

Mobil GPU Tabanlı Önemine Göre Örnekleme

Mobile GPU-Based Importance Sampling

Özkan Anıl Töral
Bilgisayar Mühendisliği Bölümü
Yaşar Üniversitesi
ozkantoral@gmail.com

Serkan Ergun, Murat Kurt
Uluslararası Bilgisayar Enstitüsü
Ege Üniversitesi
{serkan.ergun, murat.kurt}@ege.edu.tr

Aydın Öztürk
Bilgisayar Mühendisliği Bölümü
İzmir Üniversitesi
aydin.ozturk@izmir.edu.tr



Şekil 1. Mobil cihazda; farklı modellerin, farklı çevresel ışıklar altında, 8 örnek kullanılarak elde edilen görüntüleri. (a)'da Dragon modeli Grace Katedrali'nde [1] [Saniyedeki kare sayısı (*Frames per second, FPS*) = 2], (b)'de Armadillo modeli Okaliptus Korusu'nda [FPS = 2], (c)'de Buddha modeli yüksek çözünürlüklü Grace Katedrali'nde [FPS = 3] ve (d)'de Athena modeli Pisa avlusunda görüntülenmektedir [FPS = 4].

Özetçe —Bu bildiriye, mobil cihazlar üzerinde çevresel ışıklandırılmaya dayalı interaktif görüntüleme uygulaması geliştirilmiş ve filtrelenmiş önemine göre örneklemeye yönteminin, düşük güce sahip bir mobil grafik işlemci ünitesi (*Graphics Processing Unit, GPU*) üzerinde de interaktif olarak uygulanabileceği gösterilmiştir. Mobil cihazların kısıtlı güç ve ekran boyutu göz önünde bulundurularak, kullanıcının önceden tanımlanmış farklı üç boyutlu modelleri, sahneleri ve doku kaplamalarını seçebilecekleri bir arayüz tasarlanmıştır. Çift-yönlü yansıma dağılım fonksiyonlarının (*Bidirectional Reflectance Distribution Function, BRDF*) parametreleri değiştirilerek istenilen malzemenin interaktif olarak görüntülenebileceği bir ortam sunulmuştur.

Anahtar Kelimeler—*Bilgisayar Grafikleri, Gerçek Zamanlı Görüntüleme, Mobil Grafikler*

Abstract—In this paper, we developed an interactive global illumination application for mobile devices and we show that interactive filtered importance sampling is possible even with a low-power mobile graphics processing unit (GPU). Taking the limited power and the screen size of mobile devices into account, we designed a user interface that allows selecting different predefined objects, environments and textures. With our implementation, desired materials can be visualized interactively by changing the parameters of Bidirectional Reflectance Distribution Function (BRDF).

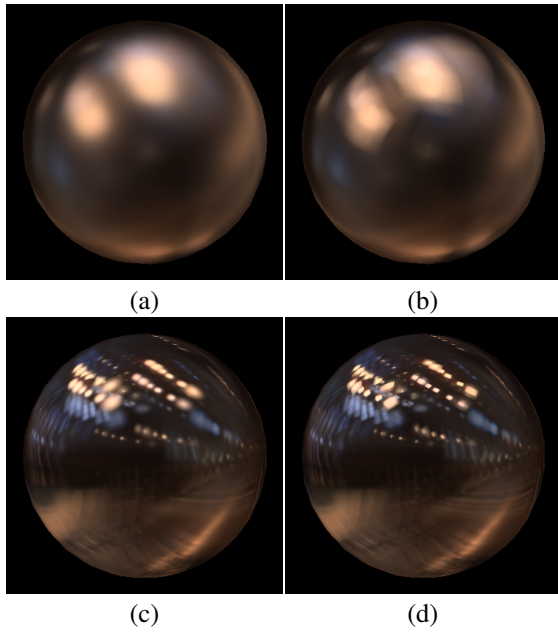
Keywords—*Computer Graphics, Real-time Rendering, Mobile Graphics*

I. GİRİŞ

Malzemelerin farklı çevresel ışıklandırma ortamlarında gerçek zamanlı olarak yansımalarının görüntülenmesi, bilgisayar grafiklerinin en yoğun çalışma alanlarından biridir. Bu konunun mobil cihazlara aktarılması ise düşük işlemci gücü ve ısınma sorunları nedeniyle zorlu bir problemdir. Bu çalışmada, mobil cihazlar üzerinde çevresel ışıklandırma ortamında interaktif görüntü oluşturulabilmesine olanak sağlayan bir yöntem öne sürülmüştür.

Mobil cihazda kullanmış olduğumuz gerçek zamanlı filtrelenmiş önemine göre örneklemeye (*Real-time Shading with Filtered Importance Sampling*) [2] yöntemi, parlak malzemelerin Monte Carlo yöntemi [3] kullanılarak görüntü tabanlı ışıklandırılmasını sağlamaktadır. Bu yöntem, minimal düzeyde ön işlem gereksinimi sağladığından ve tek bir GPU piksel programı (*Fragment Shader*) ile uygulanabildiğinden, interaktif olarak malzeme ve ışıklandırma değişikliklerinin mobil cihaza uygulanmasına olanak sağlamıştır. Yöntemin düşük örnek sayılarında da kaliteli sonuç vermesi için, Krivánek ve Colbert'in kullandığı mipmap yöntemi yerine Gaussian filtreleme [4] yöntemi kullanılmıştır. Gaussian filtresi kullanılarak elde edilen sonuçların, Şekil 2(a) ve Şekil 2(b)'deki gibi mipmap yöntemine göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Bu bildiri; Bölüm 2'de konuyla ilgili çalışmalar, Bölüm 3'de kullanılan yöntem ile ilgili teorik bilgiler, Bölüm 4'de yöntemin mobil cihaza uygulanması ve son olarak da Bölüm 5'de elde edilen sonuçlar, yorumlar ve gelecekteki çalışmalar anlatılacak şekilde hazırlanmıştır.



Şekil 2. Gaussian filtreleme yöntemi (a) ve mipmap yöntemi (b) kullanılarak 8 örnek sayısı ile elde edilen görüntülerin kıyaslanması. Yöntemlerin aşırı anizotropik malzemelerdeki $\alpha_x = 0.001$ ve $\alpha_y = 0.2$ için Gaussian filtresi (c) ve mipmap yöntemi (d) kullanılarak elde edilen hatalı görüntüleri.

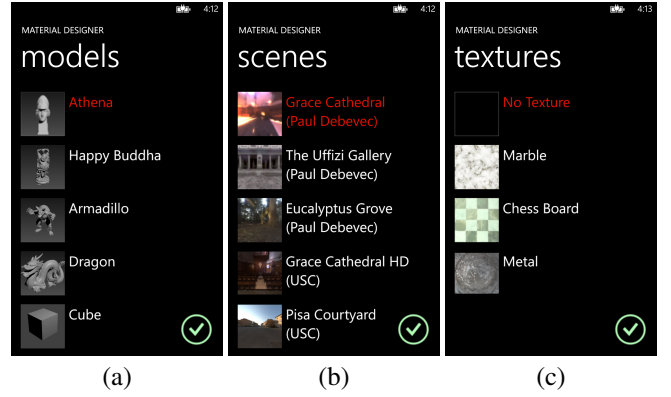
II. KONUyla İLGİLİ ÇALIŞMALAR

A. Gerçek Zamanlı Çevresel Işıklandırma

Kautz et al.'in önerdiği ön filtrelenmiş çevresel ışıklara birleşik bir yaklaşım [5] yönteminde ön filtreleme işleminden dolayı BRDF ve ışıklandırma gerçek zamanlı olarak değiştirilememektedir. Bu sorun Ramamoorthi ve Hanrahan'ın önerdiği frekans uzayında çevresel ışıklandırma [6] yöntemiyle çözüme ulaştırılsa da, parlak yüzeylerdeki yüklü ön işlem gereksiniminden dolayı mobil cihazlar için uygun görülmemiştir.

Krivánek ve Colbert'in önerdiği gerçek zamanlı filtrelenmiş önemine göre örnekleme [2] yöntemi, minimal ön işlem gereksinimi ve izotropik malzemelerin görsel olarak kabul edilebilir görüntülerinin elde edilebilmesi nedeniyle, mobil cihazlar için uygun bulunmuştur. Ancak yöntem aşırı anizotropik malzemelerde doğru sonuç vermemektedir. Wang et al.'ın önerdiği tüm frekanslar için dinamik ve değişimli yüzeylerin yansması [7] yöntemiyle, BRDF'in küresel normal dağılımların (*Spherical Gaussians*) karışımı ile ifade edilmesi sonucunda, anizotropik malzemelerde de daha doğru sonuçlar elde edilebilmektedir. Bu yöntem, aşırı anizotropik malzemelerde BRDF'in çok daha fazla küresel normal dağılımlara ihtiyaç duymasından dolayı performans sorunu ortaya çıkarmaktadır. Bu yöntemin üzerine, Xu et al.'ın önerdiği eliptik normal dağılımlar [8] yöntemi BRDF'in çok daha az eliptik normal dağılımla ifade edilmesine olanak sağlamıştır. Bu yöntem ise izotropik malzemelerin görüntülenmesinde verimli sonuç vermemektedir.

BRDF'in örneklenmesi dışında çevresel ışığın da örneklenebilmesi çoklu önemine göre örnekleme [9] yöntemi kullanılarak görüntülerde gürültünün azaltılmasında kullanılabilir. Ergun et al.'ın önerdiği çevresel ışıkların kd-ağacı kullanılarak önemine göre örneklenmesi [10] yöntemi çevrenin de örneklenebilmesine olanak sağlamıştır. Mobil cihazda elde ettiğimiz görüntülerde gürültünün azlığı ve performans ihtiyacı sebebiyle bu yöntemler kullanılmamıştır.



Şekil 3. Kullanıcının farklı (a) modelleri, (b) sahneleri ve (c) dokuları seçmesini sağlayan özelleştirilmiş menüler.

B. Mobil Cihazlarda Çevresel Işıklandırma

Lee et al.'ın önerdiği geleceğin mobil platformlarında gerçek zamanlı ışın izleme [11] çalışmasında mobil cihazlarda ışın izleme yöntemi gerçekleştirilirken, performans yetersizliklerinden kaynaklanan sorunlar anlatılmış ve yeni bir mobil GPU mimarisi sunulmuştur. Nah et al.'ın sunduğu OpenGL ES (Gömülü sistemler için OpenGL) tabanlı mobil cihazlar için melez CPU-GPU ışın izleme [12] çalışmasında ışın izlemenin bir mobil cihaza nasıl uygulanabileceği gösterilmiştir. Martin ve Wash'ın önerdiği mobil cihazlarda gelişen dinamik ışıklandırma [13] çalışmasında ise sahnenin günün zamanına göre dinamik ışıklandırılması mobil cihaza uygulanmıştır.

III. YÖNTEM

Her bir piksel için kameraya gelen tüm çevresel ışıkların hesaplanması, çevresel ışıklandırmanın esas amacıdır:

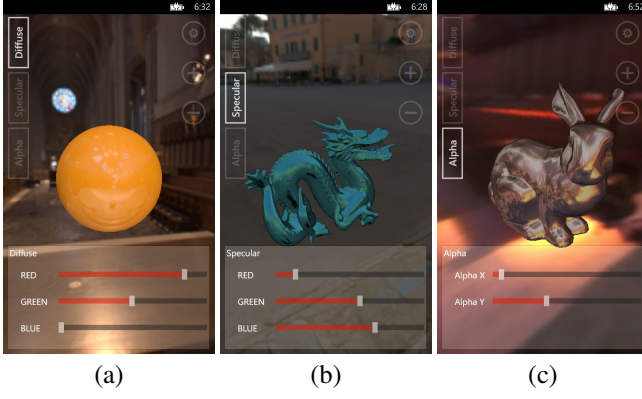
$$L_o(\omega_o) = \int_{\Omega^+} L_i(\omega_i) \rho(\omega_i, \omega_o) (\mathbf{n} \cdot \omega_i) d\omega_i. \quad (1)$$

Belirli bir alanı kapsayan çevresel ışıktan, $L_i(\omega_i)$, kamera doğrultusunda, ω_o , gelen toplam ışığın hesaplanmasında BRDF adı verilen malzeme fonksiyonu, ρ , kullanılır. Kameraya gelen toplam ışığın, $L_o(\omega_o)$, bulunması için yarı küreye, Ω^+ , düşen her ışık doğrultusunun, ω_i , yüzey normaline göre, \mathbf{n} , yansımaya olan katkısının hesaplanması gerekir. BRDF'in yüksek değerlere sahip olduğu alanlardaki çevresel ışığın yansımaya olan katkısı daha fazla olacaktır.

Işıklandırma integralinin hesaplanmasında tamamıyla her doğrultunun tek tek dikkate alınması gerekir, ama bu özellikle gerçek zamanlı yöntemler için karşılanamaz bir durumdur. Işıklandırma integralinin analitik bir çözümü de olmadığından numerik çözümlere ihtiyaç duyulmaktadır. Işıklandırma integralinin bulunması için her bir doğrultuyu hesaplamak yerine, Monte Carlo numerik integrasyonu kullanarak rastgele seçilmiş örnekler ile integral tahmin edilmeye çalışılır:

$$L_o(\omega_o) \approx \frac{2\pi}{N} \sum_{k=0}^N L_i(\omega_i^k) \rho(\omega_i^k, \omega_o) (\mathbf{n} \cdot \omega_i^k). \quad (2)$$

Eğer gerçek zamanlı olarak mobil cihazlarda sonsuz sayıda örnek üretme gibi bir imkan olsaydı, integralin kesin sonucuna ulaşılabilirdi. Bu durumun imkansız olması sebebiyle



Şekil 4. Kullanıcının Ward BRDF parametrelerinde değişiklik yapabilmesini sağlayan (a) matlık rengi, (b) parlaklık rengi ve (c) α sekmelerine ayrılmış, kullanıcı arayüz öğeleri ekran görüntüleri.

kısıtlı örnek sayısından integrali tahmin etme zorunluluğu doğmaktadır. Kısıtlı sayıdaki örnek sayısı ile integrali daha iyi tahmin etme yollarından biri de önemine göre örnekleme yöntemidir. Önemine göre örnekleme yöntemi sayesinde integrali tahmin etmek için düzgün rastgele doğrultular üretmek yerine, integralin önemli bölgelerinden daha fazla doğrultular üretmek varyansı düşürme amaçlanır. Örneklemenin bir olasılık yoğunluk fonksiyonu (*probability density function, PDF*), p , kullanılarak yapılması sayesinde istenilen doğrultular üretilebilir. Mobil cihazda da Ward modeli [14] kullanılarak BRDF'in örnekleme sağlanmıştır:

$$L_o(\omega_o) \approx \frac{1}{N} \sum_{k=0}^N \frac{L_i(\omega_i^k) \rho(\omega_i^k, \omega_o) (\mathbf{n} \cdot \omega_i^k)}{p(\omega_i^k, \omega_o)}. \quad (3)$$

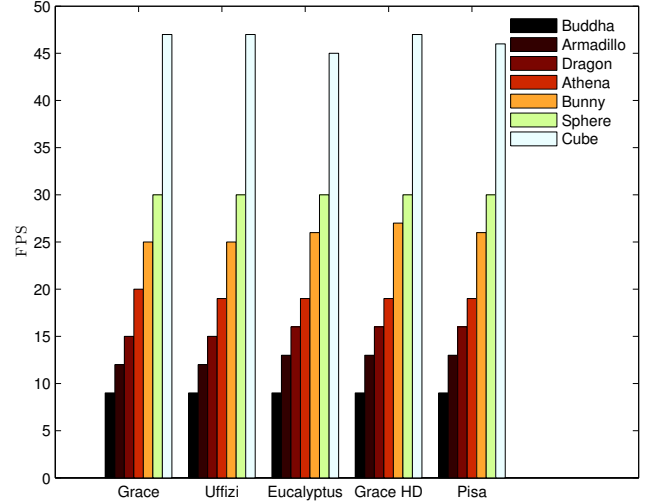
Önemine göre örnekleme yöntemi kullanılsa da, çok düşük sayıdaki örnek sayısından dolayı gürültüler oluşmaktadır. Bu sorun da örneklerin olasılıklarına göre filtrelenmiş çevresel ışıkların kullanılmasıyla, gürültünün azaltılması amaçlanmıştır.

IV. UYGULAMA

Yöntemin uygulanacağı platform olarak Windows Phone 8 (WP8) uygun görülmüştür. WP8 ile birlikte gelen C++ ve DirectX desteği sayesinde, hem yerel kod kullanılarak yazılan kütüphanelerin kullanımına hem de GPU programlarının (*shaders*) isteğe göre programlanabilmesine olanak sağlanmıştır.

WP8 platformu; geliştiricilerin, genişletilmiş uygulama biçimlendirme dili (*Extensible Application Markup Language, XAML*) ve DirectX teknolojilerinin birlikte kullanılarak uygulama geliştirilebilmesini mümkün kılar [15]. Bu sayede geliştiriciler XAML ile oluşturdukları arayüzlerini, DirectX3D grafik uygulamaları ile birleştirebilmektedirler. Her ne kadar bu etkileşim performansta azalmalara yol açılabilir de, kullanıcı arayüzünün hızlı ve kolay geliştirilebilmesini sağlar.

DirectX, Microsoft platformlarında grafik uygulamalarının da dahil olduğu çoklu ortam uygulamalarının geliştirilmesine olanak sağlayan bir uygulama programlama arayüzü (*Application programming interface, API*) koleksiyonudur. DirectX3D ise bu koleksiyonun bir parçasıdır. 2 ve 3 boyutlu grafik



Şekil 5. Modellerin farklı sahnelere göre 8 örnek kullanarak elde edilen görüntülerdeki FPS karşılaştırmaları.

uygulamalarının geliştirilmesi için kullanılır. Bu mobil uygulamada da grafik işlemleri ve GPU programlarının yazılması için DirectX3D API kullanılmıştır.

Uygulama geliştirilirken, cihazda doku ve çevresel ışık verilerinin kullanılması için DDS (*DirectDraw Surface*) dosya türü kullanılmıştır. Bu dosya türünün cihazın hafızasına kaydedilip kullanılmasını kolaylaştırması için DirectXTK (*DirectX Tool Kit*) [16] uygulamaya entegre edilmiştir.

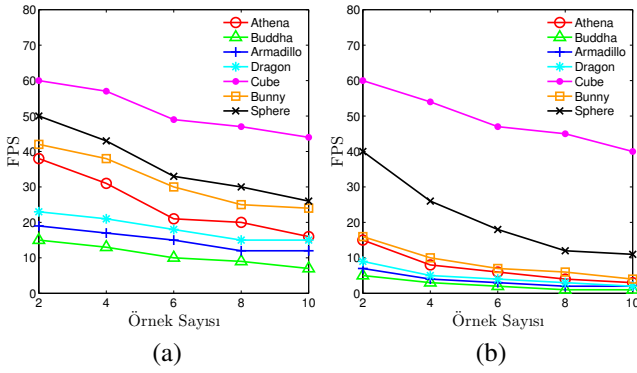
Kullanıcının isteğine göre model, sahne ve doku seçimlerini yapmasını kolaylaştıracak bir arayüz oluşturulmuştur. Daha kolay seçim yapılmasını sağlamak amacıyla, Şekil 3'te de görüldüğü gibi, XAML kullanılarak liste öğelerinin önizlemelerini de bulunduracak şekilde yeniden yapılandırılmıştır.

BRDF modeli olarak; Ward BRDF modeli [14] kullanılmıştır. Ward modeli parlaklık (*Specular*) rengi, matlık (*Diffuse*) rengi ve ışığın her iki eksen için de yansımaya alanını belirleyen α_x ve α_y parametrelerinden oluşmaktadır. Mobil cihazların ekran boyutları da göz önünde bulundurularak, kullanıcının daha rahat parametre değişikliği yapabilmesi için parametreler sekmelere ayrılmıştır. Bu sayede sadece değiştirilmek istenilen matlık ve parlaklık parametrelerinin kırmızı, yeşil ve mavi değerleri ya da α_x ve α_y parametrelerinin değerleri ekranda gösterilerek, Şekil 4'teki gibi daha geniş bir seçim olanağı sağlanmıştır.

Kullanıcının ekrana dokunarak hareket ettirmesiyle, kameranın gerçek hayata daha yakın bir etkileşim ile istenilen konuma getirilmesi amaçlanmıştır. Kameranın modele olan uzaklığı ise artı, +, eksi, - tuşlarıyla ayarlanmaktadır.

V. SONUÇLAR VE YORUMLAR

Çevresel ışıklandırmaya dayalı interaktif görüntüleme uygulaması, WP8 işletim sistemine sahip Nokia Lumia 820 üzerinde geliştirilip test edilmiştir. Şekil 5'de görüleceği gibi modellerin farklı sahnelerdeki performanslarına bakıldığında, kullanılan sahne dokusuna bağlı özel bir işlem yapılmadığından, FPS değerlerinin çevrelere göre pek bir değişiklik göstermediği görülmektedir. Ancak örnek sayılarına bağlı olacak şekilde aşırı parlak malzemelerde ($\alpha_x = 0.001$ ve $\alpha_y = 0.001$)



Şekil 6. Modellerin, farklı örnek sayıları kullanılarak (a) $\alpha_x = 0.001$ ve $\alpha_y = 0.001$ için ve (b) $\alpha_x = 0.2$ ve $\alpha_y = 0.2$ için, elde edilen FPS değerleri.

FPS değerleri yüksek çıkarken, mat malzemelerde ($\alpha_x = 0.2$ ve $\alpha_y = 0.2$) Şekil 6'da görüldüğü gibi düşük FPS değerleri elde edilmiştir. Mobil cihazda elde edilen görüntüler incelendiğinde, 8 örnek sayısının görsel olarak kabul edilebilir görüntüler oluşturduğu tespit edilmiştir. α değerlerindeki değişme miktarına göre ise FPS değerlerinin Şekil 7'deki gibi olduğu gözlemlenmiştir. Sıfıra yakın, düşük α değerlerinde çevresel ışıklardan alınan örneklerin birbirine çok yakın olması, bir sonraki örneğin önbellekte bulunma olasılığını arttırmaktadır. Bu sayede dokudan, çok daha hızlı okuma yapılarak yüksek FPS değerleri elde edilebilmektedir. n sayıdaki mipmap seviyesi için hafıza gereksinimi $\frac{2^{2n+2}-1}{3}$ piksel olurken, kullandığımız Gaussian filtreleme yönteminde ise aynı sahne dokusu için $(n+1) \times 2^{2n}$ piksel olmaktadır. Gaussian filtresi daha fazla hafıza gereksinimine ihtiyaç duymasına rağmen, düşük örnek sayısı ile yüksek kaliteli görüntüler elde edilmesine olanak sağlamıştır.

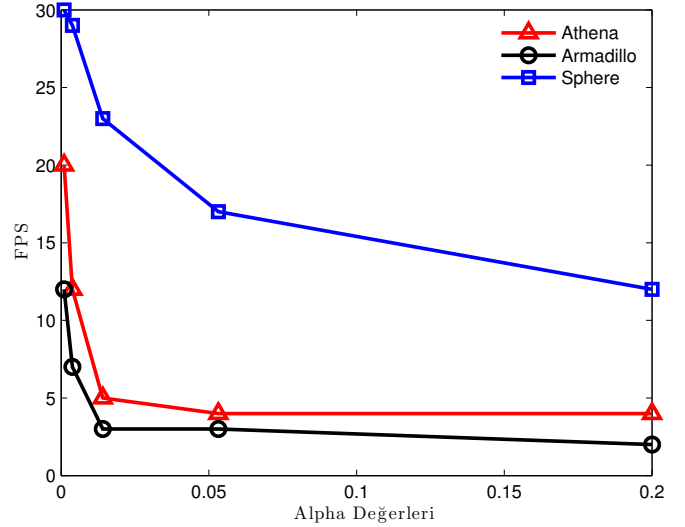
Kullanılan yöntemin aşırı anizotropik malzemelerde Şekil 2(c) ve Şekil 2(d)'deki gibi hatalı sonuç vermesinden dolayı, gelecekteki çalışmalarda bu sorun üzerinde durulması düşünülmektedir.

VI. TEŞEKKÜR

Bu bildiriadaki çalışmalar TÜBİTAK tarafından 111E208 nolu proje kapsamında desteklenmiştir. Bu bildiriadaki çalışmalar Microsoft Türkiye kurumu tarafından da desteklenmiştir.

KAYNAKLAR

- [1] P. Debevec, "Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 189–198. [Online]. Available: <http://doi.acm.org/10.1145/280814.280864>
- [2] J. Křivánek and M. Colbert, "Real-time shading with filtered importance sampling," *Computer Graphics Forum*, vol. 27, no. 4, pp. 1147–1154, 2008, eurographics Symposium on Rendering, EGSR '08. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2008.01252.x>
- [3] M. Pharr and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.
- [4] R. Gonzalez, R. Woods, and S. Eddins, *Digital Image Processing Using MATLAB*. Pearson Education, 2004. [Online]. Available: <http://books.google.com.tr/books?id=5YEMp6HjvnQC>



Şekil 7. Farklı α değerlerine göre, Grace Katedrali sahnesinde, 8 örnek kullanılarak elde edilen FPS değerleri.

- [5] J. Kautz, P.-P. Vázquez, W. Heidrich, and H.-P. Seidel, "Unified approach to prefiltered environment maps," in *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*. London, UK, UK: Springer-Verlag, 2000, pp. 185–196. [Online]. Available: <http://dl.acm.org/citation.cfm?id=647652.732274>
- [6] R. Ramamoorthi and P. Hanrahan, "Frequency space environment map rendering," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 517–526, Jul. 2002. [Online]. Available: <http://doi.acm.org/10.1145/566654.566661>
- [7] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, "All-frequency rendering of dynamic, spatially-varying reflectance," in *ACM SIGGRAPH Asia 2009 Papers*, ser. SIGGRAPH Asia '09. New York, NY, USA: ACM, 2009, pp. 133:1–133:10. [Online]. Available: <http://doi.acm.org/10.1145/1661412.1618479>
- [8] K. Xu, W.-L. Sun, Z. Dong, D.-Y. Zhao, R.-D. Wu, and S.-M. Hu, "Anisotropic spherical gaussians," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 209:1–209:11, Nov. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2508363.2508386>
- [9] E. Veach, "Robust monte carlo methods for light transport simulation," Ph.D. dissertation, Stanford, CA, USA, 1998, aAI9837162.
- [10] S. Ergun, M. Kurt, and A. Öztürk, "Real-time kd-tree based importance sampling of environment maps," in *Proceedings of the 28th Spring Conference on Computer Graphics*, ser. SCCG '12. New York, NY, USA: ACM, 2012, pp. 77–84. [Online]. Available: <http://doi.acm.org/10.1145/2448531.2448541>
- [11] W.-J. Lee, Y. Shin, J. Lee, S. Lee, S. Ryu, and J. Kim, "Real-time ray tracing on future mobile computing platform," in *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*, ser. SA '13. New York, NY, USA: ACM, 2013, pp. 56:1–56:5. [Online]. Available: <http://doi.acm.org/10.1145/2543651.2543670>
- [12] J.-H. Nah, Y.-S. Kang, K.-J. Lee, S.-J. Lee, T.-D. Han, and S.-B. Yang, "Mobirt: An implementation of opengl es-based cpu-gpu hybrid ray tracer for mobile devices," in *ACM SIGGRAPH ASIA 2010 Sketches*, ser. SA '10. New York, NY, USA: ACM, 2010, pp. 50:1–50:2. [Online]. Available: <http://doi.acm.org/10.1145/1899950.1900000>
- [13] S. Martin and M. Wash, "Advancing dynamic lighting on mobile," in *ACM SIGGRAPH 2012 Mobile*, ser. SIGGRAPH '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:1. [Online]. Available: <http://doi.acm.org/10.1145/2341910.2341915>
- [14] G. J. Ward, "Measuring and modeling anisotropic reflection," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 265–272, Jul. 1992. [Online]. Available: <http://doi.acm.org/10.1145/142920.134078>
- [15] Microsoft, "Directx and xaml interop," <http://msdn.microsoft.com/en-us/library/windows/apps/hh825871.aspx>, January 2014.
- [16] Microsoft, "Directx tool kit," <https://directxtk.codeplex.com/>, January 2014.