



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**SAVUNMA SANAYİ UYGULAMALARINA
YÖNELİK ÇOK KATMANLI ARTIRILMIŞ
GERÇEKLİK GÖRÜNTÜ ÜRETİCİNİN
TASARLANMASI VE GELİŞTİRİLMESİ**

Resul AYDOĞAN

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ağustos-2019
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Resul AYDOĞAN tarafından hazırlanan "SAVUNMA SANAYİ UYGULAMALARINA YÖNELİK ÇOK KATMANLI ARTIRILMIŞ GERÇEKLİK GÖRÜNTÜ ÜRETİCİNİN TASARLANMASI VE GELİŞTİRİLMESİ" adlı tez çalışması 28/08/2019 tarihinde aşağıdaki jüri tarafından oy birliği / ~~oy çokluğu~~ ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Doç.Dr. Bayram AKDEMİR

Danışman

Prof. Dr. Seral ÖZŞEN

Üye

Dr. Öğretim Üyesi Güzin ÖZMEN

İmza

.....
.....
.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Hakan KARABÖRK
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Resul AYDOĞAN

Tarih:

ÖZET

YÜKSEK LİSANS

SAVUNMA SANAYİ UYGULAMALARINA YÖNELİK ÇOK KATMANLI ARTIRILMIŞ GERÇEKLİK GÖRÜNTÜ ÜRETİCİNİN TASARLANMASI VE GELİŞTİRİLMESİ

Resul AYDOĞAN

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı**

Danışman: Prof. Dr. Seral ÖZŞEN

2019, 68 Sayfa

**Jüri
Prof. Dr. Seral ÖZŞEN
Doç. Dr. Bayram AKDEMİR
Dr. Öğr. Üyesi Güzin ÖZMEN**

Bu çalışmada, artırılmış gerçeklik teknolojisi uygulamalarının, akıllı gözlükler, cep telefonları ve giyilebilir mobil cihazlar üzerinde gerçekleştirilmesi için yeni bir yaklaşım olan çok katmanlı mimariye uygun görüntü üretici uygulaması geliştirilmiştir. Artırılmış gerçeklik uygulamaları, bilgisayar grafikleri teknolojisi üzerine inşa edilerek gerçek dünya ile sanal dünyayı birleştirmekte ve gerçek zamanlı olarak kullanıcıya zenginleştirilmiş görsel ile etkileşimli içerik sunmaktadır. Bu uygulama, üzerinde çalıştığı donanım platformlarının işlemci, grafik işlemci ve batarya gibi mevcut kaynaklarını üst seviyede tüketmektedir. Yüksek detay yapısına sahip modellemelerin ve yoğun grafik işlemcisi gerektiren uygulamaların gerçekleştirilmesinde mobil cihazlar yetersiz kalmaktadır. Bu problemi çözmeye yönelik olarak geliştirilen görüntü üretici mimarisi, yüksek kaynak tüketimi gerektiren grafik işlemleri, kaynak açısından kısıtlı olmayan platformlara taşımaya imkân tanımıştır. Sonuç olarak giyilebilir ve diğer mobil cihazlar üzerinde, her türlü içerik seviyesi için, düşük ve sabit düzeyde kaynak tüketimi sağlanmış ve uygulama verimliliği yükseltilmiştir. Tez sonucu, özellikle savunma sanayinde saha uygulamaları sırasında karşılaşılan problemlere çözüm getirme açısından incelenmiştir. Bu çalışmada, C++ dili ile geliştirme yapılmıştır. Pencere yönetiminde Qt Kütüphaneleri, OpenGL altyapısında OpenSceneGraph kütüphanesi, görüntü işleme ve matris operasyonlarında OpenCV kütüphanesi kullanılmıştır. ARM ve INTEL işlemci mimarisine sahip mobil cihazlar ile INTEL masaüstü bilgisayar platformlarında testler yapılmıştır.

Anahtar Kelimeler: Artırılmış Gerçeklik, Savunma Sanayi, Qt, OpenGL, Openscenegraph, OpenCV

ABSTRACT

MS THESIS

DESIGN AND DEVELOPMENT OF A MULTI-LAYER AUGMENTED REALITY IMAGE GENERATOR FOR DEFENSE INDUSTRY APPLICATIONS

Resul AYDOGAN

**Konya Technical University
Institute of Graduate Studies
Department of Electrical and Electronics Engineering**

Advisor: Prof. Dr. Seral ÖZŞEN

2019, 68 Pages

**Jury
Prof. Dr. Seral ÖZŞEN
Assoc. Prof. Dr. Bayram AKDEMİR
Assist. Prof. Dr. Güzin ÖZMEN**

In this thesis, a multi-layered architecture image generator application has been developed by introducing a new approach to realizing applications while with augmented reality technology on mobile devices such as wearable smart glasses and mobile phones. Augmented reality applications has been build on computer graphics technology . These technology is combining the real world with the virtual world, provided the user with enhanced visual / interactive content in real time. These application consumes the available resources of the hardware platforms on which it operates, such as processors, graphics processors and batteries is used at quickly high level. Mobile devices are inadequate in the realization of high-detail modeling and applications requiring intensive graphics processing. The image generator architecture, which was developed to solve this problem, has enabled the transfer of graphic processes that require high resource consumption to platforms that have no resource constraints. As a result of, low and constant resource consumption for wearable and other mobile devices for all levels of content is achieved and application efficiency is improved. Theses results, particularly in terms of bringing solutions to problems were studied during field applications in the defense industry. At the end of thesis has been analyzed to find solutions for the problems in the field application, development has been made with C ++ language. Qt Libraries are used in window management, OpenSceneGraph library in OpenGL infrastructure and OpenCV library in image processing and matrix operations. The tests have been made in the mobile devices that use ARM and INTEL processor architecture and the computers that use intel desktop platforms.

Keywords: Augmented Reality, Defence Industry, Qt, OpenGL, Openscenegraph, OpenCV

ÖNSÖZ

Bu çalışmada, maddi ve manevi destekleriyle her koşulda yanımda olan kıymetli AİLEM'e, kaliteli bir mühendis olma yolunda verdikleri eğitim ve destekler ile kıymetli hocam Prof. Dr. Mehmet Ali EBEOĞLU'na, duydukları güven ve sundukları fırsatlar ile Türk Savunma Sanayi'ne hizmet etmemi sağlayan İŞVEREN ve YÖNETİCİLERİME, tecrübeleri ve destekleri ile her koşulda yanımda olan dostlarım Dr. Muhammet Üsame ÖZİÇ'e, Ali BELLİDİNÇ'e ve Cem Murat TURGUT'a, bilgisi, tecrübesi ve en önemlisi sabrıyla desteklerini esirgemeyen değerli danışman hocam Prof. Dr. Seral ÖZŞEN'e teşekkürü bir borç bilirim.

Resul AYDOĞAN
KONYA-2019

İÇİNDEKİLER

| | |
|---|-----------|
| ÖZET | iv |
| ABSTRACT..... | v |
| ÖNSÖZ | vi |
| İÇİNDEKİLER | vii |
| SİMGELER VE KISALTMALAR | ix |
| 1. GİRİŞ | 1 |
| 1.1. Tezin Amacı ve Önemi | 1 |
| 1.2. Tezin Organizasyonu | 2 |
| 2. KAYNAK ARAŞTIRMASI | 4 |
| 3. ARTIRILMIŞ GERÇEKLİK | 7 |
| 3.1. Uygulama Alanları..... | 8 |
| 3.2. Yazılım Bileşenleri | 14 |
| 3.3. Giyelebilir ve Mobil Donanım Platformları | 17 |
| 4. MATERYAL VE YÖNTEM..... | 24 |
| 4.1. Görüntü Üretici Tasarımı ve Gerçekleştirilmesi..... | 24 |
| 4.1.1. Ana Katman Mimarisi | 27 |
| 4.1.1.1. Pencere Yöneticisi | 28 |
| 4.1.1.2. Grafik İşleme Altlığı..... | 30 |
| 4.1.1.3. Kullanıcı Arayüzü Katmanı | 31 |
| 4.1.2. AG Katmanları Mimarisi | 34 |
| 4.1.2.1. Çoklu Katman Mimarisi | 36 |
| 4.1.2.2. Saydam Katman Yöneticisi | 38 |
| 4.2. Çoklu Platform Desteği ve Yazılım Geliştirme Süreçleri | 42 |
| 5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA..... | 47 |
| 5.1. Kaynak Tüketiminin Analizi | 47 |
| 5.1.1. Temel Kaynak Tüketiminin Analiz Verileri | 49 |
| 5.1.2. AG Katmanlarının Farklı Çözünürlükler İçin Performans Analiz Verileri .. | 50 |
| 5.2. Görüntü Üretiminin Analizi | 50 |
| 5.2.1. AG Uygulaması ve GÜ Uygulamasının Karşılaştırma Verileri | 51 |
| 5.2.2. Görünüş Oranı Uyumluluğunun Analizi..... | 53 |
| 5.2.3. Ekran Yenileme Performansının Analizi | 54 |
| 6. SONUÇLAR VE ÖNERİLER | 56 |

| | |
|------------------------|-----------|
| 6.1 Sonular | 56 |
| 6.2 neriler | 61 |
| KAYNAKLAR | 64 |
| ZGEMİŐ | 67 |



SİMGELER VE KISALTMALAR

Simgeler

| | | |
|--------|---|---------------------------------|
| $Fovh$ | : | Yatay görüş açısı |
| N | : | Kamera sensörü yatay kenar boyu |
| fx | : | Yatay Odak Uzaklığı |
| fy | : | Dikey Odak Uzaklığı |

Kısaltmalar

| | | |
|--------|---|--|
| 2B | : | 2 Boyutlu |
| 3B | : | 3 Boyutlu |
| AG | : | Artırılmış Gerçeklik |
| API | : | <i>Application Programming Interface</i> |
| AR | : | <i>Augmented Reality</i> |
| CPU | : | <i>Central Processing Unit</i> |
| DARPA | : | <i>Defense Advanced Research Projects Agency</i> |
| FPS | : | <i>Frame Per Second</i> |
| GMOS | : | <i>Game Mean Opinion Score</i> |
| GPS | : | <i>Global Positioning System</i> |
| GPU | : | <i>Graphical Processing Unit</i> |
| GUI | : | <i>Graphical User Interface</i> |
| GÜ | : | Görüntü Üretici |
| IDE | : | <i>Integrated Development Environment</i> |
| IMU | : | <i>Inertial Measurement Unit</i> |
| IoT | : | <i>Internet of Things</i> |
| IPP | : | <i>Intel Performance Primitives</i> |
| Kb | : | Kilobayt |
| LTE | : | <i>Long Term Evolution</i> |
| Mb | : | Megabayt |
| MinGW | : | <i>Minimalist GNU for Windows</i> |
| NDK | : | <i>Native Development Kit</i> |
| OpenCL | : | <i>Open Computing Library</i> |
| OpenCV | : | <i>Open Computer Vision</i> |
| OpenGL | : | <i>Open Graphics Library</i> |
| OS | : | <i>Operating System</i> |
| OSG | : | OpenSceneGraph kütüphanesi |
| PC | : | <i>Personal Computer</i> |
| RAHAT | : | Rafta Hazır Ticari |
| SDK | : | <i>Software Development Kit</i> |
| QML | : | <i>Qt Modeling Language</i> |



1. GİRİŞ

Artırılmış Gerçeklik (AG), bilgisayar grafikleri ile gerçek dünya ve sanal dünyaları birleştirerek son kullanıcıya dönük görsel etkileşimli içerikler sunabilen bir teknolojidir (Palmarini ve ark., 2018). Eğitim, sağlık, endüstri, eğlence, perakende ve askeri alanlarda son dönemlerde dikkat çeken bir teknoloji olarak ortaya çıkmaktadır. AG'nin zenginleştirilmiş görsel etkileşimler ile kullanıcıya herhangi bir alanda üstünlük sağlaması son dönemlerde gelişmiş ve gelişmekte olan ülkelerde araştırılan bir konu haline gelmiştir. AG teknolojilerinin özellikle askeri alandaki uygulamaları gelişmiş ülkelerde sıklıkla çalışılmakta olan güncel bir konudur. Bu uygulamalar askeri sahada durumsal farkındalık açısından ciddi anlamda teknolojik bir üstünlük sağlamaktadır (Rathnayake, 2018). Saha uygulamaları için ülkeler ciddi fonlar ayırmakta ve AG teknolojisinin diğer askeri teknolojilerle entegre olacak şekilde çalışabilmesi için ARGE faaliyetleri yapmaktadırlar (Palmarini ve ark., 2018). Türkiye de son dönemlerde AG teknolojilerinin askeri alanda kullanılabilmesi için ciddi yatırımlar yapmaktadır. AG alanındaki her uygulama, yazılım, donanım ve mimari olarak kendine has bir model ile çalışmaktadır. Her aşamada ciddi anlamda problemler ortaya çıkabilmekte ve bu problemlerin çözümü için konu ile ilgili literatüre hâkim olmak gerekmektedir. Bu tez çalışmasında özellikle askeri alanda sahada yaşanan problemler göz önüne alınarak AG teknolojisi için yeni bir mimari model geliştirilmiştir.

1.1. Tezin Amacı ve Önemi

AG, bir teknoloji uygulamasından daha öte kendi içerisinde farklı teknolojileri bir çatı altında toplayan mühendislik alanıdır. AG uygulamaları, düşen donanım maliyetleri sayesinde yaygınlaşmaya başlamıştır. Reklam, tanıtım, oyun sektörü başta olmak üzere sağlık ve savunma alanlarında farklı uygulamalar geliştirilmektedir. Bu teknoloji alanında geliştirilen uygulamalar, donanım ve işletim sistemiyle direkt ilintilidir. Donanımların, işlem yapabilme kabiliyeti, yazılım algoritmalarını ve yeteneklerini kısıtlamaktadır. Ayrıca farklı platformlarda farklı işletim sistemlerine yazılımın tekrar yazılmasını gerektirmektedir.

AG projeleri birçok alt bileşen içerdiğinden, yazılım, mimari ve donanım açısından ciddi bir deneyim ve akademik birikim gerektirmektedir. Teknolojinin saha

uygulamalarında birçok problem ortaya çıkmaktadır. Temel problemler 3 ana başlık altında incelenebilir:

- Mobil cihazların işlemci ve batarya kapasitelerinin, çalışılan uygulama karmaşıklığının artması ile birlikte yetersiz kalması,
- Çok kullanıcı, ortak çalışma içeren uygulamaların geliştirilmesinde, halihazırda kullanılabilen artırılmış gerçeklik uygulamaları geliştirme platformlarının ve yazılım araçlarının yetersiz kalması,
- Çok kullanıcı uygulamalarda, içerik yönetiminin uç donanımlarda (akıllı gözlük, telefon, tablet vs.) yapılmasının, gerçek zamanlı içerik yönetimi açısından güncelleme zorluğu.

Bu tez çalışmasında, AG uygulamalarında bahsedilen problemlerin çözümünde kullanılmak üzere bilgisayar tarafından sentetik görsellerin üretilmesi ve gerçek dünya görseli ile birleştirilmesine yönelik çalışmalar yapılmıştır. Bu amaçla yeni bir mimari model önerilerek bu mimarinin yazılım tasarımı geliştirilmiş ve donanımsal gerçekleştirilmesi yapılmıştır. Tamamı mobil cihazlar üzerinde yapılan bu işlemlerin, uzakta bulunan ve yüksek işlem kapasitesi sunan bir sisteme taşınmasını mümkün kılan bir Görüntü Üretici (GÜ) mimarisi üzerine çalışılmış, özellikle askeri uygulamalar gibi sahada kullanıma yönelik olan AG sistemlerine uygulanabilirliği araştırılıp önerilerde bulunulmuştur. Tez çalışması kapsamında, Windows işletim sistemine sahip bir masaüstü bilgisayar, Epson BT-350 model akıllı gözlük ve Samsung S6 model akıllı telefon donanımları üzerinde çalışılmıştır. Qt yazılım geliştirme ortamında, OpenSceneGraph, OpenCV ve Android NDK kütüphaneleri kullanılarak yazılım geliştirilmiştir. Ülkemizde AG altyapıları konusunda yapılan akademik çalışmaların az olduğu göz önünde bulundurulursa bu çalışma gelecek çalışmalar için bir kaynak niteliğinde olacaktır.

1.2. Tezin Organizasyonu

Tez çalışması 6 ana bölümden oluşmaktadır.

Birinci bölüm tezin amacı ve önemini, ve tezin organizasyonunu anlatan giriş kısmından oluşmaktadır.

İkinci bölümde ise kaynak araştırmasına yer verilmiştir.

Üçüncü bölümde artırılmış gerçeklik hakkında genel bilgiler verilmiş, uygulama alanları anlatılmış, mimari özellikleri ve sistem gereksinimleri aktarılmıştır.

Dördüncü bölümde, tez kapsamında geliştirilen GÜ mimarisinden ve alt bileşenlerinden bahsedilmiştir.

Beşinci bölümde, sistem gereksinimlerinin karşılanmasına yönelik yapılan sınamalar ve sınama sistemi anlatılmıştır.

Altıncı bölümde, test sonuçları değerlendirilmiş, karşılaştırmalar yapılarak sonuçlar tartışılmıştır. Tez kapsamında geliştirilmiş olan yazılım mimarisinin saha uygulamalarında yer alması için gereken GÜ dışındaki bileşenlerin gereksinimleri ile ilerleyen süreçte yapılacak benzer çalışmalara yönelik önerilere yer verilmiştir.



2. KAYNAK ARAŞTIRMASI

Shi (2011), mobil platformlar üzerinde çalışan 3B oyunlar ve 3B video tabanlı uygulamaları, *Remote Rendering* tekniğini kullanarak mobil platformu yoğun işlemci yükünden kurtaran tekniklerde karşılaşılan ağ kaynaklı gecikmelerin düşürülmesine yönelik çalışmalarda bulunmuştur. 3B oyunlar, 3B video tabanlı çoklu görüntülü ve çoklu yayınlar gibi yoğun işlem gücü ve yüksek ağ bant genişliğine ihtiyaç duyan uygulamaları çalıştırma gücüne sahip değildir. *Remote rendering* konsepti mobil platformu yoğun işlem gücünden kurtardığı için bu problemin çözümüne yönelik genel ve etkili bir yaklaşım olarak değerlendirilse de bu konseptin en önemli sorunu da görüş açısının değiştiği kullanıcı ile etkileşim içinde olan bir mobil istemcideki ağdan kaynaklı gecikme miktarıdır. Bu gecikme miktarı günümüz kablosuz ağlarında, uygulama çalışma zamanı boyunca sürekli değişebilmektedir ve kullanıcı deneyimini olumsuz etkilemektedir. Bu çalışma, kullanıcı görüş açısındaki değişimden işlenmiş ve güncellenmiş görüntüye kadar olan zamanı düşürmeyi amaçlamaktadır. Herhangi bir andaki görüş açısı v^- 'den v^+ 'ya değiştiği anda mobil istemci yeni görüntüyü direk olarak eski görüş açısındaki verileri kullanarak yeniden yüksek kalitede yaratabilirse sorun çözülmüş olacaktır. Bu çalışmada mobil istemci *3B image warping algorithm* gibi resim tabanlı *rendering* teknikleriyle herhangi bir kullanıcı etkileşiminde yeni görüş açısını yaratabilmeye yönelik çözüm geliştirilmiştir (Shi, 2011).

Perez ve ark. (2013), insan-makine arayüzlerini çalıştıran mobil platformların performans kısıtlarını aşmak amacıyla, uzaktaki bir rendering sunucusu tarafında yapılmasına imkân veren bir sisteme yönelik etkileşim ve görsel veri aktarımı teknikleri üzerine çalışmışlardır. Bir operatörün yardımıyla çalışan endüstriyel üretim makineleri genellikle kendi insan-makine arayüzüne sahiptir. Günümüz teknolojisindeki makinelerin dokunmatik ekranlarında bu insan-makine arayüzünü görüntüleyebilmek için her makine ekstra bir CPU'ya ve gelişmiş bir ekran kartına sahip olmak zorundadır. Her makinenin bu donanımlara sahip olması da göz ardı edilemeyecek bir üretim maliyeti demektir. Bu çalışmada, çözüm olarak GUI uygulamalarının *render* işlemlerinin uzaktaki paylaşılan bir *rendering* sunucusu tarafından yapılması önerilmiştir. Bu çözüm genel olarak dört adımdan oluşmaktadır. Bunlar sırayla: birçok insan-makine arayüzündeki uygulamanın isteğinin merkezi bir sunucuda işlenmesi, her makineye özel çıktılarının kaydedilerek verimli bir video formatında ağ üzerinden istemcilere yayınlanması ve kullanıcı etkileşimi verilerinin sonucunu bir sonraki video

yayının oluşturulması için yeniden merkezi sunucuya gönderilmesidir. Bu çözüm ile her istemcideki yüksek maliyetli donanımdan kaçınmak amaçlanmıştır (Perez ve ark., 2013).

Chen ve ark. (2017), uzaktan görüntü oluşturma sisteminde sunucu ve istemci arasındaki bağlantı kablosuz olduğunda etkileşim gecikmesini azaltmak ve kullanıcı deneyimini artırmak için istemci tarafı GPU hızlandırmalı sahne işleme tekniği önermişlerdir. Dağıtılmış görüntü oluşturma tekniği, istemcideki görüntüleri sunucu tarafından oluşturulan referans arka plan görüntüleri ve karşılık gelen derinlik haritaları ile çarpıtma ve interpolasyon yapabilmektedir. Video akışına dayalı bulut oyun sistemi ile karşılaştırıldığında değerlendirme sonuçları, yüksek ağ gecikmesi durumunda bile etkileşimli oyunlar için Oyun Ortalama Görüş Puanının (GMOS) 1'den 4,1'e çıktığını göstermektedir. Önerilen sistemin bir prototipi, ticari bir akıllı telefon ile gerçekleştirilmiştir. ASUS PadFone2'de, önerilen sistem 4'lü skala faktörü ile 40 fps, 8'li skala faktörü ile 60 fps'ye ulaşabilmektedir. Ayrıca, önerilen sistem, mevcut çözümle karşılaştırıldığında, yüksek ağ gecikme koşullarında Oyun Ortalama Görüş Puanını (GMOS) 1 ila 4.1 arasında geliştirmektedir. Bu çalışmada kullanılan referans çerçeve yerleştirme stratejisi genel durumlar içindir. Bulut oyun veya sanal gerçeklik sunucusu kullanarak yapılacak olan gelecekteki çalışmaları şekillendireceği belirtilmiştir (Chen ve ark., 2017).

Palmarini ve ark. (2018), Artırılmış Gerçeklik (AG) uygulamalarının bakım, tamir ve sürdürülebilirlik alanları gibi insan operasyonlarını destekleyici faaliyetlerde kullanımını 30 farklı yayının analizi yardımıyla incelemişlerdir. Günümüzde AG teknolojisinin bakım-onarım uygulama alanları başlıca; havacılık endüstrisi, mekanik bakımlar, endüstriyel tesis bakımları, tüketici teknolojisi ve nükleer endüstridir. AG uygulamalarının bu alanlardaki kullanımı ise dört ana başlık altında toplanabilir: Onarım, eğitim ve alıştırma, montaj-demontaj faaliyetleri, teşhis-tanımlama. De Crescenzo'ya göre hava yolu taşımacılığındaki bakım/onarım faaliyetlerinde insana bağlı hataları azaltmanın başlıca çözümü AG teknolojisinin kullanımınıdır. Geleneksel eğitim süreçleri hava taşıtları gibi karmaşık sistemler için yetersiz kalmaktadır. Bu süreçler, AG ile desteklenmelidir. Bu çalışmada, Fiorentino ve ark. (2014) göre, AG ile bir motorun bakım/onarım faaliyetlerini incelediği çalışmada %79'a kadar zaman tasarrufu ve %92.4'e kadar hata oranında azalmayı rapor etmiştir. Tüketici elektroniği gibi küçük cihazların konu edildiği bir çalışmada da AG uygulamaları %30'a kadar bakım/onarım masraflarını ve %40'a kadar seyahat maliyetlerini düşürmüştür. Sonuç

olarak, geleneksel telefon üzerinden desteğe dayalı yöntemler mevcut teknolojinin karmaşıklığıyla baş edememektedir ve AG uygulamalarının zaman, hata, güvenlik ve maliyet gibi konulardaki üstünlüğü, onu, gelecekte geleneksel bakım/onarım destek sistemlerinin yerine koymaktadır. Özetlemek gerekirse günümüz AG teknolojisi hala güncel endüstriyel gereksinimleri tamamen karşılayamasa da birçok alanda geleneksel yöntemlere karşı üstün durumdadır. Bakım, tamir ve sürdürülebilirlik faaliyetlerinde eğitim ve alıştırma sürelerini kısaltmak, maliyet ve insana bağlı hata oranlarını azaltmak için bu teknolojinin kullanılması şarttır ve birçok endüstriyel alan için farklı platformlarla ve metotlarla geliştirilmeye devam edilmektedir (Palmarini ve ark., 2018).

Gallagher ve ark. (2017), tehlikeli ortamlarda gerçekleştirilen askeri dalgıç operasyonlarında AG teknolojisinin kullanımı ve bu teknolojinin getireceği faydalar üzerine çalışma yapmışlardır. Bu çalışmada, (Gallagher ve ark., 2017) askeri dalgıç operasyonlarının, dünya üzerindeki tehlikeli ortamlarda yürütülmesi ve su altı görüşündeki yetersizliğin, operasyonların yönetilmesine meydana getirdiği ciddi boyutlara varan sınırlamalar incelenmiştir. Dalgıçların iletişim, yön bulma, durumsal farkındalık, sonar görüşü gibi kritik verilere, ortamdaki görüş koşulları ne olursa olsun erişebilmelerini sağlayacak bir sistem önerilmiştir. Bu sistem, yüksek çözünürlüklü sonar video, sektör sonar fotoğrafları, yön bulma yardımcısı, AG görselleri ve metin mesajlarının, operasyon sırasında, personele uygun şekilde aktarılması üzerine çalışılmıştır.

3. ARTIRILMIŞ GERÇEKLİK

Artırılmış gerçeklik; kullanıcının bulunduğu ortamda algıladığı gerçek dünya görseli üzerine, bilgisayar tarafından üretilmiş sentetik dünya görselinin entegre edilmesi ile uygulanan bir yazılım ve donanım teknolojisidir (Palmarini ve ark., 2018). Araştırma ve uygulama çalışmaları 1960'lı yıllara kadar uzanan bu teknoloji, günümüz dünyasında son kullanıcıların kolaylıkla ulaşabildiği akıllı gözlükler ve cep telefonu gibi yüksek teknoloji içeren donanım ve yazılım bileşimi ürünler sayesinde oyun, eğitim, pazarlama, sanayi uygulamaları ve askeri projeler gibi alanlarda artan bir ivme ile günlük hayatta yer almaktadır (Palmarini ve ark., 2018). Kullanıcının bulunduğu ortam ile ilgili verileri, gerçek zamanlı ve anlık olarak aktararak, içinde bulunulan ortamın daha iyi algılanmasını ve anlaşılır olmasını sağlayan, değerlendirme ve karar verme süreçlerini geliştiren ve hızlandıran bir teknolojidir (Rathnayake, 2018). Bilgisayar tarafından üretilen 3B grafik, ses, video ve görüntü verileri ile gerçek dünya görüntüsü zenginleştirilerek kullanıcının durumsal farkındalığı geliştirilmektedir. Dokunsal geribildirim (*haptic feedback*), mimik tanıma (*gesture recognition*) ve ortam algılamaya yönelik çeşitli sensörler aracılığıyla kullanıcının bulunduğu ortamdaki durumu ve etkileşimi ile ilgili geribildirimler alınarak, anlık olarak değişen gerçek zamanlı içerik üretimi yapılmasına ve bu içeriklerin bilgisayar tabanlı yazılım teknolojileri ve donanım tabanlı ekran teknolojileri aracılığıyla birleştirilmiş bir şekilde kullanıcıya sunulmasına dayanmaktadır. Şekil 3.1'de AG teknolojisine ait bir uygulama görseli yer almaktadır.

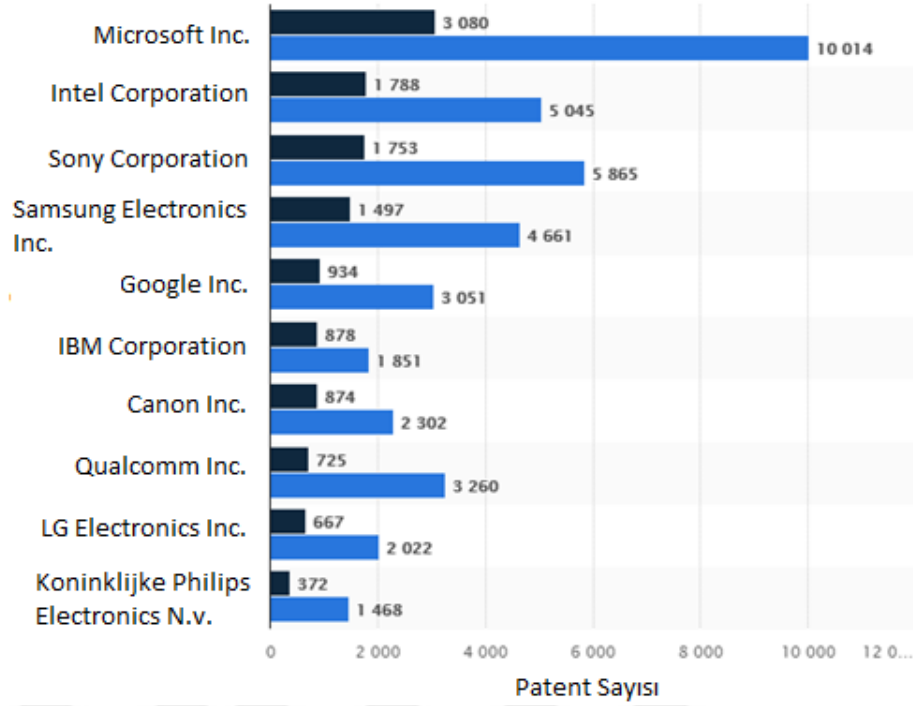


Şekil 3.1. AG teknolojisinin askeri alanda kullanımına örnek, HAVELSAN ASGER (Milscint, 2017)

Artırılmış gerçeklik uygulamaları, uzun zamandır hayatımızda yer alan cep telefonları ve tabletler gibi yüksek çözünürlük ve renk derinliğine sahip dokunmatik ekran içeren, ortam algısına yönelik kamera, GPS ve benzeri sensörler barındıran donanımlarla gerçekleştirilmektedir. Düşük güç tüketimli işlemcilerin geliştirilmesi ve üzerinde entegre devre taşıyan anakartların boyutlarının küçülmesi ve esnek devre kartlarının gelişimi ile birlikte daha kompakt ve akıllı donanımların geliştirilmesinin önünü açmıştır. Bu gelişime ek olarak batarya teknolojilerindeki gelişim sayesinde yüksek kapasiteli bataryalara benzer enerji depolarının daha küçük hacimli piller ile sağlanmasıyla, giyilebilir mobil cihazlar kavramı son kullanıcıya sunulan gerçek bir sektör haline gelmiştir (Lv ve ark., 2015).

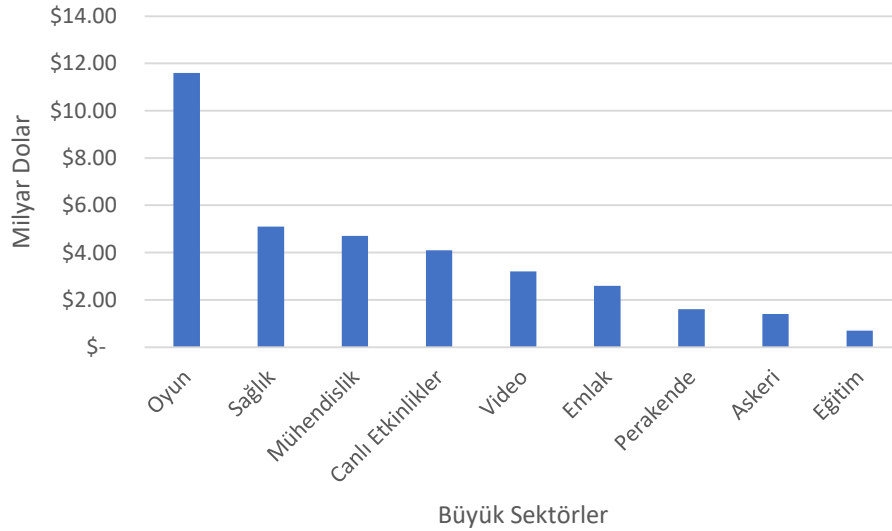
3.1. Uygulama Alanları

Artırılmış gerçeklik teknolojisi, dünya çapında teknolojiye yön veren büyük firmalar tarafından desteklenmektedir. Yazılım ürünleri geliştirmekle birlikte, hem yazılım ve donanım teknolojisinde hem de uygulama pazarında yoğun patent çalışmalarını beraberinde getirmektedir. Alphabet (Google), DAQRI, Facebook, HTC, Magic Leap, Microsoft Corporation, Osterhout Design Group, Samsung Electronics, Sony, Wikitude şirketleri artırılmış gerçeklik teknolojisine yatırım yapan, yazılım ve donanım ürünleri ile sektörde yer alan büyük firmalardır. Şekil 3.2' de şirketlerin 2019 yılına ait AG patent adetleri verilmektedir.



Şekil 3.2. Artırılmış gerçeklik teknolojisine yönelik patent adetleri (Anonim, 2019f)

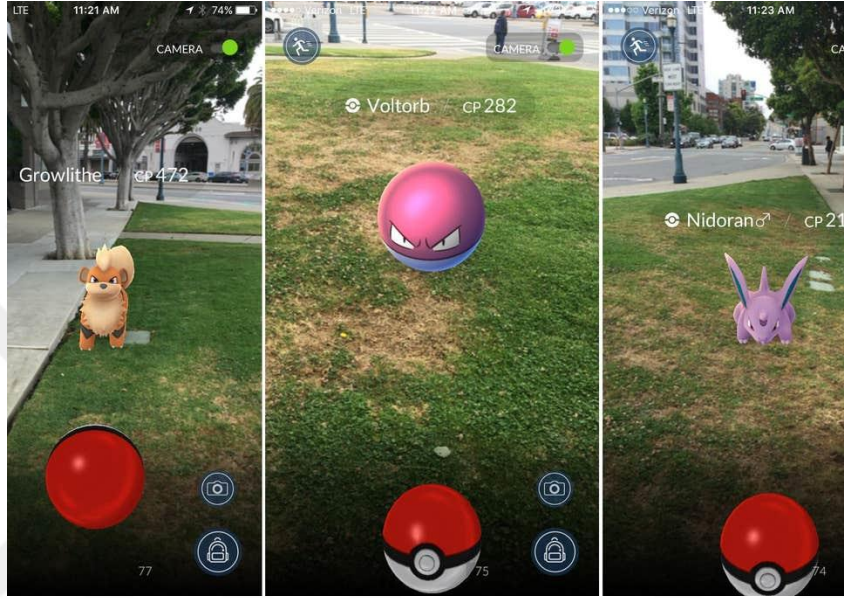
Lumus Vision tarafından Nisan 2018 tarihinde yayınlanan rapora göre AG uygulama pazarının 2025 yılı potansiyel pazar büyüklüğü Şekil 3.3’de verilmiştir.



Şekil 2.3. AG uygulama pazarının 2025 yılı potansiyel pazar gelirleri (Anonim, 2018a)

AG teknolojisi, artan bir ivme ile oyun, eğlence, sağlık, eğitim, üretim, askeri uygulamalar gibi farklı sektörlerde yaygınlaşmaktadır. Son kullanıcı nezdinde artırılmış gerçeklik uygulamasının en bilinen örneği Temmuz 2016 tarihinde yayınlanan Pokemon

GO oyunudur. Kısa sürede 2.2 milyar dolar gibi bir finansal hacim elde eden bu oyunla, sektörde ciddi bir ivmelenme ve farkındalık başlamıştır (Fogel, 2018). Bu teknolojinin bir oyun aracılığıyla tüm yaş aralığındaki insanların ilgisini çekmesinin nedeni oyunun dünya çapında bilinirliğinin yanında kullanıcılarına algısal olarak yeni bir dünyanın kapılarını açması olmuştur. Şekil 3.4’de Pokemon GO oyunundan bir ekran gösterilmektedir.



Şekil 3.4 Pokemon GO oyunu kullanıcı ekranı (Hooton, 2016)

AG uygulamaları ayrıca perakende sektöründe son kullanıcı tarafından büyük ilgi görmektedir. Donanım ve yazılım gereksinimlerini sağlayan mobil cihazların piyasada artması ile birlikte günlük hayatta kullanılan birçok faydalı uygulamanın da uygulama marketlerinde yer aldığı görülmektedir. Bunlara örnek uygulama IKEA tarafından kullanıcılara sunulmuştur. Bu uygulama ile mobilya alışverişi yapan insanların ilgili ürünleri doğrudan kendi evlerinde, satın almadan dekora dahil etmelerini sağlayarak ortam farkındalığı ve karar verme süreçlerini iyileştirmekte ve hızlandırmaktadır. Şekil 3.5’ de IKEA AG uygulamasından bir görüntü verilmektedir.



Şekil 3.5. IKEA AG uygulaması kullanıcı ekranı (Wira, 2017)

Sanayi ve endüstri uygulamalarında ise artırılmış gerçeklik uygulamaları daha kritik rol almaktadır. Kullanıcıya, bulunduğu ortamda ve gerçekleştirdiği göreve dair anlık ve gerçek zamanlı bilgi aktarımı yaparak hata yapma olasılığını en aza indirmektedir. Böylelikle görev kritik sistemlerde, ağır sanayide ve benzeri ortamlarda olası insan hatasını düşürmekte ve üretim-montaj zamanlarını en aza indirmektedir. Şekil 3.6'da Italtipresse Gauss firmasının İngiltere merkezli AVEVA firması tarafından geliştirilen AG uygulamasını bakım onarım hatlarında kullanımına dair kullanıcı ekranı yer almaktadır.



Şekil 3.6. Italtipresse Gauss firmasının bakım prosedürlerinde kullandığı AG kullanıcı ekranı görseli (Sertin, 2018)

Artırılmış gerçeklik teknolojisinin kullanıldığı kritik sektörlerden birisi de savunma sanayi uygulama alanlarıdır. 2018 yılı son çeyreğinde ABD ordusu, Microsoft

ile 100.000 adet Hololens AG gözlüğü alımı için 480 milyon dolarlık bir kontrat imzalamıştır. Bu alımın, eğitim ve saha uygulamalarında kullanılmak üzere yapıldığı açıklanmıştır (Osborn, 2018). Bu sektörde, özellikle havacılık sistemlerinin üretim, montaj, bakım prosedürlerindeki hassasiyet nedeniyle, bu hatlardaki insan hatasını en aza indirme ve zaman kullanımını açısından iş süreçlerinin yüksek verimle yönetilmesi sağlamaya imkân vermesi ile kritik önem taşımaktadır (Palmarini ve ark., 2018; Alexander ve ark., 2019). Benzer şekilde, aviyonik sistemlerin, hava aracı kullanan pilotlara geleneksel aviyonik sistemler ile sunulması yerine AG uygulamaları aracılığıyla sunulması üzerine çalışmalar yürütülmektedir (De Crescenzo ve ark., 2010). Böylelikle fiziksel sabit ekranlar yerine, pilotlara özgü düzenlenebilir ve düşük maliyetle güncellenebilir yeni nesil aviyonik sistemler kullanılacaktır. Şekil 3.7’de BAE Systems tarafından yayınlanan konsept görseli yer almaktadır.



Şekil 3.7. AG tabanlı kokpit görseli (Dormehl, 2019)

Bir diğer askeri alan uygulaması olarak, saha personelinin görev etkinliğini ve durumsal farkındalığını artırmaya yönelik geliştirilen kritik ürünler, AG teknolojisinin taşıdığı önemin daha iyi anlaşılmasını sağlamaktadır. Özellikle meskun mahal operasyonlarında durumsal farkındalık büyük önem arz etmektedir. İstihbarat verilerinin sunumu, hızlı ulaşılabilirliği, taktiksel verilerin hızlı ve uygun şekilde analizi, takım üyelerinin bilgi ve istihbarat paylaşımı, bu verilerin takımlar arası ve komutanlık seviyesinde paylaşımı gibi kritik süreçlerin verimini artırmaya yönelik uygulamalar günümüzde hızla üretilmekte ve üzerlerinde çalışılmaktadır. Şekil 3.8’de Amerika

Savunma İleri Araştırma Projeleri Ajansı (DARPA) tarafından çalışılan AG uygulamasına ait görsel yer almaktadır.



Şekil 3.8. Askeri saha personelinin AG teknolojisi temelli durumsal farkındalık uygulaması (Madrigal, 2014)

Savunma sanayi uygulamalarına bir diğer örnek, ülkemizde SimBT firması tarafından geliştirilen TAKBUL sistemidir. Aralık 2017 tarihinde Jandarma tarafından kullanılmaya başlanan akıllı gözlük üzerinde çalışan yazılım sistemidir. İçişleri Bakanlığı Kaçakçılık, İstihbarat Harekât ve Bilgi Toplama Dairesi Başkanlığı (KİHBİ) veri tabanı ile entegre edilerek, araç plakalarının, sürücü ehliyetlerinin ve yolcu kimliklerinin analiz edilmesine imkân vermektedir. Şekil 3.9'da TAKBUL sisteminin saha uygulamasından bir görüntü yer almaktadır.

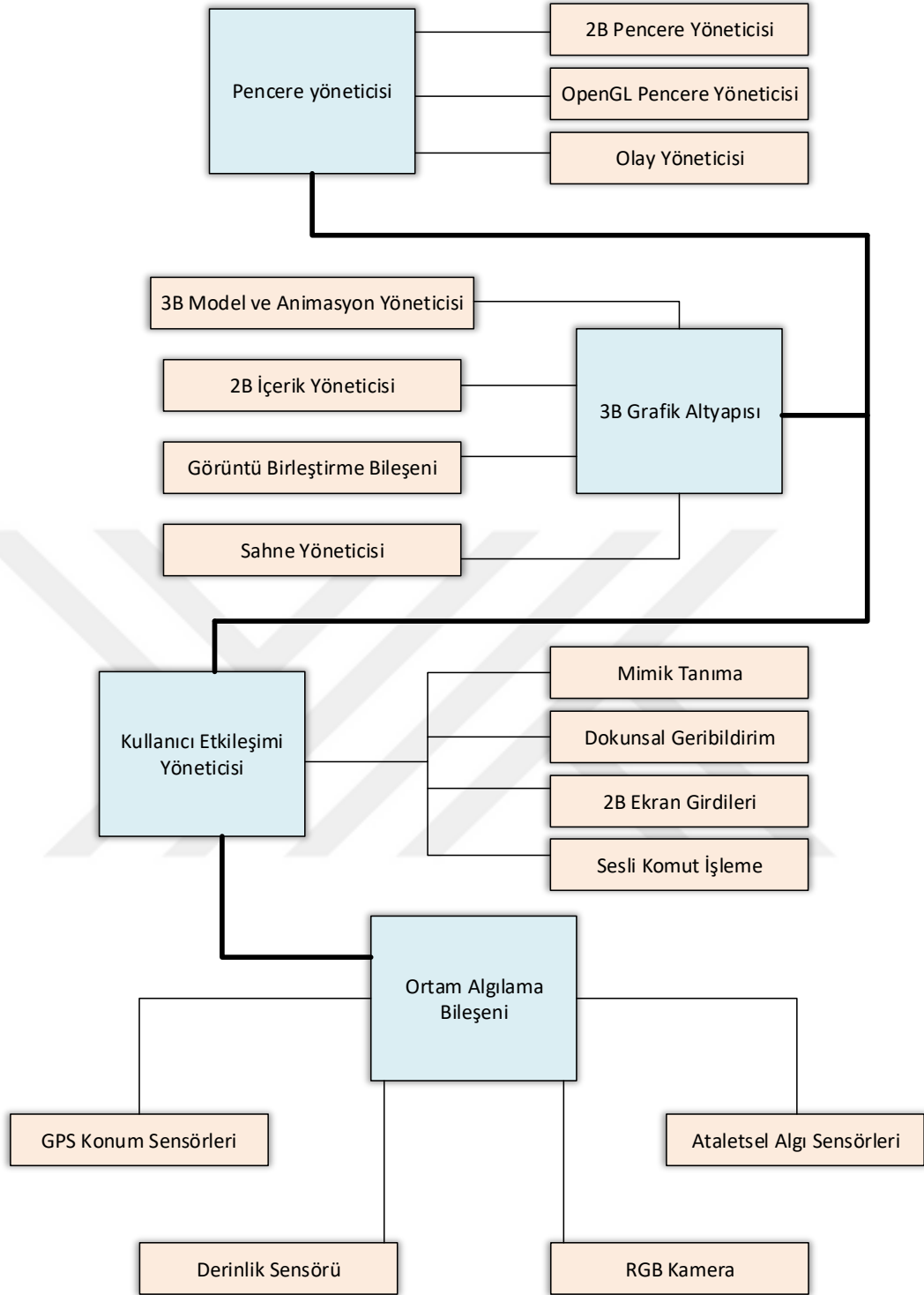


Şekil 3.9. SimBT firması tarafından geliştirilen TAKBUL sistemi uygulama sahası görseli (Anonim, 2019a)

Ülkemiz savunma sanayi de dahil olmak üzere pek çok ülkenin enstitüleri ve savunma sanayi firmaları tarafından çalışılmakta olan AG uygulama sahası, tez konusu kapsamında iyileştirilme çalışmaları yapılan yazılım mimarisi için doğrudan hedef çalışma alanında yer almaktadır.

3.2. Yazılım Bileşenleri

Artırılmış gerçeklik uygulama geliştirme platformları, birçok alt bileşenden oluşan bütünleşik bir mimariye sahiptir. Temelinde bilgisayar grafikleri ve sinyal işleme algoritmaları yer almaktadır. Tüm alt bileşenlerini kapsayacak şekilde tasarlanmış olan AG yazılımlarına ait genel mimari, Şekil 3.10'da yer almaktadır.



Şekil 3.10. Artırılmış Gerçeklik uygulamaları genel yazılım mimarisi

Giyilebilir ve diğer mobil cihazlar kendi üzerlerinde bilgisayar donanımı barındırmaktadır. Uygulama çalıştırma platformları olarak Windows 10 IoT, Android ve Gömülü Linux gibi üretici tarafından tanımlanmış farklı işletim sistemleri bulunmaktadır. Pencere yöneticisi bileşeni, işletim sistemine özgü olarak, uygulamanın cihaz ekranında işletilmesini sağlayan bileşendir. Dokunma, tıklama sürüklenme gibi

ekran üzerinde gerçekleşen kullanıcı etkileşimlerini yönetmek, 3 boyutlu bilgisayar grafikleri işlemleri sonucu üretilen 2 boyutlu görsel çıktının ekranda çizilmesini sağlamak gibi işlevleri yerine getirmektedir.

3B grafik altyapısı bileşeni, artırılmış gerçeklik uygulamalarının temelini oluşturan sentetik dünya görsellerinin hesaplanmasını ve gerçek dünya görseli üzerine yerleştirilmesini sağlamaktadır (Perez ve ark., 2013). İçerik yönetimi bileşeni ile model ve animasyonların 3B sahne ortamında, tasarlandıkları konumda ve nitelikte çizilmesini sağlamaktadır. Sahne yöneticisi bileşeni, sentetik dünyada tanımlı kullanıcı kamerasının yönetilmesini sağlayarak, ortam algılama bileşeni ile birlikte, gerçek dünya ile sentetik dünyanın ortak bir bakış açısında işlenmesine imkân sağlamaktadır. Ortam algılama bileşeni, kullanıcının içinde bulunduğu gerçek dünya şartlarının sentetik dünya içerisine taşınmasına imkân tanımaktadır. Mekânsal haritalama teknolojisi bulunan cihazlar, derinlik algılama sensörleri aracılığıyla kullanıcının içinde bulunduğu ortamın, bilgisayar grafikleri tarafında birebir tanınmasını sağlamaktadır. Böylelikle, çarpışma, gölgeleme, engelleme ve benzeri fiziksel işlemler gerçeğe yakın algılar ile kullanıcıya sunulabilmektedir. Konum ve yönelim algılama teknolojileri aracılığıyla, kullanıcının gerçek dünyadaki konumu ve bakış açısı hesaplanmakta, bu veri sentetik bilgisayar dünyasında yer alan sanal kameraların konum ve bakış açısı parametreleri olarak ayarlanmaktadır. Böylelikle gerçek dünya ile sentetik dünya görseli tam uyum ve örtüşüm ile birleştirilebilmektedir. Şekil 3.11’de mekânsal haritalama teknolojisine örnek olarak, StereoLabs dokümantasyonlarında yayınlanan bir görsel yer almaktadır.



Şekil 3.11. Mekânsal haritalama teknolojisine örnek bir görsel (StereoLabs, 2019)

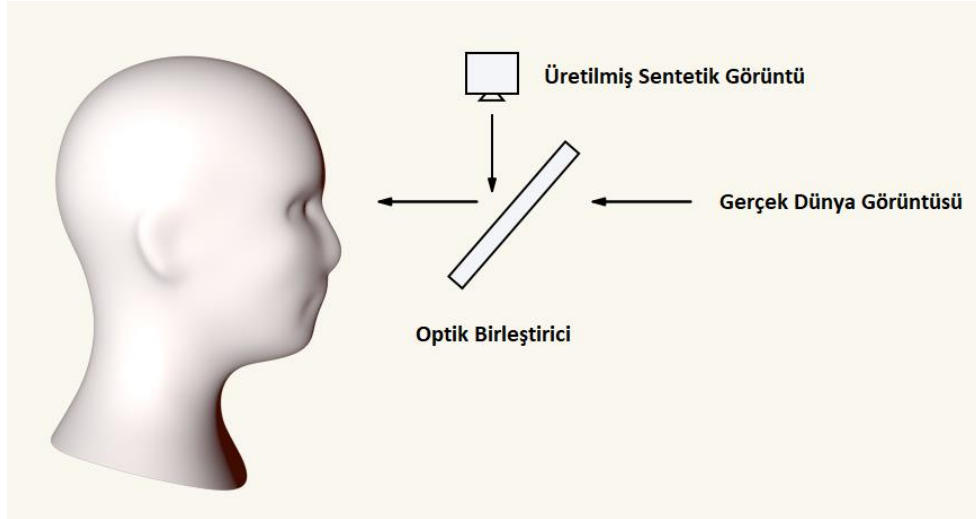
Kullanıcı etkileşimi yöneticisi, kullanıcı ile ortam arasındaki etkileşimi yönetmektedir. Mimik tanıma bileşeni ile el hareketleri aracılığıyla nesnelere tutma taşıma veya uygulama menülerinde gezme gibi işlemler yapılabilmektedir. Dokunsal geribildirim bileşenleri, sanal nesnelere temas edildiği durumlarda kullanıcının fiziksel olarak hissetmesine yönelik ısı, titreşim, basınç gibi fiziksel algılar üretmektedir. Cep telefonları veya tabletler gibi kullanıcı dostu arayüz bileşenleri bulunmayan akıllı gözlüklere yönelik olarak sesli komut alma ve işleme teknolojileri eller serbest bir şekilde AG uygulaması ile etkileşime girmeye imkân tanımaktadır. Şekil 3.12’de dokunsal geri bildirim uygulamasına örnek görsel yer almaktadır.



Şekil 3.12. Dokunsal geribildirim uygulaması görseli (Anonim, 2018b)

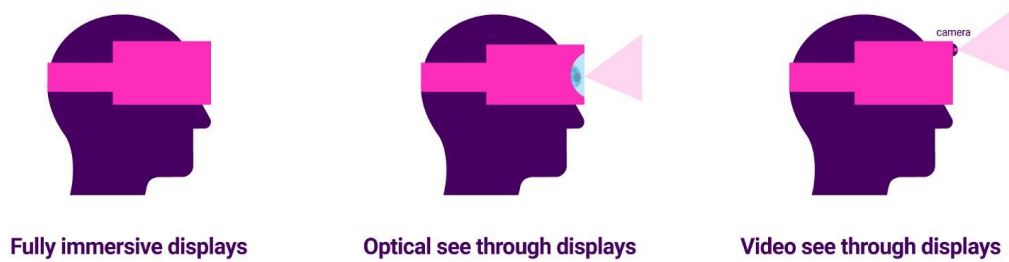
3.3. Giyilebilir ve Mobil Donanım Platformları

Artırılmış gerçeklik uygulamaları, telefon ve tabletler gibi dokunmatik ekran içeren cihazlar ile gerçekleştirilebildiği gibi, saydam ekran teknolojisine sahip akıllı gözlükler ile de gerçekleştirilmektedir. Saydam ekran teknolojisi, Şekil 3.13’de görüldüğü gibi, gerçek dünya görüntüsü ile sentetik üretilmiş bilgisayar görüntüsünü optik yöntemle birleştirerek kullanıcıya doğrudan kendi algısı ile zenginleştirilmiş görüntü sunmaktadır.



Şekil 3.13. Saydam ekran teknolojisi (Jamak, 2017)

Artırılmış ve sanal gerçeklik uygulamaları için farklı görüntüleme yaklaşımları bulunmaktadır (Suthau ve ark., 2002). Tam sürükleyici ekranlar (*fully immersive displays*) kullanıcıyı gerçek dünyadan tamamen kopartarak sanal gerçeklik dünyasına taşımaktadır. Optik birleştirici ekranlar (*optical see through displays*), kullanıcının doğrudan algıladığı gerçek dünya görüntüsünü yarı saydam ekran teknolojisi aracılığıyla sentetik üretilmiş bilgisayar grafikleri ile zenginleştirmektedir. Video görüntüleyici ekranlar (*video see through displays*) ise kullanıcının içinde bulunduğu gerçek dünyayı gözlük üzerinde yer alan kamera ile görüntülemek suretiyle, kameradan alınan görüntünün yine sentetik üretilmiş görseller ile zenginleştirilmesi esasına dayanmaktadır. Bahsi geçen görüntüleme yaklaşımları Şekil 3.14’de görselleştirilmiştir.



Şekil 3.14. AG uygulamalarında kullanılan görüntüleme teknolojileri (Kore, 2018)

Günümüzde son kullanıcıya sunulan akıllı gözlükler iki kategoriye ayrılmaktadır;

-Kendi bilgisayar ünitesine sahip olan donanımlar: Bu kategoride yer alan cihazlar, cep telefonları gibi bir mini bilgisayar mimarisine sahiptir. Geliştirilen uygulamalar dışı bağımlı olmadan doğrudan cihaz üzerine kurulabilir ve çalıştırılabilir niteliktedir.

-Yalnızca görüntüleme ünitesi olan donanımlar: Bu cihazlar ise, ekran ünitesi ve yerleşik sensör donanımları barındırmaktadır. Herhangi bir bilgisayar mimarisi ve işletim sistemi benzeri bir bileşen barındırmamaktadır. Geliştirilen uygulamayı çalıştırmak için harici bir bilgisayar ünitesine ihtiyaç duymaktadırlar. HDMI ve USB gibi veri yolları ile bilgisayar ünitesine veri aktarmakta ve veri almaktadırlar.

Tez çalışması kapsamında 3 farklı akıllı gözlük modeli incelenmiş olup avantajları ve dezavantajları açısından karşılaştırılmıştır. Şekil 3.15’de yer alan Epson BT-350 model gözlüğe ait analizler sonucu ortaya çıkan avantaj ve dezavantajlar aşağıda sıralanmıştır:

Avantajları;

- Kendi üzerinde bilgisayar mimarisine sahip
- İşlemci ünitesinin fiziksel olarak ayrılması ile ısınma sorunlarına karşı dirençli yapı.
- Kullanım kolaylığı sunan fiziksel özellikler

Dezavantajları:

- USB 2.0 veri yolu nedeniyle düşük bant genişliği
- Düşük işlemci performansı
- 23 derece sentetik ortam (AG) görüş açısı imkânı



Şekil 3.15. Epson BT-350 model akıllı gözlük(Anonim, 2019g)

Şekil 3.16’da yer alan ODG-R7 model gözlüğe ait analizler sonucu ortaya çıkan avantaj ve dezavantajlar aşağıda sıralanmıştır:

Avantajları;

- Kendi üzerinde bilgisayar mimarisine sahip
- Kompakt yapısı ile kolay kullanıma sahip

Dezavantajları:

- Manyetik USB bağlantı noktası ve USB 2.0 veri yolu sürekli kablo bağlı uygulamalarda verimsiz çalışma durumu
- Yetersiz soğuma kapasitesi
- Çalışma zamanı boyunca sıcaklık artışı nedeniyle ciddi performans düşüşleri
- 23 derece sentetik ortam (AG) görüş açısı imkânı



Şekil 3.16. ODG-R7 model akıllı gözlük (Anonim, 2019e)

Şekil 3.17’de yer alan META-2 model gözlüğe ait analizler sonucu ortaya çıkan avantaj ve dezavantajlar aşağıda sıralanmıştır:

Avantajları;

- Çalışma süresince sıcaklık artışı sorunu yaşanmaması ve sınırsız enerji bağlantısı
- Derinlik sensörü bulundurması nedeniyle farklı uygulama yöntemlerine imkân sağlaması

- HDMI veri yolu ile görüntü, USB 3.0 veri yolu ile sensör verisi aktarımı yeteneği
- 90 derece sentetik ortam (AR) görüş açısı imkânı

Dezavantajları:

- İşlemci ünitesi barındırmaması nedeniyle harici bilgisayar sistemine bağımlılık
- Büyük boyutları nedeniyle kullanıcı dostu olmayan fiziksel yapı
- Yüksek işlem kapasitesi ve ekran kartı gereksinimleri
- Yalnızca kablolu kullanım imkânı



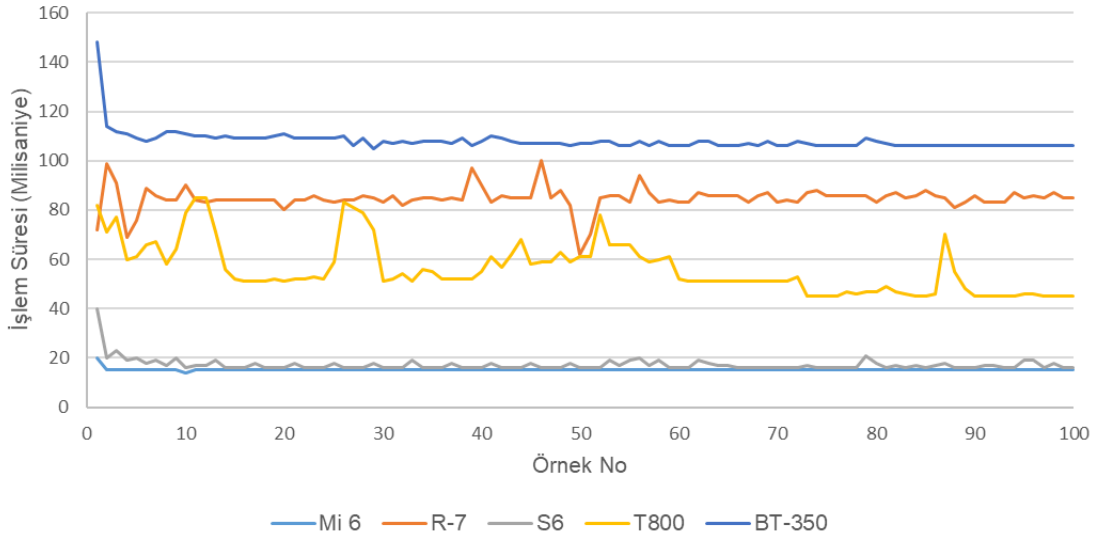
Şekil 3.17. Meta-2 model akıllı gözlük (Anonim, 2019d)

Tez çalışması kapsamında, RAHAT olarak temin edilebilen mobil cihaz ve akıllı gözlüklerden oluşan, Çizelge 3.1’de bilgileri verilen 5 adet cihaz üzerinde performans analizleri yapılmıştır.

Çizelge 3.1. Performans analizi yapılan mobil cihazlar

| No | Türü | Marka | Model |
|----|---------|------------------------|--------------|
| 1 | Gözlük | Osterhout Design Group | R-7 |
| 2 | Gözlük | Epson | BT-350 |
| 3 | Tablet | Samsung | Galaxy Tab S |
| 4 | Telefon | Samsung | S6 |
| 5 | Telefon | Xiaomi | Mi 6 |

Performans analizi için Harris Köşe Nokta Tespit İşleminin çalıştığı bir uygulama geliştirilmiştir. Uygulama, C++ dili ile OpenCV kütüphanesi kullanılarak yazılmıştır. Şekil 3.18’de verilen grafikte, 100 kez tekrarlanan işlem süreleri ölçülmüş, elde edilen sonuçlar sunulmuştur.

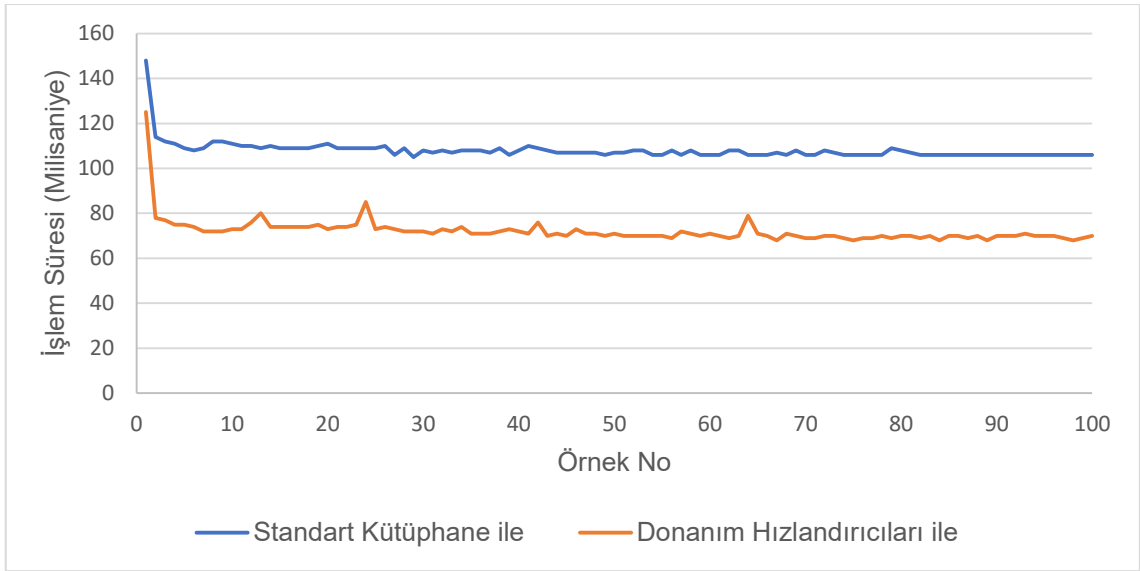


Şekil 3.18. Harris köşe tespit algoritmasının 100 iterasyonu sonucu toplanan işlem süresi verileri

Bu analiz çalışmasında hedeflenen sonuç, her iterasyonda aynı giriş verisi sunulmasına karşılık, işlem yapma süresinin her bir cihaza özgü stabil bir değerde süreklilik göstermesidir. Tez kapsamında yapılan çalışmalar neticesinde, işletim sistemi ve üzerine inşa edilen diğer bileşenlerin, yüksek işlem kapasitesi sunamayan cihazlar üzerindeki kaynak tüketimi nedeniyle, uygulamanın ihtiyaç duyduğu işlemci kaynağını anlık olarak bloke edebilmekte ve değişken bir performans sunumu ortaya çıkarmaktadır. Ek olarak, kullanılan test algoritmasının karmaşıklığı, yetersiz işlem gücüne sahip cihazlar üzerinde yüksek seviye yük oluşturmaktadır. Özellikle, Şekil 3.18’de yer alan ODG-R7 gözlük ve Samsung T800 cihazlarındaki çok değişken veri dağılımı bu durumun sonucu olarak yansımaktadır. Epson BT-350 model akıllı gözlük, diğer tüm cihazlardan daha düşük işlem yapma kapasitesine sahip olduğu için en yüksek iterasyon zamanına sahiptir. İterasyonların işleme zamanı R7 gözlük ve T800 tablete oranla daha stabil görülmektedir. Samsung S6 ve Mi 6 model telefonlar, yüksek işlem kapasitesi sayesinde diğer cihazlara oranla çok düşük iterasyon sürelerine ulaştığı görülmektedir. ODG-R7 gözlüğüne dair elde edilen bir diğer bulgu, üretim şekline bağlı olarak, yetersiz soğutma kapasitesi olduğu sonucudur. Bu sebeple, gözlük üzerinde çalıştırılan uygulamaların kaynak tüketimleri, hızlı bir şekilde ısınmaya neden olmaktadır. Bu da diğer cihazlara kıyasla iyi bir işlemci tüm devresine sahip olmasına karşın, yarı iletken teknolojisinin performansını etkileyen olumsuzlukların başında gelen yükselen ısıya karşılık, düşen performans sergilemesine neden olmaktadır.

Gömülü tüm devre işlemciler, üzerlerinde belirli operasyonları gerçekleştirmeye yönelik tasarlanmış tüm devre modülleri bulundurmaktadır. Bu modüller, işlemci

zamanından ayrı olarak, tasarlandıkları hedef işlemlere yönelik paralel ve yüksek performanslı çalışma imkânı sunmaktadırlar. Özellikle, görüntü işleme operasyonlarında yoğun olarak çalışan matematik işlemlere ait komut setleri, donanım hızlandırıcıları kullanan bir kütüphane aracılığıyla aktifleştirildiğinde önemli düzeyde performans artışı sağlamaktadır. Bu çalışmada, OpenCV kütüphanesi, Epson BT-350 model akıllı gözlük için optimize edilmiş derleme parametreleri ile donanım hızlandırıcılar etkin bir şekilde yeniden derlenmiştir. Bu cihaz için yenilenen test yine 100 iterasyon ile donanım hızlandırıcı dahil ve hariç iken Şekil 3.19'da elde edilen verilerle karşılaştırılmıştır.



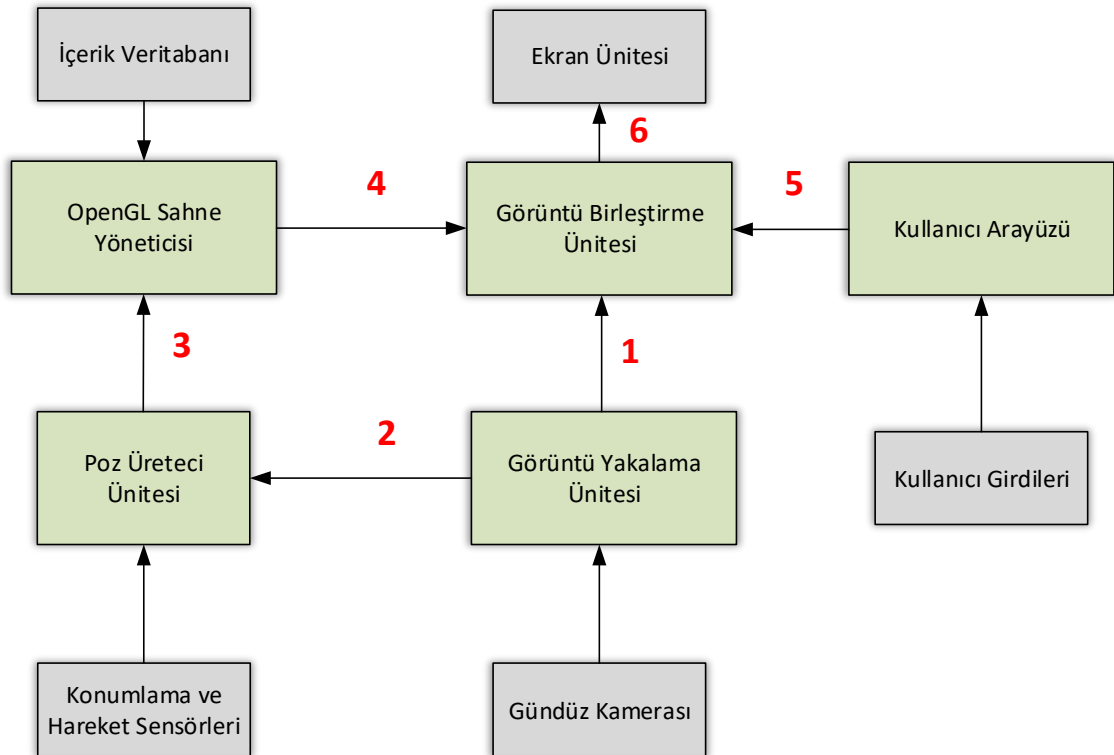
Şekil 3.19. Epson-BT350 model gözlük için OpenCV kütüphanesi performans karşılaştırma çizelgesi

Grafikte görüldüğü üzere, donanım hızlandırıcıların etkin olduğu çalışma sonucu %34 civarı performans artışı sağlanmıştır. Yapılan performans analizi, AG uygulamaları gibi yoğun hesaplama içeren uygulamaların doğrudan mobil cihazlar üzerinde çalıştırılması halinde, donanım bağımlı birtakım sebepler nedeniyle sürekli formda aynı düzeyde performans çıktısı alınamadığını göstermektedir. Bu sonuçla, mobil cihazlar üzerindeki işlem gereksiniminin düşürülmesine yönelik yapılacak olan çalışmalar, mümkün olan her donanım platformunda aynı seviyede eşdeğer performans sonuçlarına ulaşılması için önem taşımaktadır. Bu noktada tez çalışmasında tasarımı ve geliştirmesi yapılan mimari yaklaşımın önemi bir kez daha görülmektedir.

4. MATERYAL VE YÖNTEM

4.1. Görüntü Üretici Tasarımı ve Gerçekleştirilmesi

Artırılmış gerçeklik uygulamaları genel olarak Şekil 4.1'de yer alan mimari yapıya dayalı olarak geliştirilmektedir;



Şekil 4.1. Temel artırılmış gerçeklik uygulama mimarisi

1 numaralı akış;

Görüntü yakalama ünitesi, mobil cihaz üzerinde yer alan gündüz kamerası ile kullanıcının AG uygulamasını hedeflediği ortamdan alınan gerçek dünya verisini, sentetik üretilen zenginleştirme içerikleri ile birleştirme işlemine tabi tutulması için görüntü birleştirme ünitesine aktarmaktadır.

Saydam ekran teknolojisine sahip cihazlar üzerinde geliştirilen AG uygulamalarında, gerçek ortam görseli, cihaz ekranına yazılımsal süreçler ile aktarılmamakta, doğrudan kullanıcı algısı ile bütünleştirilmektedir. Bu nedenle 1 numaralı akış, saydam ekran barındıran akıllı gözlükler için geçerli değildir.

2 numaralı akış;

Gerçek ortamdan alınan veri üzerinde gerçekleştirilecek görüntü işleme teknikleri sonucu, kullanıcının mutlak pozunun üretilmesi sağlanmaktadır. Bu nedenle, görüntü birleştirme ünitesine aktarılan gerçek dünya verisinin bir kopyası, poz üretici ünitesine de aynı zaman diliminde aktarılmaktadır.

3 numaralı akış;

2 numaralı akıştan gelen görsel veri üzerinden üretilecek olan poz verisi, 1 numaralı akış ile görüntü birleştirme ünitesine aktarılan görsel veri ile aynı ana ait olduğu için, gerçek dünya verisi ile sentetik dünya verisinin zaman senkronizasyonu sağlanabilmektedir. Ancak, kullanıcının hareketli olması durumunda, görsel veride hareket bulanıklığı, ışık etkileri, bakış açısı darlığı gibi bozulmalar olacaktır. Ayrıca, görüntü işleme algoritmalarının her bir görüntü karesini işleme süresinin uzun olması nedeniyle ortaya çıkan gecikme probleminin önüne geçilmesi için farklı sensör verilerinin sinyal işleme teknikleri ile poz üretimine dahil edilmesi gerekmektedir. Bu amaçla, derinlik verisinden 3 boyutlu nokta bulutu analizi, ataletsel ölçüm sensörlerinden (IMU) hız ve yönelim tahminleri gibi farklı algoritmaların bütünleştirilmesi gibi işlemler poz üretici ünitesi tarafından gerçekleştirilmektedir.

Sonuç olarak üretilen kullanıcıya ait konum ve bakış açısı verileri, 3B sahne yöneticisi olan OpenGL bileşenine aktarılmaktadır. Böylelikle, sentetik dünyaya bakan sentetik kameranın (OpenGL kamerası) doğru konum ve bakış açısı verileri ile yeniden işlenmesi sonucu üretilen sentetik dünya görseli, gerçek dünya görseli ile doğru örtüşüme sahip olmaktadır.

4 numaralı akış;

Sahne yöneticisi, poz üreticiden alınan verileri işleyerek, artırılmış gerçeklik uygulamasının tasarımına göre şekillenen veri tabanından ilgili 3B model, animasyon ve 2B video, resim gibi verileri sentetik dünyaya dahil eder. Her bir uygulama çevriminde ürettiği sentetik dünya görüşüne ait görsel çıktıyı görüntü birleştirme ünitesine aktarır.

5 numaralı akış:

Kullanıcı, etkileşimlerle uygulama senaryosunda değişiklik yapabilir. Farklı bir sahneye geçiş, sahnedeki sentetik nesnelere etkileşim, animasyonların tetiklenmesi veya yönetilmesi, video ve ses gibi içeriklerin yönetilmesi bunlardan bazılarıdır.

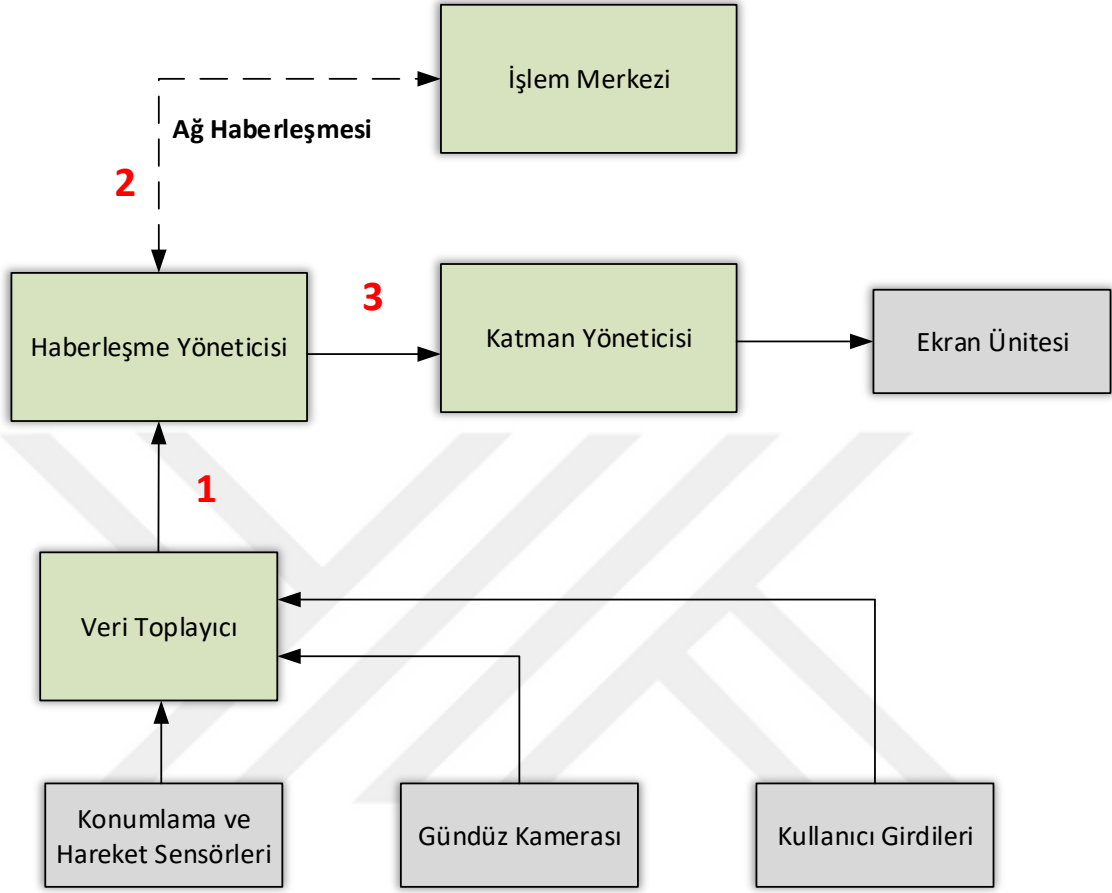
6 numaralı akış:

Gerçek dünya algısının zenginleştirilmesinden sonra, kullanıcı arayüzü katmanının en üstte ve olay etkin bir formda yerleştirilmesi ile birlikte üretilen son görsel veri, kullanıcı ekranına aktarılmaktadır.

Bu mimari, tüm işlem süreçlerinin mobil cihaz üzerinde olduğu, grafik işlemci, bellek ve işlemci gereksinimlerinin, AG uygulamasının tasarımına bağlı olarak değiştiği bir yapı tanımlamaktadır. Özellikle, 3 boyutlu modellemelerin yüksek detay yapısında olması, mobil cihazlar üzerindeki kısıtlı kaynaklar açısından olumsuz bir durum oluşturmaktadır. Birden çok kullanıcının aynı sentetik dünyada, farklı bakış açısında icra ettiği AG uygulamalarında, her bir kullanıcının konum-yönelim, etkileşim bilgilerinin diğer kullanıcılara ait cihazlar ile paylaşılması problemi ortaya çıkmaktadır. Ayrıca bu tip uygulamalarda, sentetik dünyanın bağımsız çalışan birden çok kopyası ortaya çıkmaktadır. Bu durum yazılım mimarisi açısından yönetmesi oldukça güç bir sonuç doğurmaktadır.

Bu problemlerin çözümüne yönelik olarak yeni bir metot geliştirilmesi gerekmektedir. Bu metot, kaynak tüketimi açısından uygulama güçlüğü yaratan bileşenleri, mobil cihazlar üzerinden soyutlayarak, ağ altyapısı ile erişilebilen güçlü bir bilgisayar ünitesine taşınmasına imkân tanımalıdır. Bununla birlikte, mobil cihaz üzerinde zenginleştirilmiş AG görselinin kullanıcıya sunulabilmesi için, her bir görsel veri kaynağının (gerçek ve sentetik dünya görüntülerinin) bir katman olarak ele alındığı yeni bir GÜ bileşeni mimarisi geliştirilmelidir.

Şekil 4.2’de tasarlanan yeni mimari modelinde görüldüğü üzere, mobil cihaz üzerinde yer alan işlem yükü azaltılmıştır. Sunucu üzerinde oluşturulan sentetik sahneye ait görseller, birer katman olarak ele alınmış ve yeni bir mimari tanımlanmıştır.



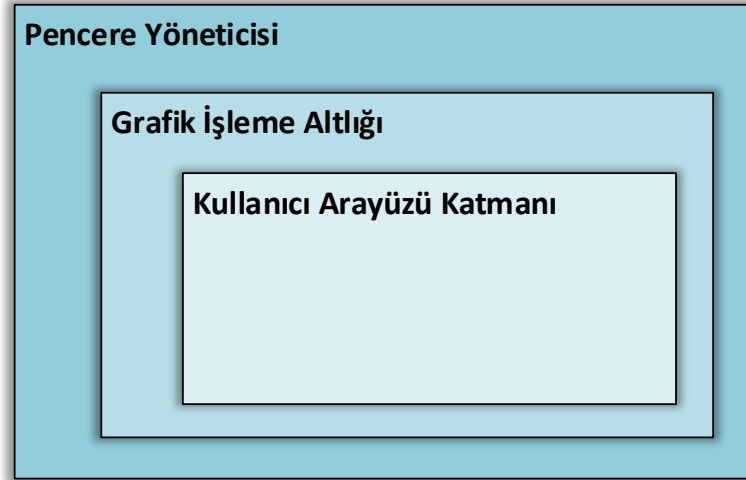
Şekil 4.2. Çok katmanlı görüntü üretici mimarisi

4.1.1. Ana Katman Mimarisi

GÜ yazılımı, görsel çizimlerin birleştirilerek mobil cihaz ekranında görselleştirilmesini sağlayan ana pencere uygulaması üzerine inşa edilmiştir. Bu uygulama,

- Pencere yöneticisi,
- Grafik işleme altlığı,
- Kullanıcı arayüzü katmanı,

olmak üzere 3 ana bileşenden oluşmaktadır. Şekil 4.3’de yer alan tasarlanmış ana katman mimarisinde görüldüğü gibi pencere yöneticisi tüm katmanlar için taşıyıcı pencere görevi görmektedir.



Şekil 4.3. Ana katman mimarisi

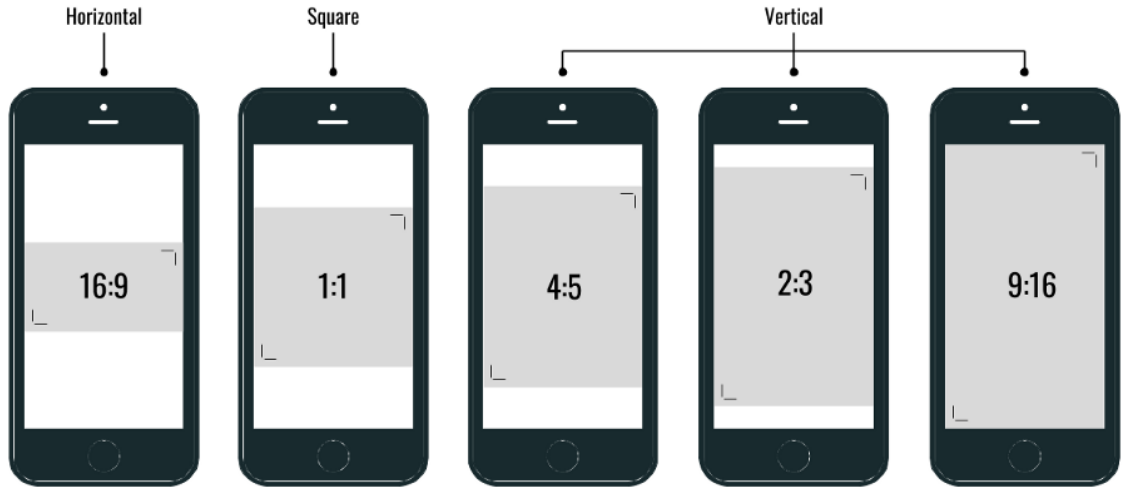
4.1.1.1. Pencere Yöneticisi

Pencere yöneticisi, mobil cihaz işletim sistemi üzerinde yer alan grafik arayüz kütüphanelerini kullanarak, cihaza bağlı bir ekran sürücüsünde görselleştirilebilir yazılım arayüzü oluşturulmasını sağlamaktadır. Bu çalışma kapsamında, Qt kütüphanelerinde yer alan QWidget sınıfları kullanılarak pencere yöneticisi inşa edilmiştir.

Yazılım arayüzü, eş zamanlı olarak icra edilen birden çok pencere ögesi barındırmaktadır. En alt katmanda yer alan taşıyıcı pencere, cihaz ekranının desteklediği farklı çözünürlüklere uyumlu olarak pencere boyutlarını otomatik olarak düzenleyen taşıyıcı pencere görevini icra etmektedir. Aynı zamanda, artırılmış gerçeklik uygulamalarının gerçekleştirilmesinde önem arz eden görünüş oranı (*aspect ratio*)'nın korunması taşıyıcı pencerenin kontrolündedir.

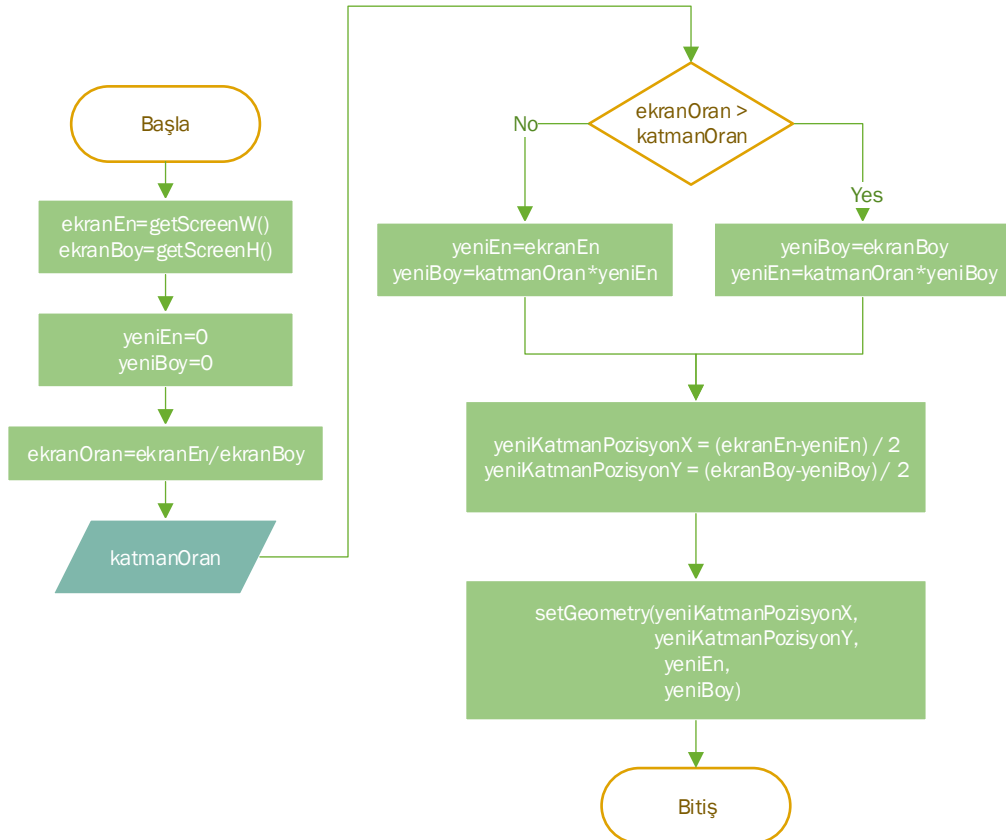
Görünüş oranı uyumluluğu, tez kapsamında geliştirilen mimari için önemli bir konudur. Farklı bir grafik altyapısı altında üretilen görüntülerin, başka bir ekran üzerinde görselleştirilmesi sırasında görünüş oranı uyumsuzluğu ortaya çıkabilmektedir. Bu durumun önüne geçmek için, görüntü üreticiye gelen tüm görüntü akış katmanlarının aynı görünüş oranında olduğu varsayılarak, taşıyıcı pencere altında yer alan grafik işleme altlığı ve kullanıcı arayüzü katmanlarının görünüş oranları, dış kaynaktan gelen görüntünün görünüş oranı doğru kabul edilerek yeniden

hesaplanmalıdır. Bu hesaplama sonucunda, eğer mobil cihaza ait ekranın en-boy oranı, katman verisinin en-boy oranı ile eşdeğer ise taşıyıcı pencere tam ekran, değil ise Şekil 4.4’de verilen örnek görselde de görüldüğü üzere üst veya yan kenarlarda ortak görünüş oranı sağlanacak şekilde boşluklar bırakılarak görselleştirilecektir.



Şekil 4.4. Farklı görünüş oranına sahip görüntü kaynaklarının görselleştirilmesi (Moriarty, 2017)

Görüş oranı uyumsuzluğunu kontrol etmek amacıyla geliştirilen algoritma Şekil 4.5’de aktarılmıştır.



Şekil 4.5. Görünüş oranı uyumsuzluğunu kontrol eden algoritma akışı

4.1.1.2. Grafik İşleme Altlığı

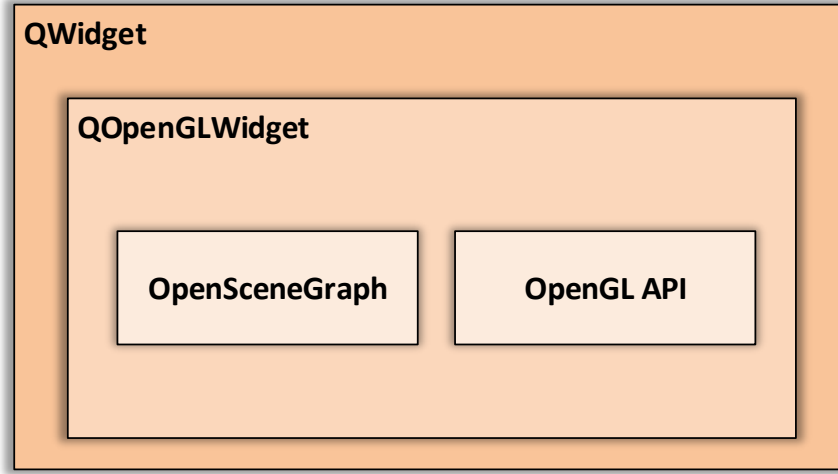
Grafik işlemleri, OpenGL sahnesinin icra edilmesi sonucu, sanal kamera görüş alanını içeren görsel verinin kullanıcıya sunulduğu arayüz bileşenidir. Bu görsel veri, AG uygulamasının zenginleştirme verisi olan sentetik bilgisayar üretimi veri katmanını içermektedir. OpenGL, 2 ve 3 boyutlu grafiklerin bilgisayar tarafından çizilmesini sağlamak için kullanılan açık kaynak bir API diğer bir deyişle yazılım kütüphanesidir. Bu kütüphane, yazılım geliştiriciye grafik donanımını yönetme imkânı sunmaktadır. OpenGL, bilgisayar ekranına grafik çizim işlemlerini standartlaştırmıştır. İşletim sisteminden ve işletim sisteminin üzerinde çalıştığı platformdan bağımsızdır. Bu sayede, taşınabilir yazılım geliştirmeye imkân veren kütüphane, tüm grafik kartlarıyla uyumlu çalışmakta ve yazılım geliştiriciyi donanım farklılıklarından soyutlamaktadır. Bu nedenlerle, OpenGL kütüphanesi popüler bir yazılım kütüphanesi olmuştur (Wagner ve Schmalstieg, 2003).

OpenGL, işletim sistemi altında çalışan pencere yöneticisinden bağımsızdır. Bu özelliği sayesinde, Windows, Mac, Linux, Android ve diğer işletim sistemlerinde sorunsuz bir şekilde çalışabilmektedir. Böylelikle, geliştirilen yazılımın çoklu platform desteği grafik işlemleri açısından sağlanabilmektedir (Wagner ve Schmalstieg, 2003).

Bu çalışmada, OpenGL API üzerine inşa edilmiş, açık kaynak üst seviye grafik kütüphanesi olan OpenSceneGraph (OSG) kullanılmaktadır. Doğrudan OpenGL kütüphanesinin kullanılmamasının nedeni, bu kütüphanenin, daha düşük kod yazma maliyeti ile etkinliği yüksek uygulamalar geliştirebilmeyi sağlamasıdır. OSG, görsel simülasyon, oyun, sanal gerçeklik, bilimsel görselleştirmeler ve modelleme gibi alanlarda kullanılmak üzere OpenGL konsorsiyumu olan KHROS GROUP tarafından önerilen yüksek seviye kütüphaneler listesinde yer almaktadır (Pavlik ve Vance, 2012).

Tez kapsamında, Şekil 4.3’de yer alan, taşıyıcı pencere görevi gören pencere yöneticisi altında, Qt kütüphaneleri içerisinde yer alan QOpenGLWidget kütüphanesinden kalıtılmış bir grafik işleme altlığı geliştirilmiştir. Bu altlık, OpenGL bağlamını (*context*), OSG kütüphanesine aktararak, tüm sahne yönetimi işlemlerini bu kütüphaneye devretmektedir. OSG sahnesinde yer alan sanal kameralar, icra ettikleri grafik çizimlerini, QOpenGLWidget bağlamı aracılığıyla taşıyıcı pencere olan pencere

yöneticisi üzerine çizerek görselleştirir. Şekil 4.6’da bu yapının hiyerarşisi ifade etmesi için çizilen görsel yer almaktadır.



Şekil 4.6. Grafik işleme altlığı yapısı

4.1.1.3. Kullanıcı Arayüzü Katmanı

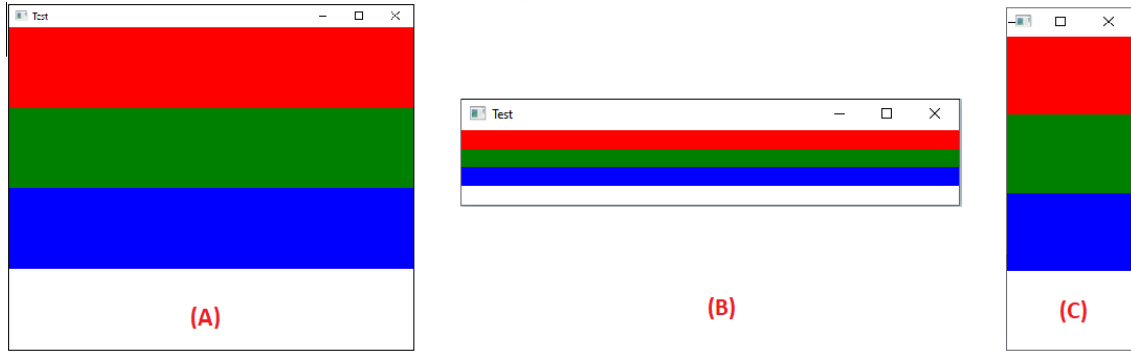
Artırılmış gerçeklik uygulamalarında, kullanıcı etkileşiminin en temel yöntemi dokunmatik ekran arayüzleridir. Bu arayüzler, kullanıcı girdilerini almak, AG senaryolarını yönetmek, sahnede yer alan 3B veya 2B nesnelere etkileşime girmek, uygulama yönetimini (açma, kapatma, listeleme vb.) gerçekleştirmek gibi görevleri icra etmektedir.

Tez kapsamında, esnek ve modüler mimarisi ve çoklu platform desteği nedeniyle Qt kütüphanesi altında yer alan QQuickWidget kütüphanesinden türetilen kullanıcı arayüzü sınıfı geliştirilmiştir. Qt Quick kütüphanesi, QML dili ile uygulama geliştirmeye imkân veren bir yazılım geliştirme bileşenidir. QML dilini işleyen bir dil motoru barındırmaktadır. Bu dil ile geliştirilmiş olan grafik arayüz yazılımlarını icra ederek görselleştirilmesini sağlar. Qt Quick kütüphanesi, qml uzantılı kod dosyalarını çalışma zamanında icra ederek kullanıcı arayüzlerini üretmektedir. Şekil 4.7’de kullanıcı arayüzüne örnek olarak yazılan qml dosyası görülmektedir. Her bir grafik arayüz bileşeni, ayrı birer modül olarak ele alınmaktadır. Modül özellikleri yazılım geliştirici tarafından düzenlenebilmektedir. Adaptif ölçeklendirme kabiliyetiyle Qt Quick kütüphanesi ile duyarlı (*responsive*) kullanıcı arayüzleri geliştirilebilmektedir. Bu kavram, tek bir grafik arayüz geliştirerek, duyarlı yapısı ile farklı çözünürlüklerde çalışan ekran ünitelerine adapte olan tasarımların gerçekleştirilebilmesini sağlamaktadır.

```
main.qml*
1  import QtQuick 2.12
2  import QtQuick.Controls 2.5
3
4  ApplicationWindow {
5      id: window
6      visible: true
7      width: 640
8      height: 480
9      title: qsTr("Test")
10
11     Item {
12         anchors.fill: parent
13
14         Rectangle{
15             id: rect1
16             color: "red"
17             anchors.top: parent.top
18             anchors.left: parent.left
19             anchors.right: parent.right
20             height: parent.height/4
21         }
22         Rectangle{
23             id: rect2
24             color: "green"
25             anchors.top: rect1.bottom
26             anchors.left: parent.left
27             anchors.right: parent.right
28             height: parent.height/4
29         }
30         Rectangle{
31             id: rect3
32             color: "blue"
33             anchors.top: rect2.bottom
34             anchors.left: parent.left
35             anchors.right: parent.right
36             height: parent.height/4
37         }
38         Rectangle{
39             id: rect4
40             color: "white"
41             anchors.top: rect3.bottom
42             anchors.left: parent.left
43             anchors.right: parent.right
44             height: parent.height/4
45         }
46     }
47 }
```

Şekil 4.7. Yazılan örnek qml kod dosyası

Kod örneğinde görülen *Rectangle* modülü ile tanımlanan 4 adet her biri kendi rengine sahip alanların, birbirleri ile olan konumsal ilişkilerinin *anchors* değerleri ile ayarlanması sonucu duyarlı arayüz tasarımına örnek teşkil etmiştir. Kod üzerinde görüldüğü gibi *Rectangle* bölgelerin hiçbiri için metrik konum veya ebat verisi girilmemiştir. Bu sayede kullanıcı arayüzü ana katmanı olan *parent* değişkeninin değerlerini referans alarak her bir grafik işlemi sürecinde yeniden hesaplanarak kullanıcı arayüzüne çizilmektedirler. Değişken çözünürlük ve ölçek altında kullanıcı yazılımın tepkisi Şekil 4.8’de görülmektedir. A görselinde en boy oranı denk, B ve C görsellerinde ise orantısız bir fark oluşturulmuş, buna karşılık içerik verisinin duyarlı tepkisi ile konum ve boyut değerlerinin otomatik değiştiği gözlemlenmiştir.



Şekil 4.8. Duyarlı kullanıcı arayüzü görüntüleri

Kullanıcı arayüzünün duyarlı olması, farklı çözünürlüklerdeki mobil cihaz ekranları için tek bir yazılım tasarımı ile uygulanabilir ve düşük geliştirme maliyetine sahip yazılımlar üretilmesine imkân tanımaktadır. Çizelge 4.1’de bölüm “3.3. Giyilebilir ve Mobil Donanım Platformları” başlığı altında incelenen donanımların ekran çözünürlükleri yer almaktadır. Tasarlanan kullanıcı arayüzü katmanı bu farklılıkları yazılım geliştirici açısından önemsiz kılmaktadır.

Çizelge 4.1. Mobil cihazların ekran çözünürlükleri

| Mobil Cihaz | Ekran Çözünürlüğü (en x boy) | Görünüş Oranı (en/boy) |
|-----------------|------------------------------|------------------------|
| ODG-R7 | 1280 x 720 | 16:9 |
| Epson BT-350 | 1280 x 720 | 16:9 |
| Mi 6 | 1920 x 1080 | 16:9 |
| Samsung S6 | 2560 x 1440 | 16:9 |
| Galaxy Tab T800 | 2560 x 1600 | 16:10 |

4.1.2. AG Katmanları Mimarisi

AG uygulamaları, doğası gereği birden çok katman barındırmaktadır. En alttan yukarıya doğru bunlar, gerçek dünya görüntüsü, sentetik dünya görüntüsü ve kullanıcı arayüzü katmanlarıdır. Uygulama geliştiriciler, en basit düzeyde bir AG uygulaması geliştiriyor olsalar dahi bu katmanları uygulama içerisine entegre etmektedirler.

Gerçek dünya görüntüsü ve sentetik dünya görüntüsü, optik kamera modeli açısından eş parametreler ile ayarlanmalıdırlar. Bu parametreler, yatay görüş açısı, odak uzaklığı (*focal length*), görünüş oranı (*aspect ratio*) ve optik merkez noktası (*optical center*) parametreleridir (*intrinsic parameters*). Bu parametrelerin, eşlenememesi durumunda, sentetik dünya verisi, gerçek dünya verisi üzerindeki referans konumlara ve bakış açısına oturmaz. Bu durumda, birbirinden alakasız iki farklı verinin üst üste yerleştirilmesi ile oluşan bir görüntü elde edilmiş olacaktır.

İçsel parametreler (*intrinsic parameters*), noktasal kamera modeli (*pin hole camera model*) yöntemi kullanılarak hesaplanabilmekte ve uygulamaya alınabilmektedir. Şekil 4.9'da görülen matris ifadesi, içsel parametreleri kullanarak, kamera ile algılanan 3 boyutta bir noktanın, kamera düzleminde oluşan iz düşümünün 2 boyutlu yansımalarını hesaplamayı sağlamaktadır.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Şekil 4.9. Noktasal kamera modeli için tanımlanan 3B-2B dönüşümü (Anonim, 2019c)

f_x ve f_y değerleri, yatay ve dikey odak uzaklığını, c_x ve c_y ise yatay ve dikey optik merkez noktasının kamera sensörü üzerine düşen piksel konumunu ifade etmektedir. Odak uzaklığı, yatay görüş açısı bilinen kameralar için Denklem 4.1 bağıntısı aracılığı ile hesaplanabilmektedir.

$$Fovh = 2 \tan^{-1}(N / (2f_x)) \quad (4.1)$$

Optik merkez ve odak uzaklığı verileri, kamera kalibrasyon yöntemleri ile çalışma zamanında kameradan toplanan görüntü verileri üzerinde çalıştırılan görüntü işleme yöntemleri aracılığıyla hesaplanabilmektedir. Denklem 4.1. aracılığıyla hesaplanarak elde edilen görüş açısı değeri, OpenGL kamerasına perspektif parametresi olarak ayarlanır.

Konum ve bakış açısını takip eden algoritmalar, kullanıcının gerçek dünyada bulunduğu konumu ve bakış açısını (*extrinsic parameters*) doğru bir şekilde sentetik dünya kamerasına uygulanmasını sağlamaktadır. Konum ve bakış açısı verilerinin doğru takip edilmesi ve sentetik dünya kamerasına uygulanması sonucu, sanal nesnelere gerçek dünya üzerinde doğru yerde ve boyutlarında algılanması mümkün olmaktadır. Nesnelere etrafında gezmek, üst açıdan veya yan açılardan bakmak gibi eylemlerde doğru perspektif çizimlerinin hesaplanması bu şekilde sağlanmaktadır.

Şekil 4.10 ile ifade edilen dönüşüm modelinde görüldüğü üzere, gerçek dünya görseli üzerinde bir konuma, sentetik dünya görsellerinin doğru bir şekilde yerleştirilmesi için, grafik işleme bileşenine, AG uygulaması için kullanılan kamera ile içsel ve dışsal parametrelerin doğru bir şekilde uygulanması gerekmektedir.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2B Görüntü Koordinatları
Kamera İçsel Parametreleri
Kamera Dışsal Parametreleri
3B Gerçek Dünya Konumu

R: Bakış Açısı T: Konum

Şekil 4.10. 3B-2B tamamlanmış dönüşüm modeli(Anonim, 2019c)

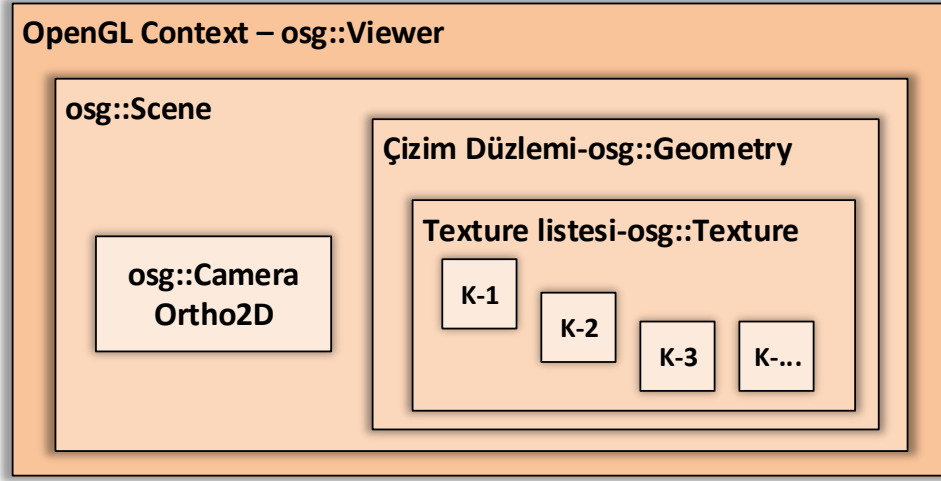
Tez kapsamında geliştirilen katman modeli, sentetik dünya işlemlerini mobil cihaz dışında, farklı bir bilgisayar ünitesinde çalıştırılmasını sağlamaktadır. Sahne yönetimi ve sentetik görüntü üretimi mobil cihaz dışına taşındığı için, yukarıda bahsi geçen içsel ve dışsal parametrelerin, yazılım çalıştırılmaya başlandığında tek seferde, kullanılan mobil cihaza özgü veriler ile dışarıdaki sahne yöneticisine aktarılması doğru parametreler ile çalışılmasını mümkün kılacaktır.

4.1.2.1. Çoklu Katman Mimarisi

Dış sahne yöneticisinden alınan işlenmiş gerçek dünya görüntüsü, üretilmiş sentetik dünya görüntüsü ve ek olarak aktarılabilecek diğer katman görsel verileri, tez kapsamında tasarlanan artırılmış gerçeklik katmanları üzerine işlenerek birleştirilmiş görüntü kullanıcı ekranına yansıtılmaktadır. Bu katmanlar, OpenGL yöntemleri ile birleştirilmekte ve AG görüntüsü elde edilmektedir. Çoklu katman mimarisi, her bir görüntü kaynağını bir katman olarak ele almaktadır. Bu görüntü kaynakları gerçek kamera verileri ile doldurulabileceği gibi, bilgisayar grafikleri ile üretilmiş sentetik dünya görselleri ile veya sabit tasarımlı 2 boyutlu kaplama (*overlay*) verisi ile doldurulabilmektedir. Oluşturulan mimari, herhangi bir kaynağı desteklemektedir. Veri, bir dosyadan okunan video, ağ üzerinden gelen görüntü akışı veya uygulama üzerinde üretilen herhangi bir görsel olabilmektedir.

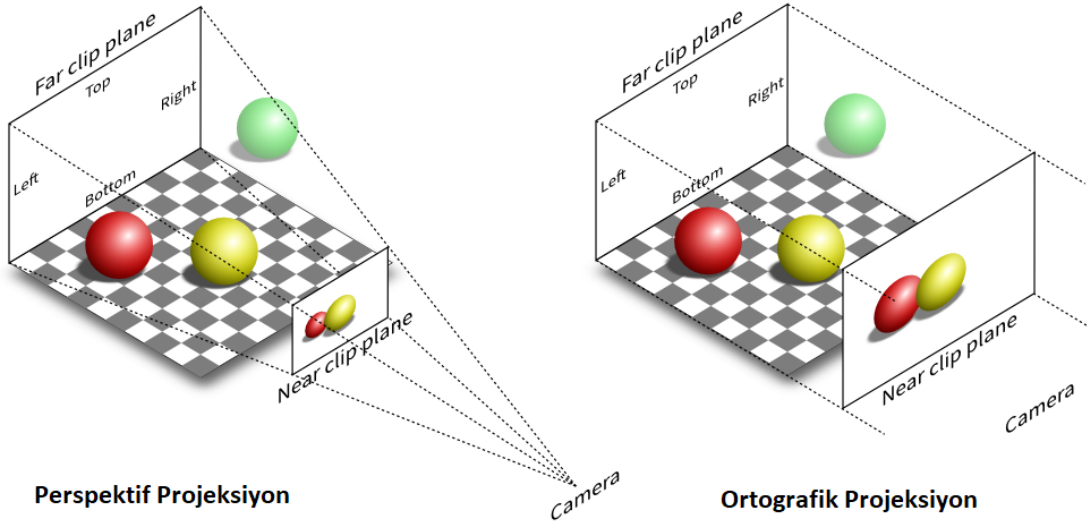
Katmanların birbirleri ile ilişkileri, öncelikli sıralama ve saydamlık işleme yöntemleri ile belirlenmektedir. Öncelikli sıralama, bir artırılmış gerçeklik katmanının hangi katmanlar arasında olduğunu, saydam olmayan piksel verilerinin üst üste binmesi durumunda hangi katmanın görünür olacağını belirlemektedir. Saydamlık işleme yöntemi ise, bir katman verisi içerisindeki saydam olmayan piksellerin katman seviyesinde görünür olmasını, saydam piksellerin ise katman maskesi olarak kullanılmasını sağlamaktadır.

Katman Modeli, OpenGL yapısı üzerine kurulmuştur. OSG kütüphanesi, OpenGL kamerası, sahne yönetimi ve 3B nesne organizasyonu için kullanışlı yöntemler ve üst seviye fonksiyonlar sunmaktadır. Bu fonksiyonlar ve yöntemler kullanılarak tasarlanan katman modeli Şekil 4.11’de yer almaktadır.



Şekil 4.11. Çoklu katman mimarisi

QOpenGLWidget aracılığı ile pencere yöneticisi altında üretilen OpenGL bağlamı (*context*), OSG kütüphanesi tarafından sunulan `osg::Viewer` sınıfına aktarılmaktadır. Böylelikle pencere yöneticisindeki OpenGL alanının, OSG tarafından güncellenebilir olması sağlanmıştır. Viewer içerisinde `osgScene` sınıfından türetilmiş bir OpenGL sahne nesnesi eklenmiştir. Tüm çizim işlemleri bu sahne üzerinde gerçekleştirilecektir. OSG sahnesine ait kamera, ortogonal formasyon parametreleri ile düzenlenmiştir. AG sahneleri, perspektif formasyonda işletilmektedir. Ancak, tez kapsamında geliştirilen katman yapısı, 3B sahneyi doğrudan işlememekte, sahneyi işleten yazılımın çıktısını katman verisi olarak alıp işlemektedir. Bu nedenle, herhangi bir kaynaktan alınan sahne görsel verisi, perspektif işlemine tabi olmadan doğrudan OpenGL sahnesinde çizilmektedir. Katmanları işleyerek görsel veriyi üreten kamera, bilinen AG sahnelerindeki gibi harici parametreleri (bakış açısı ve konum) değiştirilmeyen, sabit bir sahne görüşü altındaki düzlem objesi üzerinde yer alan doku (*texture*) verilerinin çizimini yapmaktadır. Şekil 4.12’de perspektif kamera ve ortogonal kamera arasındaki fark görülmektedir.



Şekil 4.12. Perspektif projeksiyon ve ortografik projeksiyon karşılaştırması (Anonim, 2019b)

Katman verisinin grafik kartı ile işlenmesi için her bir katmana ayrı birer doku (*texture*) nesnesi tanımlanmıştır. Şekil 4.11’de K1, K2, K3 olarak isimlendirilmiş olan doku nesnelere, grafik sahnesinde oluşturulan düzlem objesinin yüzey kaplama verisini tanımlamaktadır. Düzlem objesi, OpenGL kamerası görüş alanında bulunan, kameranın görüş alanı boyunca görüntüyü tanımlayan her bir piksel verisinin, kendi koordinatı ve rengi ile üzerine çizildiği 3B nesnedir. Her bir katmana ait doku verisi, diğer katmanlar ile olan ilişkisini ve maske (saydamlık) verisini hesaplayan *shader* programı ile işleyerek düzlem üzerine yerleştirmektedir.

4.1.2.2. Saydam Katman Yöneticisi

Shader programları, grafik kartı ile işlenen sanal nesnelerin şekil ve renkleri üzerine son çıktılarının oluşturulması amacıyla kullanılan programlardır. İlk kullanım alanı gölgeleme hesaplamaları yapmak amacı taşıyan bu programlar, günümüzde özel efektler ve gölgeleme işlemlerinin dışında video *post-process* gibi işlerde kullanıldığı gibi grafik işlemleri ile ilgisiz yoğun hesaplama içeren uygulamalarda da kullanılmaktadır (Kessenich ve ark., 2004).

Çoklu katman yapısında, katmanlar arası ilişki iki kavram ile sağlanmaktadır. İlki, katmanların sıralaması yani öncelik planlamasıdır. Yazılım geliştirici tarafından belirlenecek olan katman sıralaması, saydam olmayan ortak piksel noktaları bulunan katmanların hangisinin önde olacağını ve kullanıcı tarafından görüleceğini belirlemektedir. İkinci kavram ise 4 renk kanalından oluşan AG görsellerinin temel

karakteristiği olan saydam görüntü kanalının işlenmesi ile maskelenecek piksel bölgelerinin belirlenmesine dayanmaktadır.

Bilgisayar grafikleri birçok piksel formatının kullanılabilmesine imkân vermektedir. Tez kapsamında geliştirilen katman mimarisi *RGBA (Red Green Blue Alpha)* formatındaki 3 kanal renk 1 kanal saydamlık ifade eden toplam 4 kanaldan oluşan piksel formatı ile çalışmaktadır. Saydam katman yöneticisi, katman verileri üzerinde katman sıralaması ve saydamlık kanalını kullanarak saydamlık harmanlama (*alpha blending*) algoritmasını uygulamaktadır.

Saydamlık harmanlama algoritması, pek çok yöntemle uygulanabilir bir algoritmadır. En basit haliyle, katman verilerinin her bir pikseline ait renk ve saydamlık kanal verilerini en alttan en üste doğru işleyerek nihai görsel üretilebilir. Şekil 4.13’de geliştirilen basit bir harmanlama algoritmasına ait kod görülmektedir;

```
void bubbleSort(int pArrayLen, int pAryOrders[], int pAryIndexes[])
{
    for (int i = 0; i < pArrayLen-1; i++)
    {
        for (int j = 0; j < pArrayLen-i-1; j++)
        {
            if (pAryOrders[j] > pAryOrders[j+1])
            {
                swap(&pAryOrders[j], &pAryOrders[j+1]);
                swap(&pAryIndexes[j], &pAryIndexes[j+1]);
            }
        }
    }
}

void blending()
{
    QList<Layer> tLayers;
    //append layers to list
    //...

    int *tLayerIndexes;
    //append layer indexes to array
    //...

    int *tLayerOrders;
    //append layer orders to array
    //...

    bubbleSort(tLayers.length(), tLayerOrders, tLayerIndexes);

    QImage tFinalImage;
    tFinalImage.fill(Qt::black);

    for(int t = 0; t<tLayers.length(); t++)
    {
        for(int i = 0; i<tLayers.at(tLayerIndexes[t]).image.width; i++)
        {
            for(int j = 0; j<tLayers.at(tLayerIndexes[t]).image.width; j++)
            {
                Pixel tLayerPixel = tLayers.at(tLayerIndexes[t]).getPixel(i,j);

                if(tLayerPixel.A >= 0.5)
                    tFinalImage.setPixel(i,j, tLayerPixel.R, tLayerPixel.G, tLayerPixel.B, tLayerPixel.A);
            }
        }
    }
}
```

Şekil 4.13. İşlemci üzerinde koşan sıra tabanlı basit bir harmanlama algoritması

Örnek algoritma, öncelikle katmanları 0 en altta olacak şekilde katman sıralamasına göre “*bubblesort*” sıralama algoritması ile sıraya koyar. Daha sonra her bir katmanın saydamlık değerini inceler. Saydamlık değeri yarı değerden (0.5) büyük olan pikseller görünür, altında olan pikseller ise maske altında kalır mantığıyla gizleyerek sonuç görselini üretir. Bu algoritma, katman verilerini işlemci gücü ile hesaplamaktadır. Her bir katman verisini sırası ile ele alır, her katmana ait yüz binlerce piksel verisini sırası ile işler. Bu yöntem, çok yüksek sistem kaynağı tüketimi sebebiyle kötü bir performans sunar. Bunun yerine, saydamlık harmanlama algoritması, grafik işlemci üzerinde çalışan *fragment shader* programı ile gerçekleştirilirse, grafik işlemci mimarisinin paralel işlem gücünü kullanmayı mümkün kılar. Grafik işlemcilerin ana amacı bu tür programları yüksek performans ile paralel bir şekilde işlemektir. Şekil 4.14’de saydamlık harmanlama algoritmasının *shader* programı formunda yazılmış biçimi yer almaktadır.

```
static const char FragmentShaderProgram[] =
"   varying vec2 texCoord;                               \n"
"   void main() {                                         \n"
"       uniform sampler2D layer_1;                        \n"
"       uniform sampler2D layer_2;                        \n"
"       vec4 texColor_1 = texture2D(layer_1, texCoord);   \n"
"       vec4 texColor_2 = texture2D(layer_2, texCoord);   \n"
"       if(texColor_1.a >= 0.5)                           \n"
"           gl_FragColor = vec4(texColor_1.b, texColor_1.g, texColor_1.r, 1); \n"
"       else if(texColor_2.a >= 0.5)                       \n"
"           gl_FragColor = vec4(texColor_2.b, texColor_2.g, texColor_2.r, 1); \n"
"   }                                                       \n";
```

Şekil 4.14. Shader programı formunda yazılmış saydamlık harmanlama algoritması

Şekil 4.14’de yer alan *fragment shader* programı, 2 katman için saydamlık harmanlama algoritmasına örnek teşkil etmektedir. Program içerisindeki *if-else* yapısı, yukarıdan aşağıya işlenerek önce en üstteki katmanı sonra en alttaki katmanı işleme tabi tutmaktadır. Maskeleme kontrolü yine saydamlık kanalı verisinin yarıdan (0.5) büyük veya küçük olması durumuna göre çalışmaktadır. Şekil 4.15’de yer alan örnek görselde, *shader* programının bir katmana ait saydamlık kanalı ile *RGB* kanallarını işlemesi sonucu katman üzerindeki etkisi görülmektedir. Birden çok katmanın olduğu durumda, katmanlar arası sıralamayı belirleyen sıralama numarası hangi katmanın saydam olmayan piksellerinin üstte yer alacağını, yani kullanıcı tarafından uygulama ekranında görülebileceğini belirler.

Shader programları, statik olarak ayarlanabilmekle birlikte çalışma zamanında bir dosyadan okunarak veya çalışma zamanında belirli bir algoritmayla üretilerek devreye alınabilir. Kod yapısı Şekil 4.14’de görüldüğü gibi C++ diline benzemektedir. *Uniform* değişkenler, *shader* programına dışarıdan verilen parametreleri, *texture2D* fonksiyonu parametre olarak aktarılan dokuya ait piksel verisini, yine parametre olarak aktarılan *texCoord* piksel konum verisi ile döndürmektedir. *gl_FragColor* değişkeni hedef piksel pozisyonundaki pikselin içerdiği veriyi temsil eder. Görüldüğü üzere program içerisinde herhangi bir döngü söz konusu değildir. OpenGL *Shader* dili ile yazılan bu program, grafik işlemciye yüklendiği andan itibaren, QOpenGLWidget sınıfının `update()` (güncelleştirici) metodunun her çağırılışında çalışarak, grafik bellekte yer alan doku (*texture*) verilerini alıp her bir piksel için paralel olarak yazılan algoritmayı icra eder. Bu sayede hem masaüstü hem de mobil platformlar için mümkün olan en yüksek performans sağlanmış olur.

Fragment shader programının çalışması için öncesinde *Vertex shader* programı çalışmalıdır. *Vertex shader* programı, OpenGL sahnesinde yer alan objelerin geometrilerinin oluşturulmasını sağlamaktadır. Katman mimarisinde yer alan düzlem objesinin geometrisinin grafik işlemci tarafından oluşturulabilmesi için yazılan Şekil 4.15’de yer alan *Vertex shader* programının devreye alınması gerekmektedir.

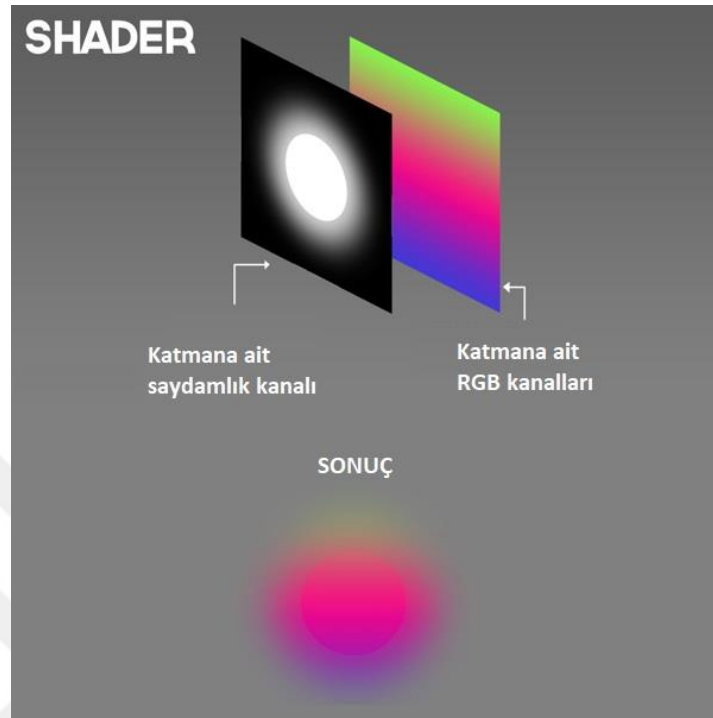
```
static const char VertexShaderProgram[] =
    "varying vec2 texCoord; \n"
    "void main() { \n"
    "  texCoord = gl_MultiTexCoord0.xy; \n"
    "  gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex; \n"
    "} \n";
```

Şekil 4.15. Örnek Vertex shader programı

Vertex shader programı, fragment shader programında olduğu gibi tüm piksel verilerini paralel olarak işlemektedir. *gl_Vertex* verisi OpenGL sahnesinde yer alan nesnelere ait *vertex* birimlerin (3 boyutlu pikseller) konumunu ifade etmektedir. Şekil 4.10 ile ifade edilen 3B-2B dönüşüm matrisi, *vertex shader* içerisinde;

```
gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
```

kod satırında görülmektedir. Buradaki projeksiyon matrisi, ortogonal projeksiyon modeline göre oluşturulmuş matristir.



Şekil 4.16. Shader programının katman verisi üzerindeki etkisi

OSG kütüphanesi ile *shader* programları Şekil 4.17’de gösterildiği şekilde ayarlanabilir.

```
osg::Shader * vertexShader = new osg::Shader(osg::Shader::VERTEX, VertexShaderProgram);
osg::Shader * fragmentShader = new osg::Shader(osg::Shader::FRAGMENT, FragmentShaderProgram);

osg::Program * tShaderProg = new osg::Program;
tShaderProg->addShader(vshader);
tShaderProg->addShader(fshader);

mView->setProgram(tShaderProg);
```

Şekil 4.17. OSG shader programı ayarlanması

4.2. Çoklu Platform Desteği ve Yazılım Geliştirme Süreçleri

Tez kapsamında geliştirilen mimari, AG teknolojisinin zenginleştirilmiş görüntü üretimini ve kullanıcıya sunumunu sağlayan OpenGL tabanlı görüntü üreticisi (GÜ) tanımlamaktadır. GÜ, günümüzde RAHAT olarak elde edilebilen giyilebilir ve mobil cihazların tümünü kapsayabilmelidir. Bu amaç doğrultusunda, RAHAT ürünlerin büyük

bir çoğunluğu, işletim sistemi ve mevcut donanım kabiliyetleri açısından incelenmiştir. Çizelge 4.2’de inceleme sonucu yer almaktadır.

Çizelge 4.2. RAHAT akıllı gözlük cihazlarının yazılım ve donanım altyapıları

| Cihaz | OS | CPU | GPU | Geliştirici Platformu |
|------------------|--|-------------------------------|----------------------|-------------------------------------|
| ODG-R7 | Reticle OS (Android Marshmallow) | Qualcomm Snapdragon 805 | Adreno 420 | Android destekli tüm platformlar |
| Epson BT-350 | Moverio (Android 5.1) | Intel® Atom™ x5 | Intel Graphics | Android destekli tüm platformlar |
| Hololens 2 | Windows 10 32- bit | Intel Atom X5-Z8100 | HoloLens Graphics | Unity |
| Magic Leap | Lumin OS Custom Android | NVIDIA® Parker SOC | NVIDIA Pascal™ | Magic SDK, Unity, Unreal |
| Vuzix Blade | Android | ARM Cortex- A53 | SoC | Android destekli tüm platformlar |
| Google Glass EE2 | Android Oreo | Qualcomm Quad Core | SoC | Android SDK |

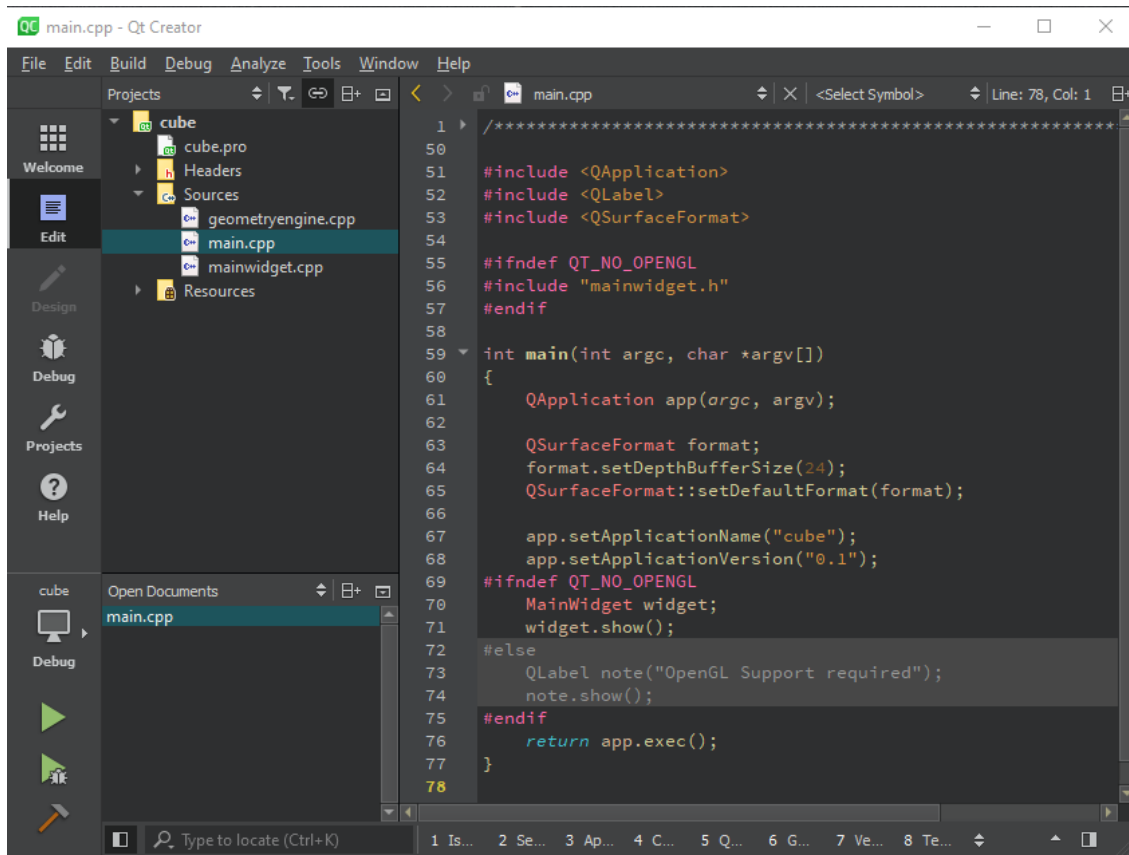
Çizelge 4.2’de aktarılan veriler, akıllı gözlük üreticilerinin farklı işletim sistemlerine, Intel ve ARM gibi farklı mimarilerdeki işlemcilere ve grafik işlemcilere ürünlerinde yer verdiklerini göstermektedir. Tez kapsamında yapılan çalışmanın birden çok platformu desteklemesine yönelik yazılım altyapısında belirli kriterler oluşturulmuş ve bu kriterler desteklenmiştir.

Windows, Android ve iOS gibi farklı işletim sistemlerini desteklemesi, Intel ve ARM gibi farklı işlemci mimarilerini desteklemesi, bölüm “3.3. Giyilebilir ve Mobil Donanım Platformları” başlığı altında incelenen donanım hızlandırıcıların farklı mimarilerde desteklenmesi başlıkları, çoklu platform desteği sağlanması için yazılım mimarisinde göz önüne alınan kriterler olmuştur.

Tez kapsamında, Windows ve Android işletim sistemleri üzerine analizler yapılmıştır. Bu işletim sistemleri altında yazılım geliştirmek için kullanılan kütüphaneler, 64-bit Intel işlemci mimarilerini masaüstü platformlar için, arm64-v8 ve x86 işlemci mimarilerini mobil platform için desteklemesi gerektiğine karar verilmiştir.

Temel yazılım çatısının inşa edilmesi için Qt Kütüphaneleri ve QtCreator adlı entegre geliştirme ortamı kullanılmıştır (IDE). Qt, Windows, Android, iOS, WinRT, Linux, Embedded Linux gibi çok sayıda işletim sistemine yönelik, tek bir ortamda yazılım geliştirmeye imkân sağlamaktadır. QtCreator ile, Windows, Linux ve Mac işletim sistemleri altında, 32 bit ve 64 bit işlemciler için Intel, ARM ve diğer bir çok mimariye yönelik geliştirme yapılabilir. Böylelikle geliştirici istediği herhangi bir

platformda, istediği herhangi bir hedef platform için uygulama üretebilmektedir (XU ve ark., 2006). Hedef platformun değişmesi durumunda, Qt kütüphaneleri kullanılarak yazılan kod parçalarında değişiklik yapılmasına gerek duyulmadan çalışan uygulama üretilmektedir. Qt, pencere yönetimi, grafik işleme, ağ haberleşmesi, giriş-çıkış donanımları işlemleri, dosya işlemleri, kullanıcı arayüzleri, veri kodlama ve çözümlenme, matematik işlemleri ve daha birçok kabiliyeti hazır olarak barındırmaktadır. Tez kapsamında yürütülen yazılım geliştirme sürecinde Qt kütüphanesi ve QtCreator IDE'si temel araç ve kütüphane olarak kullanılmıştır. Şekil 4.18'de QtCreator IDE'ye ait arayüz yer almaktadır.



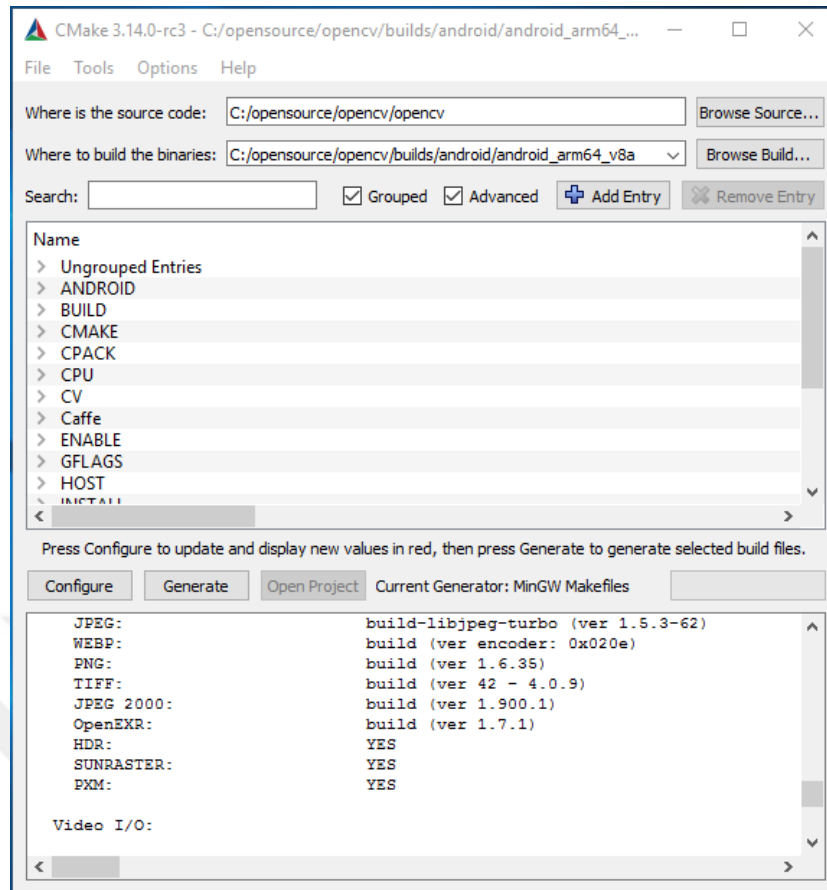
Şekil 4.18. QtCreator IDE arayüzü

OpenGL, grafik işlemlerin bilgisayar tarafından çizilmesini sağlamak için kullanılan, yazılım geliştiriciye grafik donanımını yönetme imkânı sunan açık kaynak yazılım kütüphanesidir. OpenGL, grafik çizim işlemlerini standartlaştırmıştır. İşletim sisteminden ve işletim sisteminin üzerinde çalıştığı platformdan bağımsızdır. Bu sayede, taşınabilir yazılım geliştirmeye imkân veren kütüphane, tüm grafik kartlarıyla uyumlu çalışmakta ve yazılım geliştiriciyi donanım farklılıklarından soyutlamaktadır. Buna ek

olarak, OpenGL, işletim sistemi altında çalışan pencere yöneticisinden bağımsızdır. Bu özelliği sayesinde, Windows, Mac, Linux, Android ve diğer işletim sistemlerinde sorunsuz bir şekilde çalışabilmektedir. Böylelikle, geliştirilen yazılımın çoklu platform desteği, grafik işlemleri açısından sağlanabilmektedir. Çalışma kapsamında, OpenGL API üzerine inşa edilmiş, grafik işleme fonksiyonlarını daha üst seviyede sunarak yazılım geliştiricilere kolaylıklar sağlayan, açık kaynak grafik işleme kütüphanesi olan OSG kütüphanesi kullanılmaktadır. OSG, Intel ve ARM işlemcileri desteklemekle birlikte, IPP, NEON, HFP gibi donanım hızlandırıcı tüm devre özelliklerini desteklemektedir.

OpenCV kütüphanesi, görüntü işleme uygulamaları için temel bir araç olarak kabul edilmiş ve başarılı birçok uygulamada kullanılmış görüntü işleme kütüphanesidir. Üstün grafik işleme kabiliyeti desteği ile, yüksek performanslı algoritma tasarımlarına imkân vermektedir (Baggio, 2012). OpenCV, Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD, Android, iOS, Maemo, BlackBerry 10, işletim sistemlerini desteklemektedir. OpenCV, CUDA, OpenCL, IPP, ARM NEON, HFP gibi platformları ve donanımsal hızlandırıcıları desteklemektedir.

Tez çalışması kapsamında kullanılan kütüphaneler, yazılım geliştirme sürecine başlamadan önce kullanıma hazır hale getirilmişlerdir. Qt kütüphanesine ait derlenmiş hazır dosya yapısı, www.qt.io internet sitesinde resmi olarak dağıtılan açık kaynak sürüm adresinden, Masaüstü Windows msvc2017 derleyicisi için, Android arm64-v8 ile x86 sürümleri MinGW derleyicisi için Qt 5.12.4 versiyonu indirilmiştir. OSG (<http://www.openscenegraph.org/>) ve OpenCV (<https://opencv.org/>) kütüphanelerine ait Github depoları altında yer alan güncel kaynak kodları indirilmiştir. Bu kütüphaneler, üzerinde çalışılacak işlemci mimarileri ve işletim sistemleri için derlenmişlerdir. CMake-Gui uygulaması ile kaynak kodların konfigürasyonu her bir mimari ve platform için yapılmış, masaüstü platform için Visual Studio ile birlikte Microsoft tarafından dağıtılan MSVC 2017 derleyicisi, mobil platform için Qt ile birlikte dağıtılan MinGW 7.3 araç seti ile derlenmiştir. Şekil 4.19'da Cmake-Gui arayüzü yer almaktadır.



Şekil 4.19. CMake-Gui arayüzü

Tez kapsamında kullanılan bütün açık kaynak kütüphaneler ve geliştirilen tüm yazılım bileşenleri C++ dili ile kodlanmıştır. Bu sayede C++ dilinin modüler ve nesnel yapısının sağladığı kolaylıklardan ve dilin gücünü aldığı yüksek performans kabiliyetinden faydalanılmıştır.

5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Çalışma sonucunda geliştirilmiş olan GÜ yazılımı, AG uygulamaları tarafından üretilen sentetik dünya görselini, gerçek dünya görseli ile birleştirerek ekran ünitesi aracılığıyla kullanıcıya aktaran yazılım bileşenidir. AG uygulama mimarisinde ifade edilecek olursa, Şekil 4.2’de yer alan “Katman Yöneticisi” bileşenini kapsamaktadır. Geliştirilen uygulama için tasarlanan test prosedürleri, bu bileşeni sınamaya yönelik hazırlanmıştır. Sınama, iki aşamada ele alınmıştır. İlki, tasarlanan uygulamanın, üzerinde çalıştığı sistemin kaynaklarını tüketme düzeyini incelemiş, ikinci aşama ise, 3B sahne yönetiminin mobil cihaz üzerinde olduğu Şekil 4.1’ de yer alan mimari ile performans ve kaynak tüketimi açısından karşılaştırmasını ele almıştır. Tasarlanan mimari yapı, Çizelge 5.1’de listelenen cihazlar üzerinde test edilmiştir.

Çizelge 5.1. test platformlarına ait konfigürasyonlar

| Test Cihazı | OS | CPU | RAM | GPU |
|---------------------|---------------------------------------|---------------------------------|-------|-----------------------|
| Masaüstü Bilgisayar | 64 bit Windows 10 | Intel Core i7-7700 | 16 GB | Intel HD Graphics 630 |
| Akıllı Telefon | 64 bit Android 5.1 | Exynos 7420 AP | 3 GB | Mali™-T760 MP8 |
| Akıllı Gözlük | 32 Bit Android 5.1 temelli Moverio OS | Intel® Atom™ x5-Z8350 Quad Core | 2 GB | Intel Graphics |

5.1. Kaynak Tüketiminin Analizi

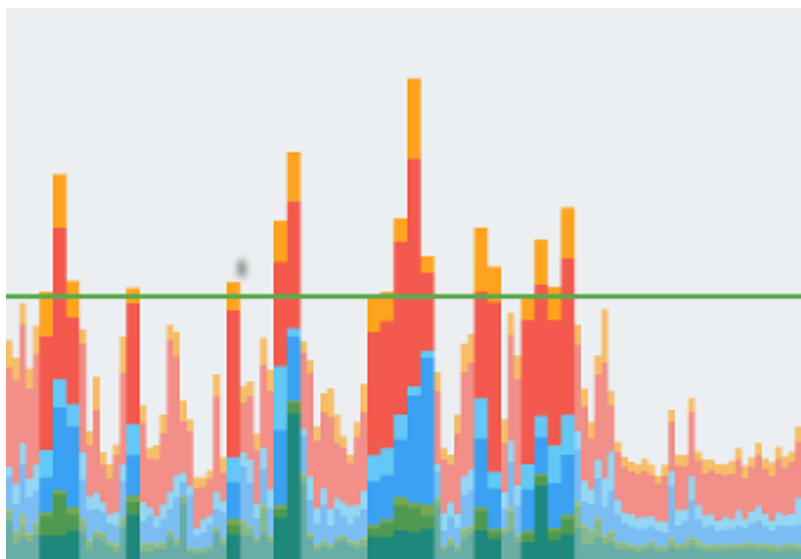
Tasarlanan test ortamı, farklı çözünürlük seviyelerindeki görüntü üretimi işlemlerinin, sistem üzerindeki kaynak tüketimlerini incelemektedir. Geliştirilen yazılım bileşenlerinin, üzerinde çalıştığı sisteme ait işlemci, bellek ve grafik işlemci üzerinde oluşturduğu yükleri analiz etmek amacıyla iki ayrı test icra edilmiştir. İlk testte, geliştirilen yazılım, temel yazılım bileşenlerinin devrede olduğu ancak grafik işlemlerin gerçekleştirilmediği formda kullanılmıştır. İkinci testte ise, temel bir AG uygulaması için 2 katmanlı (gerçek ve sentetik) görüntü işlemlerinin gerçekleştirildiği formda kullanılmıştır. Yapılan testler Çizelge 5.1’de yer alan cihazlar üzerinde gerçekleştirilmiştir. Çizelge 5.2’de, yazılımın her bir cihaz için oluşturulan çalıştırılabilir kopyasına ait mimari bilgileri yer almaktadır. Windows platformlar üzerindeki kaynak kullanımı analizi, Windows işletim sistemine ait bir uygulama olan Görev Yöneticisi (*Task Manager*) aracılığıyla, Android platformlar üzerindeki kaynak

kullanım analizi ise Android SDK ile birlikte dağıtılan Android Cihaz İzleyici (*Android Device Monitor*) uygulaması aracılığıyla gerçekleştirilmiştir.

Çizelge 1.2. Üretilen yazılımların çalışacağı hedef mimari bilgileri

| Test Cihazı | Derleyici | İşlemci Mimarisi | OpenGL Mimarisi | Pencere Yöneticisi |
|--------------------------|-----------------|------------------|-----------------|--------------------|
| Masaüstü Bilgisayar (PC) | MSVC2017 | 64-bit İntel | OpenGL 4.5 | Windows Explorer |
| Akıllı Telefon | MinGW 7.3 Tools | 64-bit arm64-v8 | OpenGL ES 2.0 | Android WM |
| Akıllı Gözlük | MinGW 7.3 Tools | 32-bit x86 | OpenGL ES 2.0 | Android WM |

Android işletim sistemlerinde, işlemci ve bellek kaynakları gerçek zamanlı olarak izlenebilmekte ancak grafik işlemlerin kaynak tüketimi izlenememektedir. Bu sorunu aşmak amacıyla yapılan çalışmalar, cihazlar üzerinde çalıştırılan test uygulamasına ait ölçülebilir GPU verilerinin alınmasını sağlayamamıştır. Denenen ilk ölçüm yöntemi, mobil cihazlar üzerinde çalışan, GPU verilerini, Şekil 5.1’de görüldüğü üzere cihaz ekranına ölçeksiz ve değersiz olarak çubuk grafik şeklinde işleyen yöntemdir. İşletim sistemine entegre çalışan bu yöntem ile yaşanan ilk sorun, grafik işlem yapan uygulamalar ile aynı anda çalışmıyor olmasıdır. Yani, cihaz ekranında OpenGL çizim sahnesi yer alıyor iken bu uygulama ekranda görülmemektedir. İkinci sorun ise, bu uygulamanın grafik işlemlere ait işlem sürelerini renkli çubuklar şeklinde sunuyor olmasıdır. Sunulan bu grafik işlemlerin büyük bir kısmı paralel işlemler olduğu halde seri işlemler mantığında görselleştirilmiş olduğu ve herhangi bir metrik veri sunmadığı için performans açısından karşılaştırma yapmaya uygun bulunmamıştır.



Şekil 5.1. Android Cihazlarda bulunan grafik işlemleri grafiği

İkinci ölçüm yöntemi ise, mobil cihazların çalışma sırasında bilgisayara USB kablo ile bağlanması sonucu *Android Debugger Bridge* adı verilen yazılım aracı ile birtakım işlemlerin yapılmasına imkân tanıyan uygulamanın kullanıldığı yöntemdir. Bu uygulama, terminal komutları aracılığı ile *unix* temelli işlemlerin yapılmasına izin vermektedir. GPU kaynaklarının kullanılmasına yönelik olarak, cihazın bağlı olduğu geliştirici bilgisayarı terminalinde “*adb shell dumpsys gfxinfo > D:/GPU_verileri.txt*” komutu çalıştırılmış, bilgisayar üzerinde belirtilen GPU_verileri.txt dosyasına bir takım veriler kaydedilmesi sağlanmıştır. Bu veriler incelendiğinde, Android OS platformuna özgü Java altyapısı ile geliştirilen, Dalvik Java VM üzerinde çalışan uygulamaların grafik işlemci üzerinde tükettikleri kaynakları sağlıklı bir şekilde okuduğu ancak tez kapsamında geliştirilen uygulama yapısında olduğu gibi tamamen Java VM altyapısı ile çalıştırılmayan, daha yüksek performans sunan yönetimsiz (*unmanaged*) olarak nitelendirilen C++ dili ile geliştirilmiş uygulamanın kaynaklarına erişemediği gözlemlenmiştir. Bu nedenle Android işletim sistemi altında Grafik İşlemci kaynak tüketimi incelenememiş ancak referans kabul edilebilecek olan Windows işletim sistemi kaynaklarına erişilebilmiş ve gerekli veriler toplanabilmiştir.

5.1.1. Temel Kaynak Tüketiminin Analiz Verileri

Çizelge 5.3’de görüntü üretici yazılımının Çizelge 5.1’de yer alan cihazlar üzerinde herhangi bir grafik işlemi çalışmıyorken, sistem kaynaklarını harcama verileri yer almaktadır.

Çizelge 5.3. Görüntü üreticinin temel kaynak tüketimi verileri

| Test Cihazı | İşlemci | Grafik İ. | RAM |
|-------------|---------|-----------|----------|
| 640x480 | | | |
| PC | 0% | 0% | 29.4 Mb |
| Telefon | 0% | - | 103.7 Mb |
| Gözlük | 0% | - | 105.0 Mb |
| 1280x720 | | | |
| PC | 0% | 0% | 41.7 Mb |
| Telefon | 0% | - | 109.7 Mb |
| Gözlük | 0% | - | 110.1 Mb |
| 1920x1080 | | | |
| PC | 0% | 0% | 82.2 Mb |
| Telefon | 0% | - | 117.5 Mb |
| Gözlük | 0% | - | 116.8 Mb |

5.1.2. AG Katmanlarının Farklı Çözünürlükler İçin Performans Analiz Verileri

Çizelge 5.4’de görüntü üreticinin 2 katman mimarisinde çalışıyor iken, farklı çözünürlükler altında harcadığı sistem kaynağı verileri yer almaktadır.

Çizelge 5.4. 2 katmanlı görüntü üretici çalışmasına ait kaynak tüketimi verileri

| Test Cihazı | İşlemci | Grafik İ. | RAM |
|-------------|-----------|-----------|----------|
| | 640x480 | | |
| PC | 0.3% | 6.3% | 41.1 Mb |
| Telefon | 7% | - | 172.7 Mb |
| Gözlük | 8% | - | 167.6 Mb |
| | 1280x720 | | |
| PC | 0.5% | 12.9% | 70.9 Mb |
| Telefon | 12.0% | - | 180.1 Mb |
| Gözlük | 11.0% | - | 183.1 Mb |
| | 1920x1080 | | |
| PC | 1.1% | 23.4% | 98.3 Mb |
| Telefon | 16% | - | 197.8 Mb |
| Gözlük | 24% | - | 202.2 Mb |

5.2. Görüntü Üretiminin Analizi

Görüntü üretimi analizi, normal bir AG uygulamasının sunduğu görüntü kalitesi ve niteliğinin, tez kapsamında geliştirilen GÜ’nin ürettiği görsel çıktı ile karşılaştırılmasına dayanmaktadır. Bu amaçla, Şekil 4.1’de yer alan mimariye sahip temel bir AG uygulaması ile birlikte, tez kapsamında geliştirilen Şekil 4.2’de mimarisi yer alan GÜ uygulaması, Çizelge 5.1’de yer alan cihazlar üzerinde çalıştırılmıştır. Her iki yazılım da OSG kütüphanesi kullanılarak geliştirilen grafik işleme altyapısına sahiptir.

AG uygulamasının gerçek dünya görselini ifade eden katmanı için, performans verilerinde sadece grafik işlemci verilerinin yer almasını sağlayarak sağlıklı sonuçlar alınması amacıyla, cihaz üzerinde yer alan kamera verisi ile işlenmemiş, Ülkemizde 2018 yılında Seferihisar/İZMİR bölgesinde gerçekleştirilen EFES2018 tatbikatına ait görüntüler kullanılarak bellekten yükleme yöntemi ile oluşturulmuştur. Böylelikle, kamera yakalama işlemlerinin CPU, bellek ve GPU üzerinde oluşturacağı yükler elenmiştir. Bilgisayar üretimi sentetik dünya verileri için, OSG kütüphanesinin açık kaynak olarak yayınladığı OpenSceneGraph-Data (<https://github.com/openscenegraph/>)

[OpenSceneGraph-Data](#)) veri kaynağında yer alan 3B modeller kullanılmıştır. Bu kaynakta, araçlar, avatarlar, geometrik şekiller, doku kaplamaları, çeşitli animasyonlar ve örnek *shader* program parçaları yer almaktadır. Test uygulamasında AG zenginleştirme verisi olarak, tatbikat görsellerine uygun olması amacıyla, askeri hava ve kara aracı modelleri kullanılmıştır. Çizelge 5.5’de modellere ait bilgiler yer almaktadır. *Vertex* sayısı 3B modelin karmaşıklığını ve işleme güçlüğüne etkilemektedir.

Çizelge 5.5. Test kapsamında kullanılan modellere ait veriler

| Model Dosyası | Vertex Sayısı | Dosya Boyutu |
|----------------|---------------|--------------|
| cessna.osg | 9.680 | 741 Kb |
| dumptruck.osg | 26.438 | 1866 Kb |
| spaceship.osgt | 2.755 | 393 Kb |

5.2.1. AG Uygulaması ve GÜ Uygulamasının Karşılaştırma Verileri

Test cihazları üzerinde çalıştırılan AG Uygulaması ve GÜ uygulamasında, zenginleştirme içeriği için 3 adet 3B model kullanılmıştır. Bu analiz sonucunda elde edilen kaynak tüketimi verileri Çizelge 5.6’da yer almaktadır. Görüntü Üretici yazılımının katman verileri için bellekten yüklenen görseller, AG uygulaması tarafından üretilen OpenGL sahnesine ait verilerin kaydedilmesi ile elde edilmiştir. Böylelikle, AG uygulaması, Şekil 4.2’de yer alan “İşlem Ünitesi” gibi davranmış ve alınan görseller, görüntü üretici uygulamasına transfer edilmiş veriler olarak kabul edilmiştir. Şekil 5.2’de katman görselleri yer almaktadır.

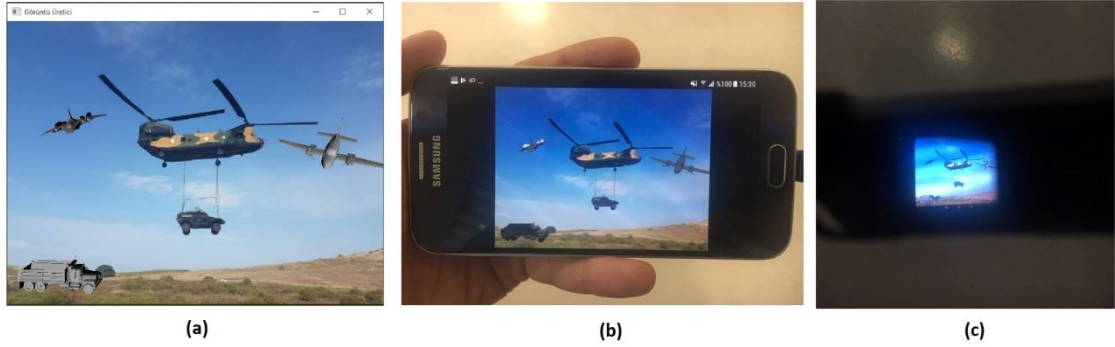


Şekil 5.2. Katman verilerine ait görseller

Çizelge 5.6. 3 adet 3B model içeren AG ve GÜ uygulamasına ait kaynak tüketim verileri

| Yazılım | İşlemci | Grafik İ. | RAM |
|---------------------|---------|-----------|----------|
| Masaüstü Bilgisayar | | | |
| AG Uygulaması | 0.9% | 5.0% | 51.4 Mb |
| Görüntü Üretici | 0.3% | 6.3% | 41.1 Mb |
| Cep Telefonu | | | |
| AG Uygulaması | 17.7% | - | 200.4 Mb |
| Görüntü Üretici | 7.2% | - | 172.7 Mb |
| Akıllı Gözlük | | | |
| AG Uygulaması | 46.5% | - | 208.5 Mb |
| Görüntü Üretici | 8.8% | - | 167.6 Mb |

Şekil 5.3’de test sırasında cihazlardan alınan görüntüler yer almaktadır.



Şekil 5.3. Üç adet 3B model içeren test aşamasına ait görseller. (a) masaüstü bilgisayar, (b) cep telefonu, (c) akıllı gözlük ekranından alınan yansıma görselidir.

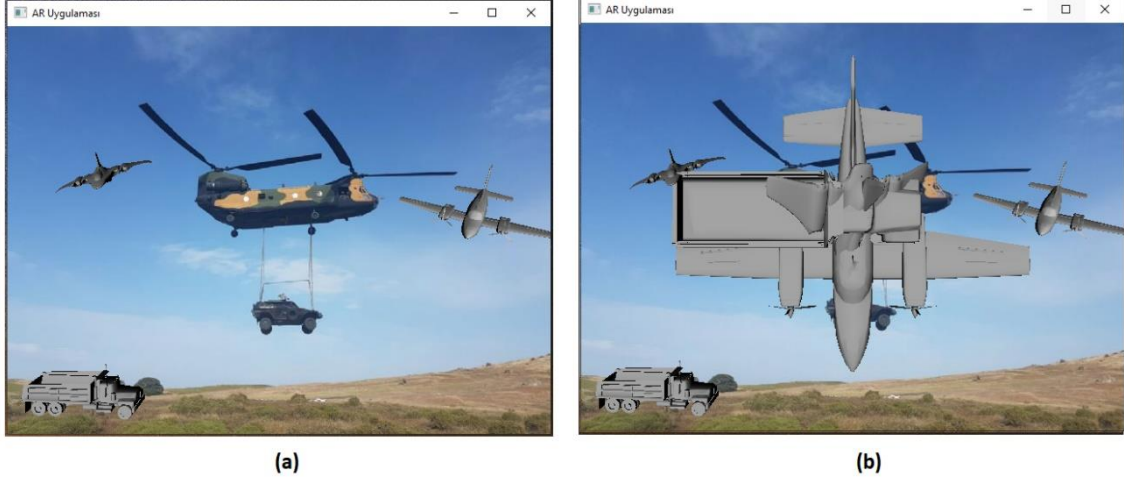
Testin ikinci aşamasında, grafik işlemci üzerinde yük oluşturmak amacıyla, AG içerik verisini oluşturan 3B modellerin sayısı 3’ten 12’ye çıkartılmıştır. Bu durumdaki kaynak tüketimleri Çizelge 5.7’de yer almaktadır.

Çizelge 5.7. 12 adet 3B model içeren AG ve GÜ uygulamalarına ait kaynak tüketim verileri

| Yazılım | İşlemci | Grafik İ. | RAM |
|---------------------|---------|-----------|----------|
| Masaüstü Bilgisayar | | | |
| AG Uygulaması | 3.4% | 14.5% | 65.4 Mb |
| Görüntü Üretici | 0.3% | 6.2% | 41.1 Mb |
| Cep Telefonu | | | |
| AG Uygulaması | 21% | - | 335.8 Mb |
| Görüntü Üretici | 7.3% | - | 172.7 Mb |
| Akıllı Gözlük | | | |
| AG Uygulaması | 47% | - | 329.1 Mb |

| | | | |
|-----------------|------|---|----------|
| Görüntü Üretici | 8.6% | - | 167.6 Mb |
|-----------------|------|---|----------|

Şekil 5.4'de test yazılımı tarafından üretilen birleştirilmiş AG görseli yer almaktadır. Şekil 5.4.a, 3 adet model zenginleştirilmiş, Şekil 5.4.b ise yük testi için 12 adet model ile zenginleştirilmiş görseli ifade etmektedir. Eklenen 9 adet model, konum ve bakış açısı sahne sıfırına ayarlanmış olarak yüklenmişlerdir.



Şekil 5.4. (a) 3 adet model zenginleştirilmiş görsel (b) yük testi için 12 adet model ile zenginleştirilmiş görsel

5.2.2. Görünüş Oranı Uyumluluğunun Analizi

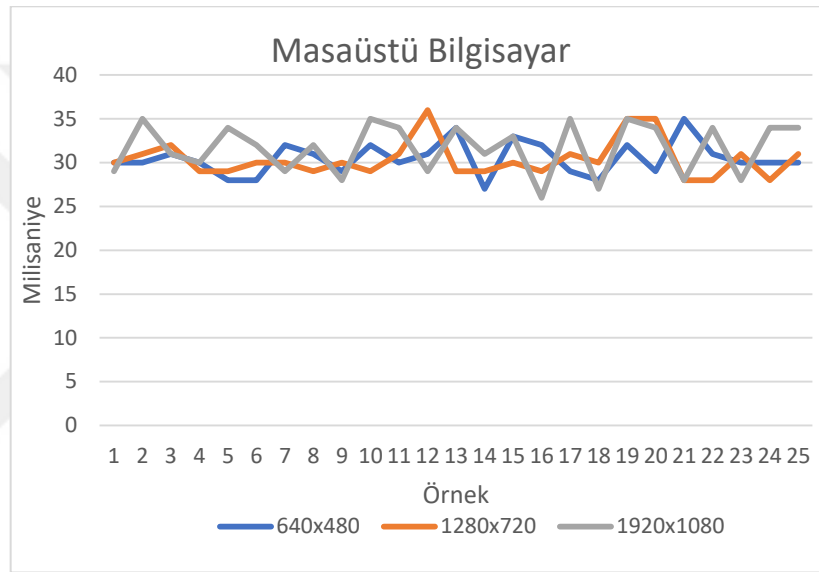
Bölüm 4.1.1.1. “Pencere Yöneticisi” başlığı altında açıklanan görünüş oranı uyumluluğu için, uygulama penceresinin farklı boyutlarda çizilmesi durumuna karşılık katman yöneticisinin davranışı incelenmiştir. Şekil 5.5’de yer alan görselde farklı en-boy oranlarında katman çizimlerinin görünüşü yer almaktadır.



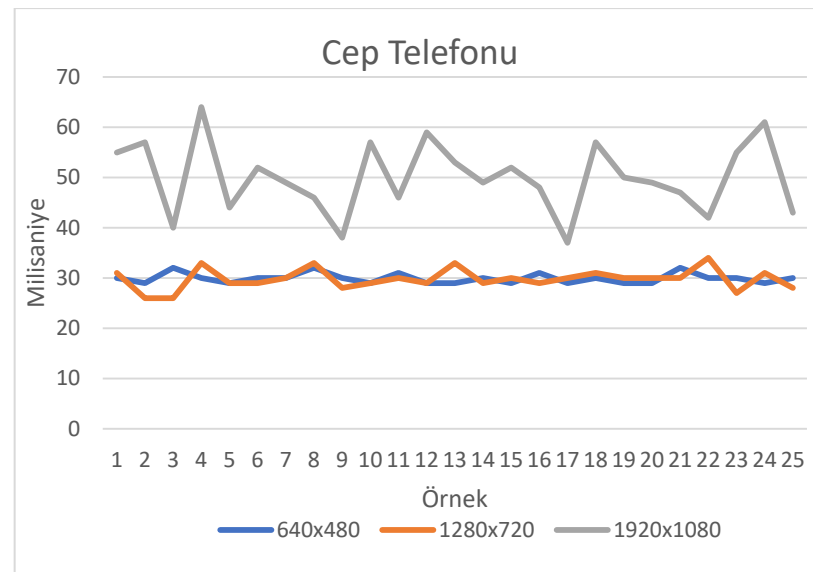
Şekil 5.5. Görünüş oranı uyumluluğu algoritmasının farklı pencere boyutlarında oluşturduğu görseller. (a) birebir boyutlar, (b) yatay daraltma, (c) dikey daraltma

5.2.3. Ekran Yenileme Performansının Analizi

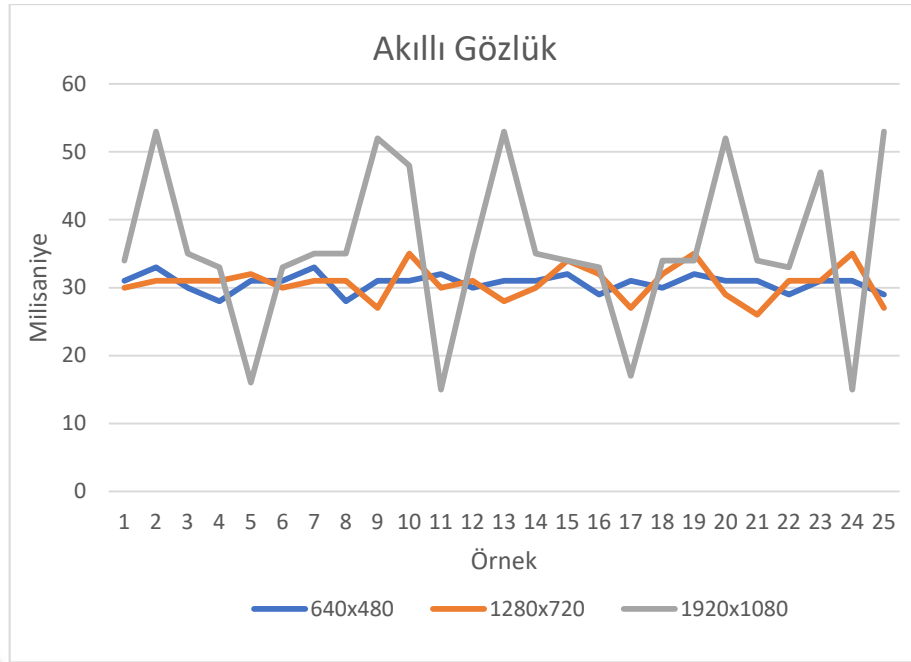
Görüntü üreticinin ekran yenileme performansının sınanması amacıyla, OpenGL sahnesinin her bir görüntü karesi için yenilenme zamanlarının dökümü çıkartılmıştır. Yazılım, 30 milisaniye çevrim süresinin hedeflendiği yenileme periyodunda çalışırken, her bir görüntü karesinin çizim zamanı 25 örnek için toplanmış ve Çizelge 5.1’de yer alan her bir cihaz için Şekil 5.6, Şekil 5.7 ve Şekil 5.8 üzerinde görselleştirilmiştir. Her bir grafik alanı, 640x480, 1280x720 ve 1920x1080 çözünürlükleri için toplanan verilerin grafik çizimlerini içermektedir.



Şekil 5.6.3 Masaüstü bilgisayar üzerinde toplanan ekran yenileme zamanları



Şekil 5.7. Cep telefonu üzerinde toplanan ekran yenileme zamanları



Şekil 5.8. 4 Akıllı gözlük üzerinde toplanan ekran yenileme zamanları

6. SONUÇLAR VE ÖNERİLER

6.1 Sonuçlar

AG uygulamalarının, üzerinde çalıştığı mobil cihazlara ait sistem kaynaklarını daha düşük seviyelerde kullanmasını sağlamak amacıyla yapılan tez çalışması kapsamında, geliştirilen GÜ uygulamasının sistem kaynakları üzerinde oluşturduğu yükler incelenmiş ve temel bir AG uygulaması ile karşılaştırmalı test verileri toplanmıştır.

Geliştirilen GÜ yazılımının, üzerinde çalıştığı sisteme ait temel kaynakları kullanımına yönelik gerçekleştirilen test işleminin sonuçlarının yer aldığı Çizelge 5.3 incelendiğinde, yazılımın temel kaynak tüketiminin mevcut sistem kaynaklarına oranla oldukça düşük seviyede olduğu görülmektedir. Bu değerler, görüntü üretimi işleminin yapılabilmesi için, geliştirilen yazılımın içerdiği diğer tüm işlemlerin, sistem üzerinde oluşturduğu yükleri ifade etmektedir. İşlemci ve grafik işlemci kaynakları, yazılımın bu formundan beklendiği gibi hiç tüketilmemektedir. Bu veriler, yazılım altyapısında kullanılan kütüphanelerin ve yöntemlerin uygunluğunun kanıtı niteliğindedir. Bellek tüketimi açısından, RAM bellek üzerinde oluşturulan *stack* ve *heap* alanlarının toplamını gösteren veriler, sistemler üzerinde yer alan toplam bellek kaynaklarına oranla oldukça düşük seviyelerde kalmaktadır. Çözünürlük artışına karşın bellek üzerinde gerçekleşen 0.2% - 0.3% oranındaki artışlar, katman verisinin depolanması için oluşturulan tampon bellek alanlarının büyümesi nedeniyle meydana gelmektedir. 640x480 çözünürlükteki her bir görüntü karesi için toplam tampon bellek alanı 1.228.800 bayt tutarken, 1280x720 için 3.686.400 ve 1920x1080 için 8.294.400 bayt olmaktadır. Bu tampon alanlar, grafik işlemci üzerinde çalışan işlemler aktifleştirildiğinde, OpenGL işlemlerinin gerçekleştirilmesi için grafik tüm devresine kopyalanmak amacıyla RAM bellekte tutulmaktadır. Her üretilen yeni görüntü karesi bu şekilde işlenmektedir.

İki katmanlı formda çalışan GÜ üzerinde yapılan analizlere ait sonuçları içeren Çizelge 5.4 incelendiğinde, GÜ'nün tüm grafik işlemleri aktif iken, işlemci ve grafik işlemci kullanımının gerçekleştiği görülmektedir. İşlemci kullanımı oranı, sistemin sahip olduğu işlemci gücüne bağlı olmakla birlikte, en yüksek çözünürlük için (1920.1080) masaüstü bilgisayar platformunda, 1.0%, cep telefonu üzerinde 16% ve akıllı gözlükte ise 24% olarak ölçülmektedir. Masaüstü bilgisayar için ölçülebilen grafik

işlemci kullanımları incelendiğinde, en düşük çözünürlükte 6.3% olan kaynak tüketimi en yüksek çözünürlükte 23.4% seviyesine çıkmaktadır. Bellek kullanımı seviyelerindeki artış, yine beklendiği gibi, grafik işlemlerin yapıldığı her görüntü karesinin, bellekler arası transfer işlemleri sırasında oluşturulan tampon alan kopyalarının anlık olarak oluşturulması ve yok edilmesi nedeniyle ortalama değerde yükselme meydana getirmesidir. Sonuç olarak, bellek kullanımı en yüksek çözünürlük altında, masaüstü bilgisayarda 0.61%, cep telefonu üzerinde 6.59%, akıllı gözlük üzerinde ise 10.11% düzeylerinde tüketilmektedir. Temel kaynak tüketimi verileri ile karşılaştırıldığında, GÜ yazılımı en yüksek grafik işlemleri seviyesinde iken mobil cihazlar üzerinde ortalama olarak 2 kat daha fazla kaynak tüketimi gerçekleştirmektedirler. Masaüstü platformun sahip olduğu kaynaklar mobil cihazlara oranla çok yüksek olduğu için artış miktarı 0.1% değerinde kalmaktadır.

Bu çalışmadan beklenen sonuç, yukarıda bahsi geçen kaynak tüketimi verilerinin, içerik karmaşası ne düzeyde olursa olsun, her tür AG uygulaması için sabit kalmasıdır. Bu nedenle gerçekleştirilen ikinci aşama testlerde, aynı yazılım teknolojisini kullanarak geliştirilen ve tamamı mobil cihaz üzerinde çalışan temel bir AG uygulaması ile alınan veriler, tez kapsamında geliştirilen GÜ yazılımına ait veriler ile karşılaştırılmıştır. Mobil cihazların kaynak limitlerine ulaşılması sonucu hatalı veriler ile analiz yapılmasına engel olmak için, ikinci aşama testlerde 640x480 çözünürlüğüne sahip görüntü verileri ile sına yapılmıştır. Çizelge 5.6'da yer alan veriler, her bir platformda ayrı ayrı çalıştırılan AG uygulaması ve GÜ uygulamasının kaynak tüketimini içermektedir. Her iki yazılımın da yaptığı işlem sonucu, Şekil 5.3'de yer alan son görsel veri kullanıcıya sunulmuştur. Çizelge 5.6'da görüldüğü üzere, aynı görsel çıktı üretilirken, AG uygulamasının masaüstü platformda 0.9% işlemci tüketimine karşılık, GÜ yazılımı 0.3% seviyesinde kalmıştır. Grafik işlemci kullanımı AG uygulamasında 5.0% iken GÜ uygulamasında 6.3% seviyesi ile daha yüksek çıkmıştır. Bellek kullanımı ise, AG uygulamasında 51.4 Mb iken GÜ uygulamasında ise 41.1 Mb seviyesindedir. Mobil cihazlarda ise kaynak tüketimi verileri daha çarpıcı sonuçlar doğurmuştur. Cep telefonundaki işlemci kullanımı AG uygulamasında 17.7% iken GÜ uygulamasında 7.2% seviyesindedir. Bellek kullanımı ise AG uygulamasında 200.04 Mb iken GÜ uygulamasında 172.7 Mb seviyesinde kalmıştır. Sahip olduğu kaynaklar açısından en düşük kapasiteye sahip cihaz olan akıllı gözlük üzerinde alınan veriler ise tez çalışmasının önerdiği kaynak tüketiminin düşürülmesi hedefinin gerçekleştiğini gözler önüne sermektedir. AG uygulamasında işlemci tüketimi 46.5% iken GÜ

uygulamasında 8.8% ile yazılım performansında 5 kattan daha büyük bir fark oluşturmuştur. Bellek kullanımı açısından ise AG uygulaması 208.5 Mb iken GÜ uygulaması 167.6 Mb seviyesinde kalmıştır. Üç adet 3B modelin yer aldığı bu test çalışması sonucunda görülmüştür ki GÜ yazılımı, aynı görsel veriyi üreten AG yazılımına oranla cep telefonunda işlemci açısından 200% 'ün üzerinde, akıllı gözlük üzerinde ise 500%'ün üzerinde performans artışı sağlamıştır.

Tez çalışmasının önerdiği bir diğer kriter ise, AG uygulama içeriği her ne seviyede karmaşıklığa sahip olursa olsun, belirli ve sabit bir kaynak tüketimi performansı sunulmasıdır. Bu kriterin sınanmasına yönelik yapılan test çalışmasında AG içeriğinde kullanılan model sayısı 3'ten 12'ye çıkartılmış ve her iki uygulama için toplanan veriler Çizelge 5.7'de sunulmuştur. Bu çalışmadaki en çarpıcı veri, grafik işlemcinin kullanımını gösteren masaüstü bilgisayarda çalışan AG ve GÜ uygulamalarına ait verilerdir. Model sayısının artması ile, grafik işlemcinin çalışma oranı bir önceki test ortamına göre 5.0% değerinden 14.5% değerine çıkmış, 300% e yakın bir yükselme göstermiştir. Buna karşılık, görüntü üretici yazılımının grafik işlemci kullanımı 6.2% seviyesinde kalarak bir önceki 6.3% seviyesi ile aynı performansı sergilemiştir. İşlemci kullanımı açısından bakıldığında, AG yazılımı 3.4% seviyesine çıkarırken, GÜ yazılımı 0.3% değerinde kalarak yaklaşık 10 kat daha düşük kaynak tüketimi ile aynı görsel veriyi üretebilmiştir. Görüntü üretici yazılımının bellek kullanımı 41.1 Mb ile aynı seviyede kalırken, test modellerinin sayındaki artış nedeniyle yüklenen veri miktarı arttığı için AG uygulamasının bellek kullanımı önceki test ortamına göre 51.4 Mb değerinden 65.4 Mb değerine yükselmiştir.

Aynı testin mobil cihazlar üzerindeki uygulaması da masaüstü platform ile aynı sonucu göstermektedir. İşlemci kullanımı açısından, cep telefonunda çalışan AG uygulaması 21.0% seviyesinde işlemci tüketirken 7.3% değeri ile GÜ yazılımının işlemci kullanımı değişmemektedir. AG uygulamasında bellek kullanımı ise model dosyalarının artışı sonucu oldukça yükselerek 335.8 Mb 'a yükselirken görüntü üretici uygulamasında 172.7 Mb seviyesinde kalmıştır. Aynı şekilde, akıllı gözlük üzerinde çalışan GÜ yazılımının kaynak tüketimleri aynı seviyede kalırken, AG uygulamasının işlemci açısından kaynak tüketimi, GÜ yazılımına göre 600% daha fazla iken, bellek kullanımı 2 kata yakın fark oluşturmuştur.

Sonuç olarak, tez çalışması kapsamında geliştirilen GÜ yazılımı, AG içeriğinin karmaşıklığından bağımsız olarak düşük ve sabit bir düzeyde kaynak tüketimi sergileyerek önerilen performans kriterlerini sağlamıştır. Çizelge 5.6'da masaüstü

bilgisayar üzerinde yapılan karşılaştırmada, grafik işlemci kullanımı, görüntü üretici yazılımı için, AG uygulamasına göre %26 daha fazla çıkmıştır (AG, 5.0% GÜ, 6.3%). Çizelge 5.7'deki sonrasında yapılan yük testine ait verilerde görülmektedir ki, AG uygulamasının grafik işlemci kullanımı 14.5% seviyesine çıkarken, GÜ yazılımının grafik işlemci kullanımı 6.2% değerinde kalmıştır. Çizelge 6.1 de yer alan verilere göre AG uygulaması, GÜ yazılımına göre 133.8% daha yüksek grafik işlem kaynağı tüketmiştir. Böylelikle, içerik verilerinin karmaşıklığından soyutlanmış bir şekilde, AG uygulamalarında kullanılabilen bir GÜ yazılımı başarılı bir şekilde gerçekleştirilebildiği görülmektedir.

Çizelge 6.1. GÜ yazılımının AG yazılımına kıyasla kaynak tüketimleri üzerindeki kazanç yüzdeleri

| İşlemci | Grafik İ. | RAM |
|---------------------|-----------|--------|
| Masaüstü Bilgisayar | | |
| 1033.33% | 133.87% | 59.12% |
| Cep Telefonu | | |
| 187.67% | - | 94.44% |
| Akıllı Gözlük | | |
| 446,5% | - | 96.36% |

GÜ yazılımının sınındığı bir diğer kriter olan görünüş oranının korunmasına yönelik yapılan testlerde alınan sonuçlar Şekil 5.5'de yer almaktadır. Şekil 5.5.a'da, 640x480 çözünürlüğe sahip 1.33 en-boy oranında bir pencere açılmış, daha sonrasında bu pencere yatayda daraltılarak 380x480 ebatlarına getirilmiştir. Bunun sonucunda Şekil 5.5.b'de görüldüğü üzere, pencere yöneticisi otomatik olarak, görsel verinin görünüş oranını koruyarak, pencerenin alt ve üst kısımlarında karanlık bölgeler oluşturmuş ve grafik işleme alanını 380x285 piksel ebatlarına indirmiştir. Böylelikle, 1.33 olan görünüş oranı korunmuştur. Aynı şekilde Şekil 5.5.c 'de görüldüğü gibi, dikeyde yapılan daraltma işlemi ile yatayda karanlık bölgeler oluşmuş ve görünüş oranı başarılı bir şekilde korunmuştur.

GÜ yazılımının genel bir performans değerlendirmesi açısından, ekran yenileme süreleri üzerine yapılan son test çalışması Bölüm 5.2.3. de yer almaktadır. Şekil 5.6'da yer alan masaüstü bilgisayar üzerinde toplanan veriler, çözünürlük farklılığına karşılık, görüntü yenileme zamanlarında +-5 milisaniyelik bir değişim ile yaklaşık aynı performansın sergilendiğini göstermektedir. Çözünürlük arttıkça, zaman verilerinde düzensizliğin arttığı grafik üzerinden analiz edilebilmektedir. Toplanan veriler genel olarak beklenen performansa karşılık gelmektedir. Şekil 5.7'de yer alan cep telefonu

görüntü yenileme zamanları, 640x480 ve 1280x720 çözünürlüklerindeki işlemlerin oldukça düzgün ve +-5 milisaniye aralığında kaldığını göstermektedir. Ancak, 1920x1080 çözünürlüğündeki işlemlerin ekran yenileme zamanlarının yükseldiği ve ortalama olarak 50 milisaniye mertebesine çıktığı görülmektedir. Ayrıca, zaman verileri düzensizleşmiştir. Bu sonuç göstermektedir ki, grafik işleme performansı, görüntü katmanı verilerinin çözünürlüğü 1920x1080 üzerine çıkartıldığında 30 fps (*frame per second*) değerinden 20 fps değerine düşmektedir. Cep telefonu üzerinde yer alan grafik işlemci tüm devresi, işlemcisi ve bellek performansı bu çözünürlük değerindeki katmanları işlemekte zorluk çekmektedir. Benzer bir sonuç, Şekil 5.8’de görüldüğü üzere akıllı gözlük cihazı üzerinde de alınmaktadır. 640x480 ve 1280x720 çözünürlüklerinde sabit ve kararlı ilerleyen ekran yenileme zamanları, 1920x1080 çözünürlüğü altında işlem yaparken düzensizleşmekte ve anlık aşırı yükselmeler ve düşmeler sergilemektedir. 30 milisaniyenin oldukça altına inen bu değerler göstermiştir ki, işlemci kaynakları açısından oldukça zayıf kalan akıllı gözlük donanımı yüksek çözünürlük değerlerinde çalışırken, ekran yenileme zamanını ölçen yazılım bileşenini dahi baskılamakta ve hatalı ölçümler alınmasına neden olmaktadır. Çizelge 4.1’de yer alan mobil cihazların ekran çözünürlükleri karşılaştırmasında görüldüğü üzere, test platformu olarak kullanılan akıllı gözlük donanımı Epson BT-350, en fazla 1280x720 çözünürlüğünü desteklemektedir. Dolayısıyla, grafik işlemciye dikte edilen 1920x1080 çözünürlüğünü sağlıklı ve kararlı bir performans ile işleyememesi doğaldır. Bu sonuç, Şekil 3.18’de yer alan mobil cihaz performans karşılaştırma sonuçlarının yer aldığı grafikte, farklı iki mobil cihaz verilerinde de görülmüştür. Bölüm 3.3 “Giyilebilir ve Mobil Donanım Platformları” dahilinde detayları aktarılan, mobil cihaz donanımlarının genel işlem kapasitesi analizinde kullanılan algoritma, ODG-R7 ve Samsung T-800 mobil cihazlarının kaynak kapasitesini zorlamış ve toplanan zaman verilerinde düzensizliklere neden olmuştur.

Sonuçlar göstermektedir ki, geliştirilen GÜ yazılımı, klasik AG yazılımlarına oranla görsel veri üretiminde daha iyi performans sergilemekte ve mobil cihazlar üzerinde üretilen AG uygulamalarında yeni bir mimariye kapı açmaktadır. Özellikle, saha uygulamalarında ihtiyaç duyulan, en az kaynak tüketimine sahip bileşenlerin kullanılması prensibinden hareketle, askeri uygulamalarda geliştirilecek olan AG teknolojisi içerikli sistemlerin, bu çalışma kapsamında yer alan yazılım mimarisini barındırması büyük avantajlar getirecektir. Daha az kaynak ihtiyacı, daha basit ve ucuz

donanımlar ile uygulama gerçekleştirmeyi sağlayacak, aynı ebatlarda ve kapasitedeki bataryalar ile daha uzun süreler çalışacak donanım teçhizatı sağlanmış olacaktır.

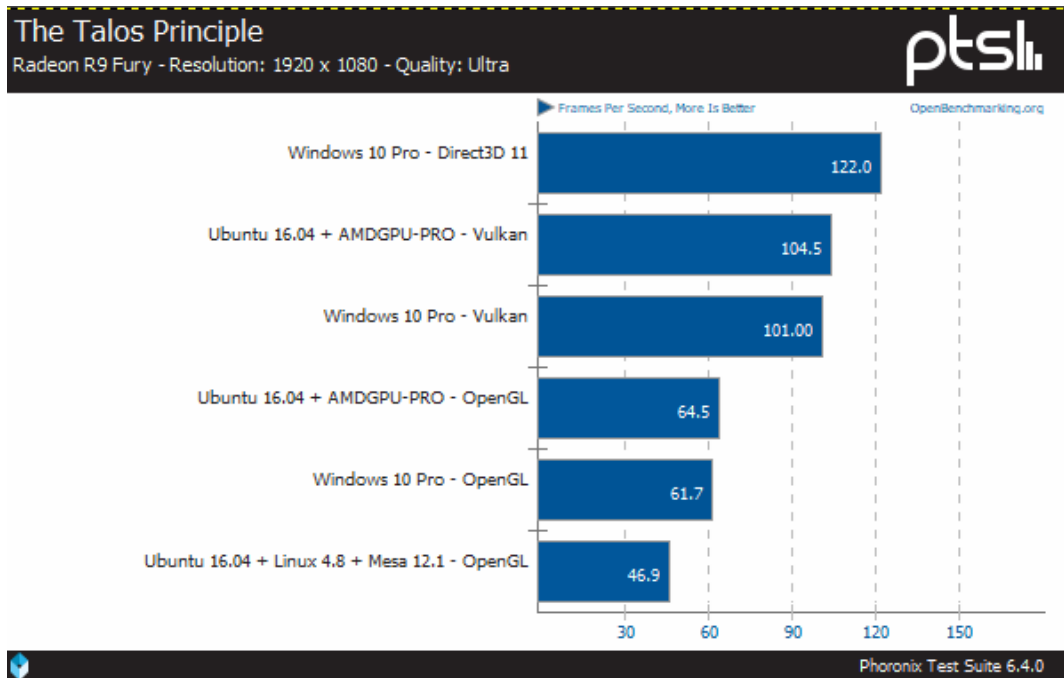
6.2 Öneriler

Tez kapsamında geliştirilen görüntü üretici yazılımı, artırılmış gerçeklik uygulamalarının temel mimarisini farklı bir yapı ile ele alarak, AG teknolojisinin temelinde yer alan grafik işleme ve sinyal işleme altyapılarını mobil cihazlardan soyutlamakta ve kaynaklar açısından kısıtlı olan bu cihazlar için daha güvenilir bir görüntüleme yazılımı önermektedir.

AG sahne yönetimi ve sinyal işleme algoritmalarının mobil cihaz sisteminden soyutlanması sonucu, uzak kaynaklar ile yüksek performanslı veri iletişimi gereksinimi doğmaktadır. Şekil 4.1’de aktarılan mimaride yer alan İşlem Merkezi ile kurulması gereken iletişim altyapısı, sağlıklı bir AG uygulaması geliştirilmesi açısından kritik öneme sahiptir. Günümüz teknolojisinde bu haberleşme altyapısı çeşitli tekniklerle sağlanabilecek durumdadır ancak yeni nesil haberleşme teknolojisi olan 5G iletişim teknolojisi, bu gereksinimi fazlasıyla karşılayacak performans parametreleri öngörmektedir. 5G teknolojisi üzerine Qualcomm firması yaptığı testlerde, laboratuvar şartlarında 4.5 *gigabit* veri hızına ulaşıldığını açıklamıştır. Bu değer ulaşılan en yüksek hız değeri olarak değerlendirilmekle birlikte, ortalama veri aktarımı 1.4 *gigabit* olarak değerlendirilmektedir (Aktan ve Beswick, 2019). Bu veriler ışığında, tez çalışması kapsamına giren, yoğun görsel veri transferi yapan sistemler için 5G teknolojisi önemli çözümler ile birlikte gelmektedir. Ülkemizde Havelsan, Turkcell, Vodafone gibi büyük kurumlar tarafından üzerinde çalışılan 5G teknolojisi, yüksek miktardaki verilerin, gerçek zamanlı olarak aktarımını sağlayarak Endüstri 4.0 gereksinimlerini karşılayacak altyapılar sunacaktır. Tez kapsamının ilgi alanına giren haberleşme formatı görüntü aktarımıdır. 5G teknolojisinin sunduğu yüksek bant genişliği ve düşük gecikmeli iletişim, gerçek zamanlı bir uygulama olan AG teknolojisinin temelinde yer alabilir. Bu nedenle, 5G teknolojisi ile iletişim, 5G donanımlarının mobil cihazlarda yer alması, 5G veri ağları gibi konularda yapılacak olan çalışmaların, tez konusunun geliştirilmesinde önem taşıdığı düşünülmektedir (Omale, 2019).

Özellikle askeri teknolojilerin gelişmesi, sahada kullanılan sistemlerin yüksek veri işleme ve aktarma gereksinimini doğurmuştur. Bu nedenlerle, yazılım geliştiricilerden araç platform üreticilerine kadar, savunma sanayi dahilindeki kurumların büyük kısmı, askeri personelin saha çalışmalarında mobil haberleşme altyapılarını kullanmasına yönelik çalışmalar yürütmektedirler. Bu çalışmalar, tez kapsamında yorumlandığında, yüksek verimli ve gerçek zamanlı veri aktarımı ortamlarının sahada hazır bulunması sonucu, artırılmış gerçeklik uygulamalarının da sahada yer almasının önünün açılacağı düşünülmektedir.

Mobil cihaz üzerinde çalışan grafik işleme altyapısı için, üstün kaynak erişimi nedeniyle geliştirme süreçlerinde gerek duyulan dokümantasyon desteği ön plana çıkan ve uygulama performansı açısından yüksek seviyede başarımlar sunan OpenGL API tercih edilmiştir. Ancak günümüzde yeni nesil grafik işleme altyapısı olan Vulkan API, donanım üreticileri ve yazılım platform geliştiricileri tarafından kullanıma sunulmuş durumdadır. OpenGL, masaüstü platformlarda doğrudan, mobil platformlarda ise OpenGL ES versiyonu ile destek vermektedir. Vulkan ise, tek bir API ile tüm platformlara destek verebilen bir konumdadır. Şekil 6.1’de, openbenchmarking.org adresinde yayınlanan, Phoronix Test Suite ile gerçekleştirilen The Talos Principle adlı 3B grafik tabanlı oyun üzerine yapılmış karşılaştırmalı test sonucu yer almaktadır.



Şekil 6.1. OpenGL ile Vulkan API performans karşılaştırması (Larabel, 2016)

Test sonucundan da görüldüğü üzere Vulkan, OpenGL'e göre üstün bir performans sunmaktadır. Direct3D API test sonucunda daha iyi performans vermesine karşın, sadece Windows platformları desteklediği için OpenGL ve Vulkan gibi çoklu platform desteği sunan API'ler ile değerlendirilmesi uygun bulunmamıştır. Bu sonuçlara dayanarak, yapılacak yeni araştırma çalışmalarında, görüntü üreticiye ait grafik işleme altyapısında Vulkan API ile çalışılması, performans açısından daha üstün sonuçlar alınmasını sağlayacaktır.



KAYNAKLAR

- Aktan, S. ve Beswick, E., 2019, 5G teknolojisi nedir, ne kadar hızlıdır ve hayatımızı nasıl değiştirecek?, <https://tr.euronews.com/2019/02/28/5g-mobil-teknolojisi-nedir-ne-kadar-hizlidir-ve-hayatimizi-nasil-degistirecek-3g-4g-ve-5g>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Alexander, S. A., Rozo, J. S., Donadio, B. T., Tenhundfeld, N. L., de Visser, E. J. ve Tossell, C. C., 2019, Transforming the Air Force Mission Planning Process with Virtual and Augmented Reality, *2019 Systems and Information Engineering Design Symposium (SIEDS)*, 1-4.
- Anonim, 2018a, Augmented Reality Trends, <https://lumusvision.com/augmented-reality-trends-infographic/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Anonim, 2018b, Award-winning Smart Glove heading to Consumer Electronics Show in Las Vegas, <https://tssg.org/2018/01/award-winning-smart-glove-heading-to-consumer-electronics-show-in-las-vegas/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Anonim, 2019a, Akıllı gözlük TAKBUL'a ilgi büyük, <https://m.timeturk.com/akilli-go-zluk-takbul-a-ilgi-buyuk/haber-805356>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Anonim, 2019b, Modern OpenGL, <https://glumpy.github.io/modern-gl.html>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Anonim, 2019c, OpenCV, Camera Calibration and 3D Reconstruction, https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html: [Ziyaret Tarihi: 10 Temmuz 2019].
- Anonim, 2019d, Metavision, <https://www.metavision.com/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Anonim, 2019e, ANIWAA, <https://www.aniwaa.com/product/vr-ar/odg-r-7/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Anonim, 2019f, Leading owners of augmented and virtual reality (AR/VR) patents worldwide as of 2019, <https://www.statista.com/statistics/963685/worldwide-augmented-virtual-reality-patent-top-owners/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Anonim, 2019g, EPSON, <https://www.epson.com.tr/products/see-through-mobile-viewer/moverio-bt-350>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Baggio, D. L., 2012, Mastering OpenCV with practical computer vision projects, Packt Publishing Ltd, p.
- Chen, Y.-J., Hung, C.-Y. ve Chien, S.-Y., 2017, Distributed rendering: Interaction delay reduction in remote rendering with client-end GPU-accelerated scene warping technique, *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 67-72.
- De Crescenzo, F., Fantini, M., Persiani, F., Di Stefano, L., Azzari, P., Salti, S. J. I. C. G. ve Applications, 2010, Augmented reality for aircraft maintenance training and operations support, 31 (1), 96-101.

- Dormehl, L., 2019, Defense giant BAE wants to replace fighter jet controls with augmented reality, <https://www.digitaltrends.com/cool-tech/bae-systems-augmented-reality-planes/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Fogel, S., 2018, 'Pokémon Go' Global Revenue Grew 37% in 2018 (Analyst), <https://variety.com/2019/gaming/news/pokemon-go-global-revenue-2018-1203098512/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Gallagher, D. G., Manley, R. J., Hughes, W. W. ve Pilcher, A. M., 2017, Divers augmented vision display (DAVD) emerging technology development, *OCEANS 2017-Anchorage*, 1-7.
- Hooton, C., 2016, Pokemon GO UK release date - and how to get it now on iPhone and Android, <https://www.independent.co.uk/arts-entertainment/pokemon-go-uk-release-date-and-how-to-get-it-now-on-iphone-and-android-apk-a7127326.html>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Jamak, A., 2017, A World of AR and Magic Sunglasses, <https://blog.prototypr.io/a-world-of-ar-and-magic-sunglasses-593e247e9822>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Kessenich, J., Baldwin, D. ve Rost, R. J. L. v., 2004, The OpenGL shading language, 1.
- Kore, A., 2018, Fundamentals of display technologies for Augmented and Virtual Reality, <https://hackernoon.com/fundamentals-of-display-technologies-for-augmented-and-virtual-reality-c88e4b9b0895>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Larabel, M., 2016, Direct3D 11 vs. OpenGL vs. Vulkan Radeon Benchmarks On Windows & Linux, https://www.phoronix.com/scan.php?page=news_item&px=D3D11-OpenGL-Vulkan-Radeon-AUG: [Ziyaret Tarihi: 10 Temmuz 2019].
- Lv, Z., Halawani, A., Feng, S., Ur Rehman, S., Li, H. J. P. ve Computing, U., 2015, Touch-less interactive augmented reality game on vision-based wearable device, 19 (3-4), 551-567.
- Madrigal, A. C., 2014, How DARPA's Augmented Reality Software Works? Why is the military succeeding where Google Glass failed?, <https://www.theatlantic.com/technology/archive/2014/05/how-darpas-augmented-reality-software-works/371652/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Milscint, 2017, DSEI 2017 Özel Dosya: HAVELSAN, Komuta Kontrol ve Savaş Yönetim Sistemleri ile Londra'daydı, <http://www.milscint.com/tr/dsei-2017-ozel-dosya-havelsan-komuta-kontrol-ve-savas-yonetim-sistemleri-ile-londradaydi/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Moriarty, T., 2017, A Brief History of Vertical Video (So Far), <https://medium.com/@terezdotcc/a-brief-history-of-vertical-video-so-far-51701f889f15>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Omale, G., 2019, Gartner Says 100 Million Consumers Will Shop in Augmented Reality Online and In-Store by 2020, <https://www.gartner.com/en/newsroom/press-releases/2019-04-01-gartner-says-100-million-consumers-will-shop-in-augme>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Osborn, J., 2018, Microsoft's HoloLens will be used in US Army combat missions, <https://www.techradar.com/news/microsofts-hololens-will-be-used-in-us-army-combat-missions>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Palmarini, R., Erkoyuncu, J. A., Roy, R., Torabmostaedi, H. J. R. ve Manufacturing, C.-I., 2018, A systematic review of augmented reality applications in maintenance, 49, 215-228.

- Pavlik, R. A. ve Vance, J. M., 2012, VR JuggLua: A framework for VR applications combining Lua, OpenSceneGraph, and VR Juggler, *2012 5th workshop on software engineering and architectures for realtime interactive systems (SEARIS)*, 29-35.
- Perez, P. G., Beer, W. ve Dorninger, B., 2013, Remote rendering of industrial HMI applications, *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, 276-281.
- Rathnayake, W. P., 2018, Usage of Mixed Reality for Military Simulations, *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 1-5.
- Sertin, C., 2018, Italtipress Gauss uses AVEVA's virtual and augmented reality for maintenance, <https://www.oilandgasmiddleeast.com/32551-italpresse-gauss-leverages-avevas-virtual-and-augmented-reality-for-maintenance>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Shi, S., 2011, Reduce latency: The key to successful interactive remote rendering systems, *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 391-392.
- StereoLabs, 2019, Capturing a Spatial Map, Mesh, <https://www.stereolabs.com/docs/spatial-mapping/>: [Ziyaret Tarihi: 10 Temmuz 2019].
- Suthau, T., Vetter, M., Hassenpflug, P., Meinzer, H.-P., Hellwich, O. J. I. A. O. P. R. S. ve SCIENCES, S. I., 2002, A concept work for Augmented Reality visualisation based on a medical application in liver surgery, 34 (5), 274-280.
- Wagner, D. ve Schmalstieg, D., 2003, First steps towards handheld augmented reality, *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, 127-135.
- Wira, N. N., 2017, Ikea, Apple collaborate for augmented reality app, <https://www.thejakartapost.com/life/2017/06/21/ikea-apple-collaborate-for-augmented-reality-app.html>: [Ziyaret Tarihi: 10 Temmuz 2019].
- XU, D., TAN, Z. ve GAO, Y. J. J. o. N. A. U., 2006, Developing application and realizing multiplatform based on Qt framework [J], 3 (2006), 018.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Resul AYDOĞAN
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Konya / 19.10.1987
Telefon : 0 555 4949009
Faks : -
E-Posta : resulaydogan@gmail.com

EĞİTİM

| Derece | Adı | İlçe | İl | Bitirme Yılı |
|---------------|---|------|----|--------------|
| Lise | : Selçuklu Atatürk Lisesi, Selçuklu, KONYA, | | | 2004 |
| Üniversite | : Dumlupınar Ün., Merkez, KÜTAHYA, | | | 2011 |
| Yüksek Lisans | : | | | |
| Doktora | : | | | |

İŞ DENEYİMLERİ

| Yıl | Kurum | Görevi |
|-----------|-----------------------------|-------------------------|
| 2019- | HAVELSAN A.Ş. | Kıdemli Ar-Ge Mühendisi |
| 2016-2019 | BİTES SAVUNMA A.Ş. | Ar-Ge Takım Lideri |
| 2011-2016 | AKINSOFT, AKINROBOTICS LTD. | Baş Tasarım Mühendisi |

UZMANLIK ALANI

Artırılmış Gerçeklik Teknolojileri
 Görüntü ve Sinyal İşleme Teknolojileri
 Robotik Teknolojiler (İnsansı Robot Teknolojileri)
 Otonom Sistemler ve Savunma Sanayi Uygulamaları
 Gömülü Sistem Tasarımı ve Yazılım Geliştirme Süreçleri
 Kontrol Sistemleri Teorisi ve Uygulamaları
 Algoritma Tasarımı
 Sistem Tasarımı ve Uygulamaları

YABANCI DİLLER

İngilizce

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

Tübitak Teydeb Proje Hakemliği – 2019

Türk Kızılayı Selçuklu Şube Yönetim Kurulu Üyeliği - 2019

YAYINLAR