

Çoklu Ekran Kartı ile Hızlandırılmış Ayrık Haar Dalgacık Dönüşümü Temelli Görüntü Sıkıştırma

Selçuk ASLAN¹, Hasan BADEM¹, Tayyip ÖZCAN¹, Derviş KARABOĞA¹, Alper BAŞTÜRK¹

¹Erciyes Üniversitesi, Bilgisayar Mühendisliği, Kayseri, Türkiye

ÖZET: Dalgacık (wavelet) fonksiyonları, sinyal işleme uygulamalarında oldukça önemli olan dönüşümlerin temelini oluşturur. Ardışık piksel çiftlerine ait değerlerin ortalama ve farklarının alınmasına dayalı temel yaklaşımının getirdiği işlem kolaylığı sebebi ile ayrık Haar dalgacık dönüşümü, sahip olduğu çok sayıda izlek ile uygun paralelleştirme senaryolarında çalışma sürelerini önemli oranlarda azaltan grafik kartların kullanımının önünü açmaktadır. Bu çalışmada iki boyutlu ayrık Haar dalgacık dönüşümü, Message Passing Interface (MPI) kütüphanesi ve Compute Unified Device Architecture (CUDA) programlama mimarisi kullanılarak işlemci ve grafik kartlar kümelerine aktarılmıştır. Farklı boyutlu resimlerin iki boyutlu ayrık Haar dalgacık dönüşümü ile işlenmesi için gereken sürenin, dönüşümün tek işlemci ve tek grafik kartlı eş değerlerine kıyasla yürütme zamanını önemli oranda iyileştirdiği görülmüştür.

Anahtar Kelimeler: *Ayrık Haar Dalgacık Dönüşümü, Grafik Kartı Kümeleme, Paralel Hesaplama*

Image Compression with Multi-GPU Accelerated Discrete Haar Wavelet Transform

ABSTRACT: Wavelet functions play a vital role in the vast majority of the transforms used in the signal processing applications. The easy implementation of the discrete Haar wavelet transform by taking the average and difference of the sequential pixel values has increased the usability of the graphics card with multiple threads on suitable parallelized scenarios. In this paper, discrete Haar wavelet transform ported in a compute-efficient way to CPU and GPU clusters by utilizing Message Passing Interface (MPI) and Computer Unified Device Architecture (CUDA), respectively. Experimental studies conducted as part of the parallelization strategies for two-dimensional discrete Haar wavelet transform showed that the total running time required to process all rows and columns of an image with different size is significantly decreased when compared its counterparts on single CPU and single GPU.

Keywords: *Discrete Haar Wavelet Transform, GPU Cluster, Parallel Computing*

1. Giriş

Zaman dizileri, belirli bir sürecin sonunda örneklenen veya ölçümü yapılan değere ilişkin gözlemler topluluğu olarak nitelenebilir. Zaman dizilerinin daha iyi yorumlanması ve temsil ettikleri sistemin özelliklerinin belirlenebilmesi adına, dizinin zaman alanından frekans alanına aktarılması veya dönüştürülmesi gerekmektedir. Zaman uzayından frekans uzayına geçişi sağlayacak çok sayıda matematiksel yaklaşım olmakla birlikte, Fourier, Hilbert, Radon dönüşümleri ile son zamanlarda uygulama sahası oldukça genişleyen görece yeni ve ilk kez Alfred Haar'ın doktora tezinde bahsettiği dalgacık sözcüğü ile anılan Dalgacık dönüşümü bunlardan bazılarıdır [1]. Dalgacık dönüşümü, Fransız matematikçi Joshep Fourier tarafından saptanmış bütün periyodik fonksiyonların, ağırlıklandırılmış basit trigonometrik fonksiyonların toplamı şeklinde ifade edilebileceği temeline dayanır [1].

Dalgacık dönüşümü başlangıçta sismik verilerin analizi amacıyla sunulmuş olmasına karşın, işaret işlemeden yapay sinir ağlarına, veri madenciliğinden kısmi diferansiyel denklemlerin çözümüne kadar çok sayıda değişik uygulama sahalarında kullanılmıştır [1,2]. Dönüşümün, değişimi oldukça fazla sinyallerin analizi dışında önemli bir sonucu olan sıkıştırma özelliği, büyük boyutlu resim ve video verilerinin işleme sahasında da geniş kullanım alanı bulmasına olanak sağlamıştır [2,3]. Güncel sayılabilecek resim sıkıştırma standartlarından biri olan JPEG-2000 ile önemli bir video kodlama standardı haline gelmiş MPEG-4 dalgacık tabanlı algoritmalar kullanılmaktadır [2,3].

Yüksek çözünürlüklü, gerçek-zamanlı uygulamalara artan talebe bağlı olarak, depolama, resimlerin depolanma, kullanım ve işleme süreleri oldukça artmıştır. Bu dijital verilerin kayıpsız olarak depolanması ve talep halinde işlenip gönderilmesi, mevcut matematiksel modeller ve bu modellerin paralelleştirilmiş uygulamalarının önemini daha da

*Sorumlu Yazar: Selçuk Aslan selcukaslan@erciyes.edu.tr

artırmaktadır. Resim verisinin sunumu ve dönüşümün piksel değerleri üzerinde görece basit aritmetik işlemlerinin sıralı yapıma zorunluluğunun olmaması, paralel veri işleme kapasiteleri sebebi ile grafik kart tabanlı uygulamaların geliştirilmesinin önünü açmıştır. Wong ve diğerleri giriş seviyesi ekran kartları ile JPEG-2000 sıkıştırma algoritmasını önemli oranlarda hızlandırmıştır [4]. 2008 yılında da Tenllado ve diğerleri iki boyutlu ayrık dalgacık dönüşümünün sıkıştırma performansını artırılması ve farklı dalgacık fonksiyonlarına ait paralel modellerinin karşılaştırılması amacı ile programlanabilir grafik kartlarını kullanmışlar [5], %10 ile %140 arasında değişen oranlarda çalışma performansı iyileştirilmiştir [5]. Sonrasında Franco ve diğerleri ile Van der Laan ve diğerleri doğrudan CUDA platformunu kullanarak farklı dalgacıklar üzerinde dönüşüm performansını çeşitli üretici firmalara ait işlemciler üzerinde çalıştırılmış seri uygulamalar ile karşılaştırmışlardır [6,7]. Franco ve diğerleri, Nvidia Tesla c870 grafik kartı ile 20 kat performans artışı sağlarken [6] Laan ve diğerleri, satır-sütun ve blok temelli metodlardan yola çıkarak yeni bir hibrit metod önermiş, iki ve üç boyutlu resim verilerinin dalgacık temelli dönüşümlerinde işlemci temelli seri uygulamalara kıyasla oldukça çalışma sürelerini oldukça kısalttığı görülmüştür [7].

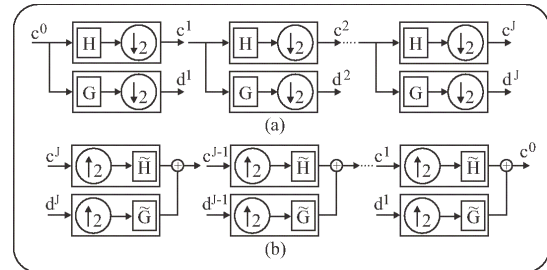
Bu çalışmada, iki boyutlu ayrık Haar dalgacık dönüşümünün, grafik kart destekli hesaplama kümesi üzerinde çalıştırılmak üzere önce MPI (Message Passing Interface) sonrasında hesaplama düğümleri üzerindeki grafik kartları için CUDA paralel hesaplama mimarisine aktarıldığı paralel uygulaması yapılmıştır. İki seviyeli paralel dönüşümün, işlemci ya da izlek temelli paralelleştirme senaryolarında, işlemci veya izlek sayısı ile ilişkilendirilebilir hızlandırma ve verimlilik metriklerinin, grafik kart kümelerinde işlenecek veri boyutlarına bağlı olarak değişim gösterip göstermeyeceğinin belirlenmesi adına farklı boyutlarda resimler kullanılmıştır. Çalışmanın devamında, ayrık Haar dalgacık dönüşümüne ait teorik bilgi verilmiş, üçüncü bölümde MPI ve CUDA platformlarının temel özelliklerinden bahsedilmiş, dördüncü bölümde seri ve önerilen paralel senaryosuna ait farklı boyutlardaki resim verilerinin üzerinde gerçekleştirilen test sonuçları verilmiş, son bölümde dönüşümün iki seviyeli paralelleştirme senaryolarındaki performansı ve yapılması muhtemel çalışmalardan bahsedilmiştir.

2. Ayrık Haar Dalgacık Dönüşümü

Dalgacık dönüşümü temelde, kompleks fonksiyonların, temel dalgacık olarak isimlendirilecek prototip fonksiyonun ölçeklenme ve ötelenmesiyle elde edilecek daha basit fonksiyonları birleşimi şeklinde ifade edilebilmesi durumunu referans almaktadır [1-4]. Haar dalgacığı bahsedilen temel dalgacık fonksiyonlarından en eski ve basit olanıdır [1-4]. Diğer dalgacık dönüşümlerinde olduğu gibi, Haar dalgacık dönüşümü de

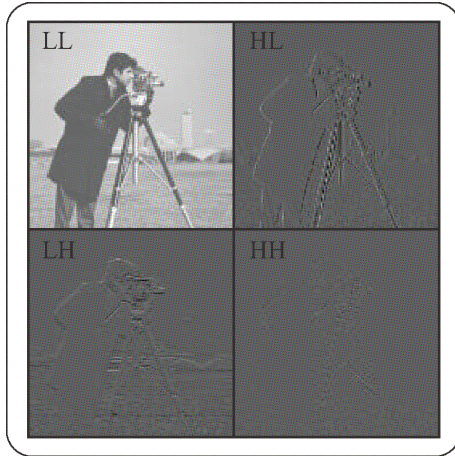
ayrık bir sinyali, eleman sayıları dönüşüm uygulacak ayrık sinyalin eleman sayısının yarısı olan yaklaşım ve detay adı verilen iki sinyal bandına ayırır. N , 2'nin kuvveti olan bir pozitif tam sayı olmak üzere, N elemanlı tek boyutlu ayrık sinyalin $c^0 = (c_0^0, c_0^1, \dots, c_0^{N-1})$ $c = c^l$

yaklaşım ve d^l detay bandlarının elde edilmesi için Haar dalgacık dönüşümü, H ve G ile gösterilmiş alçak-geçiren ve yüksek-geçiren filtre bandlarının uygulanması ile sağlanır. C_0 sinyalinin filtre bandından geçirilmesi ile bulunan c^l yaklaşım bandı, c^2 ve d^2 ile gösterilen ikinci seviye yaklaşım ve detay sinyallerinin elde edilmesi için bir kez daha alçak ve yüksek geçiren filtre bandlarından geçirilir. J dönüşümün seviyesini göstermek üzere, aşağıda Haar dalgacık dönüşümünün J defa uygulanması, dönüşüm tamamlandıktan sonra c_0 sinyalinin yaklaşım ve detay sinyallerinden tekrar elde edilmesi gösterilmektedir.



Şekil 1. İleri (a) ters (b) dalgacık dönüşümü

İki boyutlu imge verisinin Haar dalgacık dönüşümü, tek boyutlu ayrık sinyalin Haar dalgacık dönüşümü ile işlenmesi durumunun iki boyut referans alınarak geliştirilmiş halidir [1-4]. Ayrık Haar dalgacık dönüşümünün resim verisi üzerinde uygulanması, önce satır sonra da tüm sütunlarda alçak ve yüksek geçiren filtre bandının her dönüşüm seviyesine bağlı olarak ardışık uygulanmasına dayanır [1-4]. Ayrık Haar dalgacık dönüşümünün birinci derecede uygulandığı imge, her biri orijinal verinin satır-sütün çözünürlüğünün yarısı olan dört farklı banda ayrılır. Bu bantlardan sol alttaki alt bant, alçak-yüksek bandı (Low-High), satırlara alçak-geçiren, sütunlara yüksek-geçiren filtrenin uygulanması sonucu bulunmuştur. Sağ alttaki alt bant, yüksek-yüksek bandı (High-High), satır ve sütunlara yüksek-geçiren filtrenin uygulanmasıyla bulunmuştur. Üst sol köşedeki alt bant, alçak-alçak bandı (Low-Low), satır ve sütunlara alçak-geçiren filtrenin uygulanması ile bulunmuş iken, üst sağ alt köşedeki bant, yüksek-alçak bandı (High-Low), satırların yüksek-geçiren filtre, sütunların ise alçak-geçiren filtre ile işlenmesi sonucu elde edilmiştir. Aşağıda, tek seviye ayrık Haar dalgacık dönüşümü uygulanarak, bahsedilen alt bantların elde edilmesi örneklendirilmiştir. Dönüşümün diğer seviyedeki bantlarına, orijinal resim verisinin sıkıştırılmış hali olan alçak-alçak filtrelenmiş bölgeye ait işaretler üzerinden ulaşılabilir.

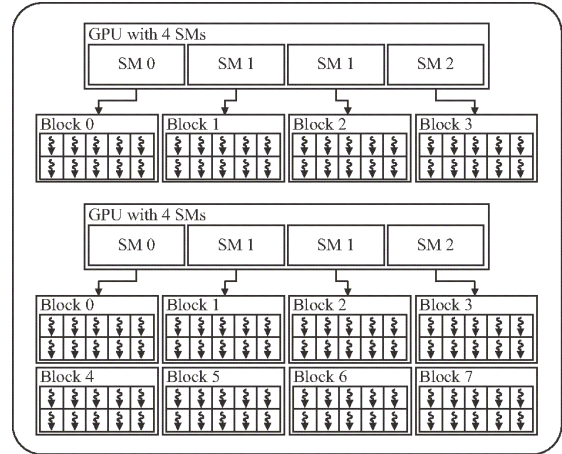


Şekil 2. Görüntüdeki yaklaşım ve detay alt bantları

3. CUDA Platformunun Temel Özellikleri

Kullanıcı taleplerinin gerçek zamanlı, oldukça yüksek çözünürlüklü grafikler üzerinde yoğunlaşması, çok sayıda paralel çalışabilir çekirdeğe sahip yüksek işlem gücü ve band genişlikli grafik işleme birimlerinin geliştirilmesinin önünü açmıştır [8-12]. Grafik işleme birimlerinin bu özellikleri, merkezi işlem birimlerinde, grafik işleme birimlerine kıyasla daha çok transistör iş akış kontrolü ile veri depolama için ayrılmıştır, aritmetik işlem miktarının veri taşıma, kopyalama işlem miktarına oranının daha yüksek olduğu sinyal işleme, modelleme ve biyoinformatik problemlerinin çözüldüğü pek çok uygulamada hesaplama maliyetlerini düşürmüştür [8-12].

2006 yılında NVIDIATM firması, ürettikleri grafik işleme birimlerinin, paralelleştirilebilirliği yüksek ve veri bağımlılığı az kompleks problemlerin çözümünde kullanılabilmesi için CUDA adını verdikleri genel paralel hesaplama, programlama platformunu, C gibi üst seviye bir programlama dilinin de aralarında bulunduğu pek çok dil desteği ile uygulama geliştiricilere duyurmuştur [8]. CUDA programlama platformu, grafik işleme biriminde farklı CUDA thread yapısı aracılığı ile kez paralel çalıştırılacak fonksiyonu kernel olarak isimlendirmektedir [8-12]. Probleme ve çalıştırılacağı cihazın sınırlamalarına bağlı olarak thread elemanları tek, iki veya üç boyutlu thread bloklarından oluşmuş, yine tek, iki veya üç boyutlu grid yapıları şeklinde organize edilmektedir. CUDA uygulamasından kernel çağrıldığında, kernel için belirlenmiş thread blokları eş zamanlı çalışabilme özelliğine sahip Streaming Multiprocessor (SM) birimlerine dağıtılır. SM birimleri de kendi içlerinde Single Instruction Multiple Thread (SIMT) adı verilen bir mimariyi kullanarak çok sayıda izlek elemanını paralel çalıştırabilecek özellikte inşaa edilmiştir [9]. Aşağıda dört SM birimi üzerine dört ve sekiz izlek bloğunun dağıtılması durumu örneklenmiştir.



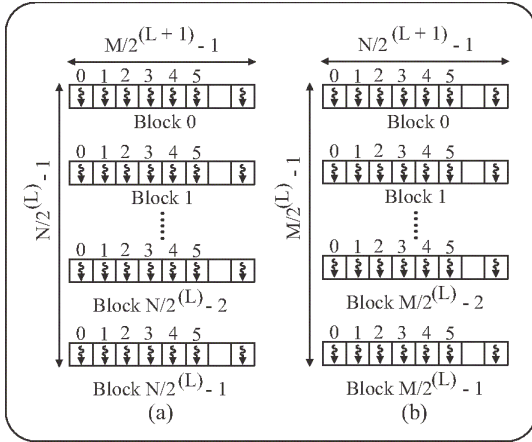
Şekil 3. Blokların kart üzerinde dağıtılması

4. Ayrık Haar Dalgacık Dönüşümünün Grafik Kart Kümesi için Düzenlenmesi

İki boyutlu ayrık Haar dalgacık dönüşümünün grafik kart destekli bir hesaplama kümesi için düzenlenmesi, önce her bir işlemci-hafıza ikilisinden oluşan hesaplama düğümü sonrasında düğümde bulunan grafik kartları için iki seviyeli bir paralelleştirme senaryosunun uygulanmasını gerektirir. Önce resim verisi, kullanılacak düğüm sayısı kadar eşit boyuta ayrılmalı, her düğüme aktarılan veri grafik kartının hafıza uzayına taşınmalıdır.

MPI kütüphaneleri kullanılarak hesaplama düğümlerine eşit olarak bölünmüş resim verisi dağıtıldıktan sonra, her düğüm üzerindeki grafik kartı üzerinde işlenecek resim bölümleri için ikinci seviye paralelleştirme senaryosuna geçilebilir. İki boyutlu ayrık Haar dalgacık dönüşümünün önce tüm satırlar sonrasında da tüm sütunlar üzerinden geçirilen alçak ve yüksek geçiren filtre çifti ile yapılıyor olması, aynı dönüşüm seviyesinde tüm satır ve sütunlarda filtreleme işleminin paralel olarak gerçekleştirileceği iki farklı CUDA kernel tasarımını mümkün kılmaktadır. Bütün satırlar üzerinde filtrenin paralel geçirildiği kernel için her bir blok bir satır, bütün sütunlar üzerinde filtrenin paralel geçirildiği kernel için de her bir blok bir sütünü işleyen izlek kümelerinden oluşmaktadır. L uygulanacak ayrık Haar dalgacık dönüşümünün seviyesi, M ve N sırasıyla grafik kartına aktarılan verinin sütun ve satır sayısı olmak üzere, herbiri $M/2^{(L+1)}$ izlek bulunduran $N/2^L$ blok konfigürasyonlu fonksiyon karta aktarılan verinin satırlarını işleyerek yaklaşım ve detay bandlarını bulmak üzere kullanılmıştır. Satır, sütun ve dönüşüm seviye bilgisi kullanılarak benzer bir ölçkleme, tüm satırlar işlendikten sonra sütunların işleneceği grafik kart fonksiyonu için de tanımlanabilir. Buna göre, $N/2^{(L+1)}$ izlek bulunduran $M/2^L$ blok şeklinde düzenlenmiş fonksiyon, sütunların işlenmesi için kullanılmaktadır. Aşağıda, satır, sütun ve dönüşüm seviyesi arasındaki

bağıntıyı gösterir izlek-blok ölçekleme modeli gösterilmektedir.



Şekil 4. Satır (a) ve sütun (b) çediklerinin izlek yapısı

Grafik kartı ile işlemci-hafıza çifti arasındaki veri transferinin minimizasyonu, CUDA uygulamalarında performansın artırılması için dikkate alınacak ilk durumdur. Grafik kartlar ile güçlendirilmiş hesaplama kümesinde, düğümlere aktarılmış resim eşit boyutlu resim parçaları, bir kez düğüm hafızadan grafik kartın hafıza uzayına aktarılmış, belirtilen dönüşüm seviyesi tamamlandıktan sonra grafik kart hafızasından düğüm hafızasına aktarılmıştır. Algoritma 1'de, işlenecek resimin düğümlere dağıtılması, sonrasında grafik kartlarında işlenerek düğüm hafızalarına aktarılması ve seçilmiş düğümlerden biri tarafından birleştirilmesi durumlarını gösteren işlem adımları verilmiştir

Algoritma 1 Paralel dönüşümün temel adımları

```

1: MPI_Comm_size( MPI_COMM_WORLD, &size )
2: MPI_Comm_rank( MPI_COMM_WORLD, &rank )
3: if rank % size == 0 then
4:   Read the entire MxN image.
5:   Equally divide image to Mx (N/size) sized sub-parts.
6:   Scatter parts of image to 0... (size - 1) ranked GPU nodes.
7: end if
8: Copy Mx (N/size) image part to GPU global memory.
9: for L = 0...TL do
10:  processRows<<<M/2^L, (N/size) / 2^(L+1)>>>( ... )
11:  processColumns<<<N/2^L, (M/size) / 2^(L+1)>>>( ... )
12: end for
13: Copy Mx (N/size) image part back to host memory.
14: if rank % size == 0 then
15:   Gather parts of image from 0... (size - 1) ranked GPUs.
16:   Combine image fragments into MxN image.
17:   Write the entire MxN image.
18: end if

```

5. Deneysel Sonuçlar

İki boyutlu ayrık Haar dalgacık dönüşümünün seri ve paralel uygulamaları, dört hesaplama düğümü içeren hesaplama kümesi kullanılarak yapılmıştır. Hesaplama düğümlerinin her biri çalışma frekansları 3.4 ile 3.8 GHz arasında değişen Intel I5 4670 kodlu işlemci ve Asus Geforce GTX 780 Direct CUII grafik kartına sahiptir. Hesaplama düğümleri arasındaki haberleşme standart Gigabit ethernet bağlantısı ile sağlanmaktadır. Boyutları

512x512, 1024x1024 ve 2048x2048 olan üç farklı test verisi, dört seviye yani orijinal resim boyutunu on altı kat küçültecek şekilde, iki boyutlu ayrık Haar dalgacık dönüşümü ile on farklı kez bağımsız olarak tek işlemci, çoklu işlemci, tek grafik kartı ve çoklu grafik kartı senaryolarında işlenmiş, dönüşümün tamamlanması için gereken süre milisaniye olarak kaydedilmiş ve on farklı koşmanın ortalaması alınarak Tablo 1'de verilmiştir.

Tablo 1: Farklı hesaplama ortamlarında bulunan ortalama yürütme zamanları

Resim Boyutu	Hesaplama Ortamı			
	Tek CPU	Tek GPU	CPU Kümesi	GPU Kümesi
512x512	4.0102	0.4479	1.2517	0.2923
1024x1024	23.6937	1.3701	9.6567	0.5213
2048x2048	100.596	5.0491	35.2685	1.4427

Tablo 1'de verilen sonuçlar incelendiğinde, dört düğümün birlikte kullanıldığı grafik kart küme senaryosunda, orijinal imgeye kıyasla on altı kat sıkıştırılmış durumun dönüşümle elde edilmesi için gereken süre, veri kopyalama maliyetleri grafik kartlar için toplam dönüşüm süresine dahil edilmiştir, önemli oranda azaltılmıştır. Resim boyutu 512x512'den 1024x1024'e, 1024x1024'ten 2048x2048'e dörder kat arttırılırken, grafik kartlara ait hesaplama kapasitesi daha iyi kullanıldığından diğer senaryolardaki işleme süresindeki artış, grafik kart kümesi için daha düşük oranlarda gerçekleşmiştir.

Paralel senaryoların seri uygulama ya da uygulamalar ile karşılaştırılmasında sıklıkla kullanılan metrikler arasında hızlandırma (speedup) ve verimlilik (efficiency) gelmektedir. Hızlandırma, seri uygulamada ölçülen yürütme zamanının paralel uygulamada ölçülen yürütme zamanına oranı iken alabileceği en büyük değer kullanılan düğüm sayısına eşittir. Verimlilik değeri ise, hızlandırma değerinin düğüm sayısına oranı şeklinde tanımlanmış olup alabileceği en büyük değer bire eşittir. Tablo 2'de çoklu grafik kartların kullanıldığı senaryoya kıyasla elde edilmiş hızlandırma verilmiştir.

Tablo 2. Dört düğümün kullanıldığı grafik kart kümesi ile elde edilen hızlandırma oranları

Resim Boyutu	Hesaplama Ortamı		
	Tek CPU	Tek GPU	CPU Kümesi
512x512	13.719x	1.532x	4.282x
1024x1024	45.451x	2.628x	18.524x
2048x2048	69.727x	3.499x	24.446x

Grafik kart kümesi ile yapılan dönüşüme ait hızlandırma ve verimlilik değerleri bulunurken, tek işlemcili ve çok işlemcili senaryo yerine, tek grafik kartlı senaryonun doğrudan kullanılması sonuçların değerlendirilmesi adına daha uygun olacaktır. Tablo 3'de bu değerler sadece grafik kart ve grafik kart kümesi tabanlı paralel senaryolar üzerinden tekrar verilmiştir. Sonuçlar incelendiğinde, işlemci temelli paralelleştirme

senaryolarında da bir miktar etkili olduğu görülen veri boyutunun, grafik kartlar ile kurulacak hesaplama kümelerinde verimlilik metriği üzerinde daha önemli bir belirleyici olduğu görülmektedir. 512x512 boyutlu imge üzerinde, tek grafik kartı 0.4479 ms ortalama süresi ile dönüşümü tamamlar iken dört düğümlü grafik kart kümesi ile elde edilen ortalama yürütme zamanı 0.2923 ms, verimlilik değerinin 0.5'in altında kalmasına sebep olmuştur. Ancak veri boyutu arttıkça bu oran sırasıyla 0.66 sonrasında 0.88'e ulaşarak elde edilebilir en yüksek değer olan 1'e önemli oranda yaklaşmıştır. Verimlilik değeri üzerinden durum dikkate alındığında, grafik kart kümeleri üzerinde yapılacak hızlandırma işlemlerinde veri boyutunda doğrudan dikkate alınması gerektiğini göstermektedir.

Tablo 3. Dört düğümün kullanıldığı grafik kart kümesi ile tek ekran kartı kullanım durumuna karşı elde edilen hızlandırma ve verimlilik değerleri

Resim Boyutu	Hesaplama Ortamı		Hızlandırma	Verimlilik
	GPU	GPU Kümesi		
512x512	0.4479	0.2923	1.532x	0.383
1024x1024	1.3701	0.5213	2.628x	0.657
2048x2048	5.0491	1.4427	3.499x	0.874

6. Sonuçlar

Bu çalışmada, önce MPI sonrasında CUDA platformları kullanılarak iki seviye paralelleştirilen ayrık Haar dalgacık dönüşümünün grafik kart destekli hesaplama düğümlerinden oluşan bir küme ile farklı boyutlu resimler üzerinden seri uygulamaya kıyasla dönüşüm için gerekli işlem süresini kısaltma performansı üzerine bir analiz yapılmış, çoklu grafik kartlarının uygun problem ve paralelleştirme senaryoları altında çalışma sürelerini seri eşdeğerlerine kıyasla oldukça azalttığı görülmüştür. Elde edilen sonuçlardan yola çıkılarak, veri transferi ya da hesaplama kaynaklarının ekran kartlarında yüksek verimlilikte kullanılmadığı durumlarda elde edilen performans artışının, tek kart üzerinde bütün bir imgenin işlendiği duruma kıyasla istenilen seviyelerde olmadığı görülmüştür.

7. Kaynaklar

- [1]. R. Polikar, "The Story of Wavelets", Physics and Modern Topics in Mechanical and Electrical Engineering, N. Mastorakis (Eds.), World Scientific and Electrical Engineering Society, 1999.
- [2]. P. Porwik, A. Lisowska, "The Haar-Wavelet Transform in Digital Image Processing: Its Status and Achievements", Machine Graphics & Vision, 2004, s. 79-98.
- [3]. K.H. Talukder, K. Harada, "Haar Wavelet Based Approach for Image Compression and Quality

- Assessment of Compressed Image", International Journal of Applied Mathematics, 2007, s. 1-9.
- [4]. T.T. Wong, C.S. Leung, P.A. Heng ve J. Wang, "Discrete Wavelet Transform on Consumer-Level Graphics Hardware", IEEE Transactions on Multimedia, 2007, s. 668-673.
- [5]. C. Tenllado, J. Setoain, M. Prieto, L. Pinuel ve F. Tirado, "Parallel Implementation of the 2D Discrete Wavelet Transform on Graphics Processing Units: Filter Banks versus Lifting", IEEE Transaction on Parallel and Distributed Systems, 2008, s. 299-310.
- [6]. J. Franco, G. Bernabe, J. Fernandez ve M.E. Acacio, "A Parallel Implementation of the 2D Wavelet Transform Using CUDA", 17th Euromicro International Conference Parallel, Distributed and Network-based Processing, Weimar, 2009, s. 111-118.
- [7]. W.J. Van der Laan, A.C. Jalba, B.T.M. Roerdink, "Accelerating Wavelet Lifting on Graphics Hardware Using CUDA", IEEE Transactions on Parallel and Distributed Systems, 2011, s. 132-146.
- [8]. NVIDIA, NVIDIA CUDA C Programming Guide V. 5.5, NVIDIA Corporation, 2013.
- [9]. NVIDIA, NVIDIA CUDA C Best Practices Guide V. 5.5, NVIDIA Corporation, 2013.
- [10]. S. Cook, CUDA Programming A Developer's Guide to Parallel Computing with GPUs, Morgan Kaufmann, 2013.
- [11]. D. B. Kirk, W. W. Hwu, "Programming massively parallel processors: A Hands-on Approach", Burlington, USA, 2010.
- [12]. J. Sanders, E. Kandrot, "CUDA by example: an introduction to general-purpose GPU programming", Pearson Education, Boston, USA, 2011.