework for Image Encryption, **Chaos Theory and Applications**, 2, 1, 17-22, (2020).
- [26] Trajkovic, M. ve Hedley, M., Fast corner detection, **Image and Vision Computing**, 16, 75-87, (1998).
- [27] Rosten, E. ve Drummond, T., Machine Learning for High-Speed Corner Detection, **European Conference on Computer Vision**, 430-443, United Kingdom, (2006).
- [28] Cuevas, E., Zaldivar, D., Pérez-Cisneros, M., Sánchez, E. ve Ramírez-Ortegón, M., Robust Fuzzy Corner Detector, **Intelligent Automation and Soft Computing**, 17, 415-429, (2011).
- [29] Harris, C. ve Stephens, M., A combined corner and edge detector, **Proceedings of the 4th Alvey Vision Conference**, 147-151, (1988).
- [30] Aybar, E., Sobel İşleci Kullanılarak Renkli Görüntülerde Kenar Bulma, **Afyon Kocatepe Üniversitesi Fen Bilimleri Dergisi**, 8, 205-217, (2008).
- [31] Koyuncu, İ., Tuna, M. ve Alçın, M., FPGA tabanlı farklı nümerik algoritmalar ile kaotik osilatör tasarımları, **International Eurasian Conference on Science, Engineering and Technology**. 2532-2541, Ankara, Turkey, (2018).
- [32] Koyuncu, İ., Implementation of High Speed Tangent Sigmoid Transfer Function Approximations for Artificial Neural Network Applications on FPGA, **Advances in Electrical and Computer Engineering**, 18, 3, 79-86, (2018).
- [33] Alçın, M., Pehlivan, İ. ve Koyuncu, İ., Hardware design and implementation of a novel ANN-based chaotic generator in FPGA. **Optik-International Journal for Light and Electron Optics**, 127, 13, 5500-5505, (2018).