

Foton Haritalama Algoritmasının Evrensel Aydınlatma Amaçlı Paralleştirilmesi

Parallelization of Photon Mapping Algorithm for Global Illumination

Yusuf Yavuz, Süleyman Tosun

Bilgisayar Mühendisliği Bölümü

Ankara Üniversitesi

yyavuz@eng.ankara.edu.tr, tosun@eng.ankara.edu.tr

Özet

Işığı simüle ederek fotoğraf gerçekliğinde sahneler üretme işlemi, bilimsel görselleştirme, film endüstrisi, oyun endüstrisi ve mimari modelleme gibi birçok önemli sektörde geniş kullanım alanlarına sahip olmasından dolayı bilgisayar grafiğinde önemli bir yere sahiptir. Işın izleme, ısı transferi ve foton haritalama gibi evresel aydınlatma algoritmaları karmaşık hesapsal işlemler ve donanımsal kısıtlar nedeniyle hız konusunda sıkıntı çekmektedir. Bu makalede foton haritalama algoritmasına ait iş parçacıklarının işlem birimleri üzerinde dağıtılarak gerçekleştirme zamanının azaltılmasına yönelik bir paraleleştirilme tekniği ortaya konmuştur. Ortaya konulan paraleleştirilme tekniği grid hesaplama sistemi tabanlıdır ve ileriki çalışmalara yol gösterecek sonuçlar ortaya konmuştur.

Abstract

Simulating light behavior and photorealistic image synthesis plays a central role in computer graphics since it has wide range of application areas such as scientific visualization, film industry, gaming and architectural modeling. Global illumination algorithms such as ray-tracing, radiosity and photon mapping suffer from speed because of complex computational operations and hardware limitations. In this paper, we present a parallelization technique for the photon mapping algorithm to reduce the rendering time of the images by distributing the job over computing units. Our parallelization method is based on a grid computing system and we obtained promising performance improvements for future studies.

1. Giriş

Işık özelliklerinin simüle edilmesi, bilgisayar grafiğinin en aktif konularından birini oluşturmaktadır. Yıllar boyunca değişik materyal özellik ve koşullarında ışık hareketlerinin simüle edilmesi amacıyla birçok teknik ortaya konmuştur. Bu tekniklerden en önemlileri, ışın izleme [1] ve ısı transferi [2] yöntemleridir. Ortaya konan bu yöntemler özellikle belirli yüzey ve materyal koşullarında iyi sonuçlar vermektedir. Örnek olarak, ışın doğrulama yöntemi parlak ve yansımali yüzeyler için başarılı iken, emici özelliğe sahip yüzeylerde başarılı değildir. Diğer taraftan ısı tranfer yöntemi tam tersi bir karakteristik göstererek emici yüzeylerde başarılı olmasına rağmen parlak yüzey özelliğine sahip yüzeyleri render etmede sorun yaşamaktadır. Bunun yanında ısı tranfer yöntemi kostik gibi özel efektleri simüle edememektedir. Işın doğrulama [3]

ve ısı transferi [4] yöntemlerindeki sorunların aşılmasına yönelik birçok yöntem ortaya konmasına rağmen gerçek anlamda tüm yüzeyler için çalışan ortak bir çözüm yöntemi ortaya konamamıştır.

Belirtilen kısıtlamaların çözülmesi amacıyla foton haritalama yöntemi ortaya konmuştur [5]. Foton haritalama yöntemi iki aşamalı bir yöntem olmakla birlikte hem parlak hem de emici materyal özelliğine sahip yüzeylerde başarılı sonuçlar vermektedir. Yöntem ayrıca kostik gibi özel efektlere de destek vermektedir.

Evrensel aydınlatma algoritmalarının geneline bakıldığında, foton haritalama algoritması ışığı simüle edebilme yeteneği açısından en başarılı seçenek gibi gözüke de oldukça fazla hesapsal güç gerektirmesi en belirgin dezavantajı olarak sayılabilir. Bu nedenle oldukça karmaşık sahnelerin oluşturulması günleri alabilmektedir. Bu hesapsal karmaşıklık aşmak için birçok yöntem ortaya konmuştur [6,7]. [7]'de sunulan teknikte foton haritalama algoritması grafik işlemci birimi üzerinde gerçekleştirilmiştir. Donanım bağımlılığı yöntemin en büyük dezavantajıdır.

Foton haritalama algoritmasını hızlandırmanın yöntemlerinden diğer biri, işlemcilerin (CPU) paraleleştirilmesidir. CPU tabanlı paraleleştirilme çözümü, düşük maliyetli bilgisayarların bir ağ yardımıyla birleştirilmesi sonucunda çok yüksek işlem güçlerine ulaşılabilmesini temel almaktadır. Foton haritalama yöntemi, ışın doğrulama algoritmasını temel aldığından dolayı paraleleştirmeye uygun bir yapıya olanaklı kılmaktadır [8].

Bu makalede foton haritalama algoritmasının paraleleştirilmesine yönelik grid hesaplama tabanlı bir yöntem ortaya konulmuştur. Grid mimarisi paralel hesaplama için esnek ve kolay genişletilebilir bir altyapı sağlamaktadır.

Makalenin İkinci bölümünde, varolan evrensel aydınlatma algoritmaları incelenmiştir. Üçüncü bölümde, foton haritalama algoritmasının tek işlemci ve paralel işlemciler üzerindeki gerçekleştirmeleri ele alınmıştır. Dördüncü bölümde, gerçekleştirilen yönteme ait deneysel sonuçlar yer almaktadır. Beşinci bölümde ileriye yönelik hedefler ile sonuç bölümleri yer almaktadır.

2. Evrensel Aydınlatma

Evrensel aydınlatma kavramı ışık davranışlarının simülasyonunda anahtar rol oynamaktadır. Evrensel aydınlatma lokal aydınlatmadakine ek olarak çevreden gelen

yansıma ve kırılmaları da dikkate alır. Çevreden gelen bu etkilerin de dikkate alınması ortaya yapay olmayan ve gerçeğe yakın bir görüntünün çıkmasına olanak sağlar. En önemli evrensel aydınlatma algoritmalarından bazıları, ışın doğrulama, ısı transferi ve foton haritalamadır.

2.1. Işın Doğrulama

Işın doğrulama tekniği, evrensel aydınlatma algoritmaları arasındaki en eski yöntemlerden biridir. Temel olarak ışın obje etkileşimlerinin arasında gerçekleşen hesaplamalara dayanır. İnsanın görme mantığı ve ışınların yayılma prensibinin basitleştirilmiş hali model olarak alınır. 3 boyutlu ortamda sahneyi algılayan (kamera) objeden sahneye doğru ışınlar gönderilir. Gönderilen ışınlar belirli seviyelerde, objelere çarparak yansıma ve kırılma gerçekleştirir. Bunun sonucunda, yüzey üzerindeki renk yoğunluk değeri hesaplanır [9].

2.2. Isı Transferi

Isı transfer yöntemi objeler arasındaki ısı transfer ilkelerinin bilgisayar grafiğine uyarlanmış şeklidir. 3 boyutlu sahne parçalara ayrılarak her parçanın diğer parçalar üzerine olan ısı transfer etkisi hesaplanır ve bu da belirlenen parça üzerindeki renk değerinin hesaplanmasını sağlar. Bu yöntem oldukça fazla hesap gücü gerektirmesinden dolayı çevrimdışı olarak gerçekleştirilir. Bu durum ısı transfer yönteminin gerçek zamanlı uygulamalarda kullanımını verimsizleştirmektedir.

2.3. Foton Haritalama

Foton haritalama yöntemi en önemli evrensel aydınlatma algoritmalarından biridir [5,10]. Bu algoritma gözün görme mantığı ve ışık kaynaklarından objeler üzerine yayılan enerjilerin hesaplanmasını temel almaktadır. Foton haritalama kendi içerisinde birçok basamak ihtiva etmektedir. Bunlardan ilki fotonların sahneye yayılmasıdır. Bu işlemde ışık kaynağından sahneye gönderilen ışınlar objelere çarpar ve sahnedeki objelerin enerjilerini toplar. Bu aşamada daha göz devreye girmemiştir. Sahneye yayılan foton yapılarına daha sonra verimli bir şekilde ulaşmak için birçok veri yapısı kullanılmaktadır. Bu veriyapılarının foton haritalama algoritması üzerindeki hızlandırma etkisi güncel bir araştırma konusudur. Veri yapılarında muhafaza edilen foton yapıları nihayetinde foton haritası olarak adlandırılan yapıyı oluşturur. Bu yapı, algoritmanın ikinci bölümünde kullanılacağından dolayı oldukça önemlidir. Algoritmanın ikinci bölümünde gözü başlangıç noktası olarak sahneye ışınlar gönderilir ve parlak yüzeyler için ışık değerleri hesaplanmış olur. Emici yüzeyler için ise birinci kısımda oluşturulmuş olan foton haritası veri yapısı kullanılmaktadır. Gözden objelere gönderilen ışınların objelere çarptıkları noktanın etrafındaki fotonları olarak enerji hesabı yapılır ve o noktadaki ışık değeri ortaya çıkar.

Foton haritalama algoritması fotoğraf gerçekliğinde sahneler üretme açısından başarılı bir yöntem olmasına rağmen oldukça fazla hesap gücüne ihtiyaç duymaktadır. Bu çalışma kapsamında algoritmanın hızlandırılması amacıyla işlemcilerin paralelleştirilmesi yöntemi seçilmiştir. Sistemin paralelleştirilmesinin en önemli nedeni, foton haritalama yönteminin çalışma prensibi olarak paralelleştirilmeye uygun olmasıdır. Algoritmanın ilk ve ikinci aşamasında kullanılan tüm yöntemler paralel işlemcilerle dağıtılabilmekte ve hız

kazancı elde edilebilmektedir. Günümüzde düşen işlemci maliyetleriyle ters orantılı olarak işlemci hızları, çok yüksek işlemci güçlerine ulaşmamızı olanaklı kılmıştır. Böylece düşük maliyetli bilgisayarlar ile çok yüksek kapasiteli bilgisayarların işlem gücüne ulaşılabilmesi de bu seçimi destekleyen unsurlardan biridir.

3. Gerçekleştirim

Foton eşleme algoritması, tek işlemci ve paralel işlemciler olmak üzere iki aşamada gerçekleştirilmiştir. Tek işlemci üzerindeki gerçekleştirim tek bir bilgisayar ve tek bir işlemci donanımı varsayılarak geliştirilmiştir. Bu varsayım problemin daha küçük parçalara ayrılmasını sağlamıştır. Paralel işlemciler üzerindeki gerçekleştirim ise, ilk aşamada tek işlemciler üzerindeki gerçekleştirimin ortaya koyduğu sonuçlar üzerinden hız kazanmayı amaçlamaktadır.

3.1. Tek İşlemci Üzerinde Gerçekleştirim

Tek işlemci üzerinde gerçekleştiriminin ilk aşaması foton eşleme algoritmasının gerçekleştirilmesidir. Foton eşleme algoritması temel olarak iki aşamadan oluşmaktadır.

İlk aşamada foton yapısı oluşturulur ve ışık kaynağından 3 boyutlu sahneye doğru fotonlar gönderilir. Sahneye çarpan foton parçacıkları o noktadaki enerji değerini saklar. Sahneye gönderilen fotonlar algoritmanın ikinci aşamasında kullanılmak üzere özel bir veri yapısı olan kd-tree veri yapısında saklanır. Bu veri yapısının kullanılmasının sebebi, gerek bellek kullanımı gerekse en yakın k adet komşu bulma probleminin çözümü açısından oldukça başarılı sonuçlar ortaya koymasındadır [5]. Algoritmanın ikinci aşamasında ışın izleme algoritması çalışmaktadır. Geleneksel ışın izleme algoritmasından en önemli farkı ışınların çarptığı noktada enerji kestirimi yapılmasıdır. Enerji kestirimi algoritmanın ilk aşamasından gelen ve kd-tree veri yapısında saklanan fotonlar vasıtasıyla gerçekleştirilmektedir. Işının çarptığı noktada k en yakın komşu algoritması uygulanmakta ve o noktadaki emicilik değeri bulunmaktadır. En yakın k nokta araması $O(\log n)$ [10] hesapsal karmaşıklık değerinde çalışmaktadır.

Foton veri yapısının en önemli özelliği, geometriden bağımsız olmasıdır. Böylece sahne içerisindeki objeler değişse bile fotonlardan gelen bilgiler doğrultusunda doğru bir kestirim yapılabilir.

Algoritmanın ilk aşamasında sahneye gönderilen fotonların bir veri yapısında saklanması gerekmektedir. Bu veri yapısının, bilgisayar hafızasını verimli bir şekilde kullanabilmesi, oluşturulduktan sonra yaprak düğümlerin hızlı bir şekilde düzenlenebilmesi ve hızlı bir şekilde en yakın k nokta araması yapabilmesi gibi özellikleri sağlaması gerekmektedir. Bu beklentiler ışığında kd-tree veri yapısı seçilmiştir.

3.2. Paralel İşlemciler Üzerinde Gerçekleştirim

Foton eşleme algoritmasının tek işlemci üzerinde gerçekleştirilmesi ortaya konulan çok sayıda optimizasyon tekniğine rağmen oldukça yavaştır. Bu yavaşlık algoritmanın çok fazla işlem yükü gerektirmesinden kaynaklanmaktadır. Bu amaçla algoritma üzerinde yapılan birçok geliştirme tekniğine ve günümüzde hızlı gelişerek artan işlemci hızlarına rağmen bu durum devam etmektedir. Bu durumun üstesinden

gelebilenin bir yolu süper bilgisayarlar kullanmaktır. Yüzlerce işlemciye sahip olan bu bilgisayarlar sahip olmak bir yana, kiralamak bile astronomik rakamlar üzerinden gerçekleşmektedir. Süper bilgisayarların ortaya koyduğu güce çok sayıdaki az işlem gücüne sahip paralel bilgisayarlar vasıtasıyla ulaşılabılır. Bu yöntemin diğer bir önemli artışı süper bilgisayarlardan farklı olarak karşılanabilir bir maliyet oranının olmasıdır. Bu etmenlerin sonucunda prosesin tamamını paralelleştirmek sonuca en kesin giden yol olarak ortaya çıkmaktadır. Foton eşleme algoritması ışın doğrulama tabanlı bir algoritma olmasından dolayı paralelleştirilmeye oldukça uygun bir yapıya sahiptir.

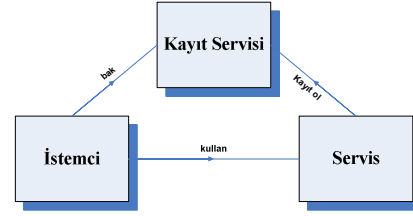
Sistemin paralelleştirilmesi en uygun çözüm gibi görünse de bu yöntemin de bazı zayıf noktaları bulunmaktadır. Üzerinde çalışılacak veri ile kullanılan işlemci sayısı arasında hassas bir ilişki bulunmaktadır. Oluşturulacak sahne gereğinden küçükse çok fazla işlemci kullanmak performansı bir noktaya kadar artıracaktır. Sınır noktasından sonra oluşturma zamanı azalmak yerine artacaktır. Bu nedenle paralel olarak işlenecek veri ile işlem elemanları (donanım) arasındaki optimum oranın dengeli bir biçimde oluşturulması gerekmektedir.

Parallelleştirme metodu JGrid [11] mimarisi üzerine geliştirilmiştir. JGrid, Jini mimarisi üzerine geliştirilmiş ve paralel uygulamaların geliştirilmesini sağlayan bir mesajlaşma altyapısıdır. JGrid teknolojisi sayesinde birçok servis objesi oluşturulabilir ve hesap ünitelerinin bu servisleri nasıl kullanacağı belirlenebilmektedir. Servis tabanlı mimari gereğince her bileşen bir servis veya bir servisin kullanıcıları olarak tanımlanmaktadır. Sistemin birbiriyle haberleşmesi amacıyla üç adet ana yapı bulunmaktadır; servis, kayıt servisi ve istemci (Hesap Ünitesi). Sistemin genel görünüşü Şekil 1'de verilmektedir.

Servis yapısı, ana işin yapıldığı kısmı tanımlamaktadır. Sistemimizde ana iş görselleştirmeyi yapan sistemdir. Kayıt servisi hesap ünitelerinin sistemde kayıtlı olan servisleri görebilmesini sağlar. Servisler kayıt ünitelerine kayıt olurlar ve sistem başlatıldığında hesap üniteleri kayıt ünitesine bakarak hangi servislerin aktif olduğunu belirler. Servislere kayıt olan hesap ünitelerinin o andan sonraki görevi, serviste belirlenen işleri gerçekleştirmektir. İstemciler, çok sayıda bulunabilen yapılardır ve paralel sistemlerdeki işlemci gücünü oluşturan ana yapıyı oluştururlar.

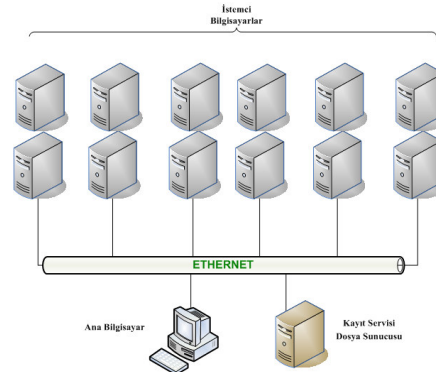
Sistemin ağ topolojisi Şekil 2'de görüldüğü gibi bir ana bilgisayar, kayıt servisi ve dosya sunucusu için bir bilgisayar ve eklenebildiği kadar hesap ünitesinden oluşmaktadır. Sistemin yönetimi ve iş dağılımı ana bilgisayar vasıtasıyla gerçekleştirilmektedir. Sahneye ait geometrik bilgilerin bulunduğu dosya ve kaplama dosyaları dosya sunucusunda tutulmaktadır. Sistem başlatıldığı andan itibaren herhangi bir zamanda hesap üniteleri dinamik olarak sisteme bağlanabilir.

Hesap ünitelerinde bulunmayan model ve kaplama dosyaları, dosya sunucusu vasıtasıyla bilgisayarlara yayılabilir. Kayıt servisi ise hesap ünitelerinin hangi işi yapacağını belirler. Kayıt servisi ve dosya sunucusunun aynı bilgisayarda çalışmaktadır. Kayıt servisine mimari olarak başka servisler de eklenebilir ve hesap üniteleri bu servislerle bağlanabilir fakat sistemin genel amacı çerçevesinde sadece bir adet servis, servis komponenti tarafından kayıt servisine kayıt olmaktadır.



Şekil 1: Servis tabanlı mimari

Parallelleştirme işlemi sırasındaki diğer bir önemli nokta ise iş dağılımıdır. İş dağılımının gerçekleştirilebilmesi amacıyla sistem en küçük iş parçacığı olan "parça"(bucket) kavramı ile ifade edilmektedir. Oluşturulacak geometri ana bilgisayar tarafından eşit aralıkta parçalara bölünmektedir. Bu parçaların boyutu kullanıcı tarafından belirlenebilmektedir. Bu parça sayısının belirlenmesi sırasında bilgisayardaki hafıza gereksinimleri, sahnenin karmaşıklığı ve çözünürlük gibi birçok parametre dikkate alınıp ortaya bir değer konulması gerekmektedir. Aksi takdirde elimizdeki hafıza, sahne ve çözünürlüğe uygun olmayan parça büyüklükleri sahnenin oluşturma süresinin uzamasına yol açabilir.



Şekil 2: Servis tabanlı mimari için ağ yapısı

Bizim sistemimizde parça büyüklüğü için jenerik bir değer olan 32x32 pixel değeri seçilmiştir. Böylece sahne 32x32 pixel değerinde iş parçacıklarına bölünüp hesap ünitelerine gönderilmektedir. Örnek bir sahne parçalama yapısı Şekil 3'de verilmektedir. Hesap üniteleri aldıkları bu parçaları sahne oluşturma servisi ile işleyip sonucu ana bilgisayara tekrar göndermektedirler. Ana bilgisayar bu gelen parçaları toplayıp tek bir imge oluşturmakla yükümlüdür. Ana bilgisayar kendisine gelen bilgileri toplayıp tek bir imge olarak ekrana gönderdiğinde işlem tamamlanmıştır.

Ana bilgisayar ile hesap üniteleri arasındaki bağlantı TCP/IP socket arayüzü vasıtasıyla gerçekleştirilmektedir. Dosya sunucusu olarak HTTP sunucusu seçilmiştir. Sahne üzerinde çalışan işlemci başlatıldığı zaman o an aktif olan hesap ünitesi adedince işlemci devreye girer ve mevzu bahis sahne dosyasının işlenebilmesi için çalışır. Sistemde bir hata oluşması durumunda tüm durumların bir log dosyası yardımı ile kaydı alınır ve sorunun olduğu kısım belirlenir. Sistem sahneyi oluşturduktan sonra bir sonraki işi bekler. Ana konsoldan bir sonraki iş yüklenir ve sistem başlatılır. Hesap üniteleri tekrar işleme başladığında yukarıda belirtilen işlem döngüsü başlar.

Sistemdeki diğer bir önemli nokta da mesajlaşma mimarisidir. Güvenilir bir mesajlaşma mimarisi olmadan sistem komponentlerinin birbirleri ile haberleşmesi imkansızdır. Burada mesaj olarak tabir edilen yapı içerisinde küçük komutlar ve bazen veriler için kullanılan yapılar bulunmaktadır. Bu mesajlar TCP/IP arayüzü vasıtasıyla bilgisayarlar arasında iletilir ve sistemin gerekli işi yapması sağlanır. Sistemde iki tür mesaj bulunmaktadır: Başlangıç mesajları ve sistem mesajları.

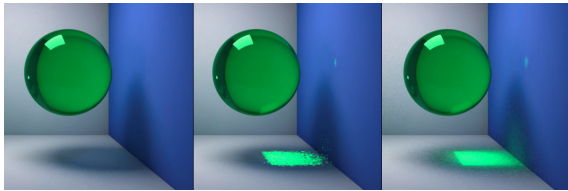
Başlangıç mesajları, sistemin başlatılması, servisler arası bağlantının sağlanması ve sistemin tam olarak çalışmaya hazır hale getirilmesini sağlayan mesajlar topluluğudur. Bu mesajlar sayesinde servisler kayıt servislerine bağlanabilmekte, hesap üniteleri sistemde kayıtlı olan servisleri listeleyp kayıt olabilmektedir.

Sistem mesajları ise sahne oluşturulması işlemini gerçekleştiren ara işlemi yapan mesajlardır. Bu mesajları ana bilgisayardan hesap ünitelerine gönderilen mesajlar ve hesap ünitelerinden ana bilgisayar gönderilen mesajlar olmak üzere iki grupta toplayabiliriz.. Ana bilgisayardan mesaj gönderilmeden önce sahne ilk olarak parçalara ayrılır ve her bir hesap ünitesine gidecek olan iş parçacığı belirlenmiş olur. Belirlenen iş parçacığı istemci bilgisayara gönderilir. Gönderilme sırası işlemcilerin doluluk oranına göre belirlenir. Sistemdeki en az dolu olan bilgisayara ilk önce iş atanır. Eğer eşit işlemci yükleri eşitse, iş ağdaki en yakın bilgisayara gönderilir.

4. Deneysel Sonuçlar

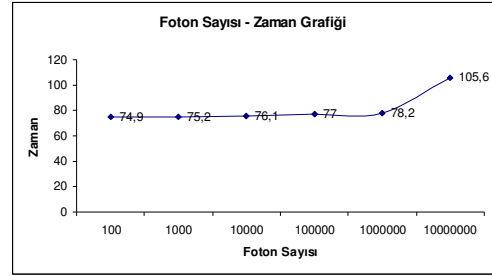
Sistem, Core 2 duo 2.33 GHz CPU ve GeForce 8600 GT grafik kartı donanımı ile, Microsoft Windows XP işletim sistemi üzerinde test edilmiştir. Sistemin paralelleştirilmesi sırasında yukarıda belirtilen konfigürasyona sahip elimizde mevcut bulunan 6 adet bilgisayar kullanılmıştır. Sistem verimliliğinin test edilmesi amacıyla 2 adet tek işlemci testi, 2 adet paralel işlemciler testi gerçekleştirilmiştir.

İlk deney tek işlemci üzerinde gerçekleştirilerek 3 boyutlu sahneye gönderilen foton sayısının görsel kaliteye olan etkisi test edilmiştir. İlk aşamada cam küre (Şekil 4) üzerine 100.000 foton gönderilmiş ve kostik etkisinin oluşmadığı gözlenmiştir. Ortaya çıkan sonuç soldaki imajda gözlenebilmektedir. İkinci aşamada sisteme 1.000.000 ve üçüncü aşamada 10.000.000 foton gönderilmiştir. Ortaya çıkan sonuçlardaki kostik etkisi ortadaki ve en sağdaki imajlarda açıkça gözlemlenebilmektedir. Bu etki, görselliğin gerçeğe yakın olmasında önemli bir yere sahiptir.



Şekil 4: Foton sayısının etkisi. Soldaki imaj 100.000 foton, ortadaki imaj 1.000.000 foton ve en sağdaki imajda 10.000.000 foton kullanılmıştır.

İkinci deneyde tek işlemci üzerinde artırılan foton sayısının sahnenin oluşturulma zamanına olan etkisi test edilmiştir. Bu deney için 1024x768 çözünürlükte kostik material özelliklerine sahip "Cornell Box" kullanılmıştır. Şekilde görüldüğü üzere kostik etkisi oluşana kadar render süresinde çok fazla bir artış gerçekleşmiyor. Kostik üretimi 1.000.000 fotonun sonra başlamakta ve bu noktadan sonra sahne oluşturma süresinde belirgin bir artış gözlenmektedir. Bu durumun temel sebebi artan hesapsal karmaşıklıkta. Diğer taraftan kostik üretiminden sonra görsel kalite artmaktadır.



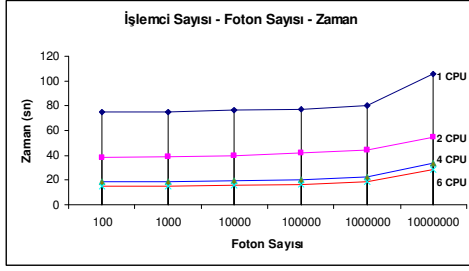
Şekil 5: Artan foton sayısı ile sahnenin oluşturulma zamanı arasındaki ilişki.

Paralel sistemin test edilmesi için oluşturulan ilk deney için 1024x768 çözünürlükteki "Cornell Box" içerisinde, bir adet cam küre ve bir adet yansımali material özelliğine sahip küre modeli kullanılmıştır. Cam küre kostik performansını gözlemlemek amacıyla kullanılmıştır. Ortaya konan sonuçlar Şekil 6'da ve oluşturulan bir örnek sahne Şekil 7'de görülmektedir.

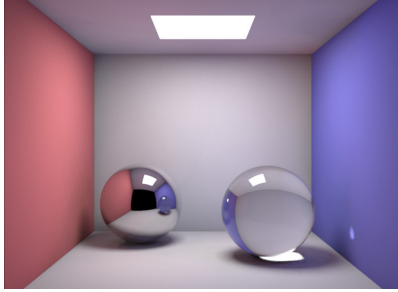
Bu deneyde ilk olarak tek CPU üzerinde foton sayısı artırılmış ve sahne oluşturma zamanı 79 sn olarak belirlenmiştir. Kostik üretimi başladığında tek işlemci üzerindeki performans azalmaktadır. Bunun nedeni, kostik üretiminin tüm sisteme getirdiği hesapsal yüküdür. İkinci aşamada iki işlemci kullanılarak aynı testler tekrarlanmıştır. Bu deneyde ortalama sahne oluşturma süresi 42 sn olarak belirlenmiştir. Bir önceki test ile son sonucu karşılaştırdığımızda sahne oluşturma süresinde meydana gelen azalma dikkat çekmektedir. Sistemdeki işlemci sayısının artışıyla ters orantılı olarak çalışma süresi azalmıştır. Aynı deney 4 işlemci üzerinde denendiğinde sonuç 20 sn olarak gözlemlenmiştir. Son olarak 6 işlemci kullanıldığında render süresi 19 sn olarak belirlenmiştir. Bu sefer sahne oluşturma süresinde beklenen düşüş gerçekleşmemiştir. Bunun nedeni paralel işlemciler arasında gerçekleşen haberleşme işlemlerinin ağ hızına bağlı olarak sahne oluşturma işleminden çok daha fazla zaman almasıdır.

Paralel işlemcilerle gerçekleştirilen ikinci deneyde 1024x768 çözünürlükte, içerisinde cam materyal özelliklerine sahip ejderha modeli bulunan "Cornell Box" kullanılmıştır. Ejderha modeli Stanford Üniversitesi grafik laboratuvarından alınmıştır.

Bu deneyde bir önceki deneydekine benzer basamaklar uygulanmıştır. Fakat burada karmaşıklığı artıran foton sayısı değil, kullanılan model içerisindeki çokgen sayısıdır. Şekil 8'de farklı sayıda CPU kombinasyonları için karşılaştırma tablosu verilmiştir.

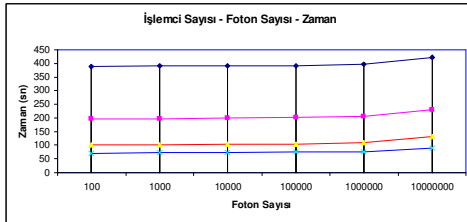


Şekil 6: Artan foton ve işlemci sayısı ile sahne oluşturma zamanı arasındaki ilişki.



Şekil 7: 6 işlemci ile oluşturulmuş bir test sahnesi

Tek işlemci kullanımında ortalama işleme süresi 400 saniyedir. Sahneye 1.000.000 foton ekledikten sonra kostik'e bağımlı olarak karmaşıklık artmıştır. Aynı test 2 işlemci için tekrarlandığında işleme süresi 200 saniye, 4 işlemci için bu süre 100 saniye ve 6 işlemci için 60 saniyedir. 6 işlemci ile gerçekleştirilen testte bir önceki teste göre verim elde edilmiştir. Bu verimin nedeni karmaşıklığı yüksek bir model kullanılmasından kaynaklanmaktadır. Şekil 9, 6 işlemci ile oluşturulmuş edilmiş bir modeldir.



Şekil 8: Foton sayısı ile sahne oluşturma zamanları arasındaki ilişki.

5. Sonuçlar

Bu çalışmada evrensel aydınlatma algoritmalarından biri olan foton haritalama yöntemi için tasarlanmış bir paralelleştirme tekniği ortaya konulmuştur. Geliştirilen sistem, tek işlemci üzerinde ışık davranışlarını simüle edebildiği gibi, paralel işlemciler üzerinde de aynı işlemleri gerçekleştirebilmektedir. Karmaşıklığı yüksek sahneler için doğrusal hız kazancı elde edilmiştir. Geliştirilen sistem kostik gibi etkilerin yanında karmaşıklığı çok yüksek olan sahneleri de verimli bir şekilde render edebilmektedir.



Şekil 9: Cam materyal özelliklerine sahip ejderha modelinin 6 işlemci ile oluşturulmuş hali.

Gerçekleştirilen araştırmanın iki koldan devam etmesi planlanmaktadır. İlk olarak dinamik yük dağıtım şeması sistemle entegre edilerek, paralel işlemciler üzerinde daha verimli yük dağıtımlarının yapılması hedeflenmektedir. Dinamik yük dağıtım işlemi ana bilgisayardan işlem birimlerine gönderilen ve büyüklüğü sabit olan iş parçacıklarının büyüklüklerinin dinamik bir şekilde dağıtılmasının sağlanması hedeflenmektedir. Araştırmanın diğer kanadında ise sualtı, ateş sahneleri gibi farklı materyal özelliklerine sahip sahnelerin fiziksel olarak doğru ve aynı zamanda hızlı bir şekilde render edilmesinin sağlanması hedeflenmektedir.

6. Kaynaklar

- [1] WALD I., SLUSALLEK P.: *State-of-the-Art in Interactive Ray Tracing*. EUROGRAPHICS 2001, pages 21–42, 2001.
- [2] GIBSON S, HUBBOLD R. J., A Perceptually-Driven Parallel Algorithm for Efficient Radiosity Simulation, IEEE Transactions, 2000.
- [3] DUTRE P, JENSEN H. W., ARVO J., *State of the art in Monte Carlo global illumination*, SIGGRAPH '04, 2004
- [4] BENTHIN C., WALD I., SLUSALLEK P.: *A Scalable Approach to Interactive Global Illumination*. Proceedings of Eurographics, 2003.
- [5] JAROSZ W., JENSEN H. W., DONNER C., *Advanced Global Illumination Using Photon mapping*, Submitted for ACM SIGGRAPH '08, 2008.
- [6] NIJASURE M., PATTANAIK S., GOEL V.: *Realtime global illumination on GPU*. Submitted for publication, 2003.
- [7] PURCELL T. J., DONNER C., CAMMARANO M., JENSEN H. W., HANRAHAN P. *Photon mapping on programmable graphics hardware*. Proceedings of the conference on graphics hardware, 2003.
- [8] SINGH S., FALOUTSOS P., *SIMD Packet Techniques for Photon Mapping*, RT '07. IEEE Symposium, 2007.
- [9] CHRISTENSEN P.H, FONG J., LAUR D. M., BATALI D.. *Ray tracing for the movie 'Cars'*. In Proceedings of the IEEE Symposium on Interactive Ray Tracing, 2006.
- [10] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Natick, MA, 2001.
- [11] JGrid: A Jini-based Universal Service Grid, <http://www.irt.vein.hu/jgrid>.