T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MİG-MAG KAYNAĞI DİKİŞ FORMUNUN VOKSELLEŞTİRİLMİŞ PARABOLOİD KULLANARAK ÜÇ BOYUTLU GÖRÜNTÜSÜNÜN ELDESİ

DOKTORA TEZİ

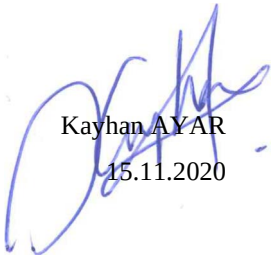
Kayhan AYAR

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Prof. Dr. Cemil ÖZ**

Aralık 2020

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.



Kayhan AYAR

15.11.2020

TEŞEKKÜR

Doktora eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, değerli danışman hocam Prof. Dr. Cemil ÖZ'e teşekkürlerimi sunarım.

Beni bugünlere getiren ve hayatımın her döneminde bana destek olan aileme teşekkürlerimi sunarım.

Doktora süresi boyunca birlikte çalıştığımız ve tez çalışmamda bana çok katkısı geçen arkadaşım Dr.Öğr. Üyesi Soydan SERTTAŞ'a ve Dr.Öğr. Üyesi Gülüzar ÇİT'e teşekkürlerimi sunmarım

Ayrıca çalışmamı maddi açıdan destekleyen Sakarya Üniversitesi Bilimsel Araştırma Projeleri (BAP) Komisyon Başkanlığına (Proje No: 2015-50-02-033) teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

2B	: İki-boyutlu
3B	: Üç-boyutlu
B-rep	: Boundary representation
cm	: Santimetre
CT	: Computer Tomography / Bilgisayarlı Tomografi
GLSL	: Graphics Layer Shader Language
GMA	: Gas Metal Arc
GMAW	: Gas Metal Arc Welding
GPU	: Graphics Processing Unit
GTA	: Gas Tungsten Arc
HIP	: Haptic Interaction Point / Haptic Etkileşim Noktası
HMD	: Head Mounted Device / Başa Takılan Ekran
MAG	: Metal Active Gaz / Metal Aktif Gaz
MC	: Marching Cubes
MIG	: Metal Inert Gaz / Metal Pasif Gaz
MRI	: Magnetic Resonance Imaging / Manyetik Rezonans Görüntüleme
ms	: Milisaniye
MSE	: Mean-Squared Error / Ortalama Hata Karesi
OBB	: Oriented Bounding Box
OpenGL	: Open Graphics Library / Açık Grafik Kütüphanesi
RAM	: Random Access Memory / Rastgele Erişimli Bellek
sn	: Saniye
TIG	: Tungsten Inert Gaz
TrainLM	: Train Levenberg-Marquardt
YSA	: Yapay Sinir Ağı

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Kaynak dolgusu, sanal kaynak simülatörü, vokselizasyon, yürüyen küpler, sekizli ağaç veri yapısı

Bu çalışmada gerçek zamanlı bir sanal kaynak simülatörü ve bu simülatörde kullanılacak yeni bir kaynak dikiş modeli geliştirilmiştir. Kaynak simülatörü gerçek bir kaynak işlemini sanal ortamda gerçekleştirebilmek için kullanılmaktadır. Bu sayede gerçek kaynak eğitiminde meydana gelebilecek kazalardan uzak bir ortam oluşturulmasının yanı sıra eğitim sırasında kullanılan malzeme israfının da önüne geçilmiş olmaktadır. Ayrıca öğrenciler, başlarında herhangi bir eğitimci bulunmaksızın simulator üzerinde kaynak işlemini istedikleri kadar gerçekleştirebilirler.

Simülatör, kullanıcıların gerçekçilik hissini kaybetmemeleri için gerçek zamanlı olarak gösterim yapacak şekilde tasarlanmıştır. Buna göre bir saniye içerisinde kullanıcılara gösterilen resim sayısının etkileşimi bozacak sayıların altına düşmemesi amaçlanmıştır. Simülatörün yazılım mimarisi ve kaynak dikiş modelinin elde edilirken kullanılan algoritmalar gerçek zamanlı çizim elde edilecek şekilde tasarlanmıştır.

Simülatörle kaynak işlemini gerçekleştirebilmek için çeşitli hareket ve oryantasyon sensörleri kullanılmıştır. Kullanıcının el hareketlerini takip edebilmek için torç üzerine bir adet hareket ve oryanstasyon sensörü yerleştirilmiştir. Bu sensör aracılığıyla kaynak esnasında kullanıcının el hareketleri devamlı olarak yakalanmakta ve elde edilen hareket verilerine göre de bir kaynak dikişi sahne üzerine yerleştirilmektedir

Gerçek kaynak dikişine benzerliği göz önüne alınarak simülatörde oluşturulacak kaynak dikiş modeli için paraboloid şekli seçilmiştir. Paraboloid modellerini sanal ortamda oluşturabilmek için gerekli olan parametreler için bir yapay sinir ağı geliştirilmiştir. Yapay sinir ağı kullanıcının hareket verilerini giriş parametresi olarak alıp elde edilecek kaynak dikişinin parametrelerini çıkış olarak vermektedir. Yapay sinir ağını eğitirken literatürde yapılmış testlerden elde edilen veriler kullanılmıştır. Birden fazla paraboloid şeklinin birleştirilmesi ve nüfuziyetin gösterilebilmesi için kaynak dikişi vokselleştirilmeye tabi tutulmuştur. Elde edilen voksel verileri optimize edilmiş bir ağaç veri yapısında saklanarak erişim hızı arttırılmıştır. Voxel bilgileri kullanılarak yürüyen küpler algoritması aracılığıyla sahne üzerinde gösterilecek üçgen yüzeyler elde edilmiştir. Kaynak dikişinin elde edilmesi işlemi basamaklarının farklı çözünürlüklerde aldığı süreler sonuçlar bölümünde gösterilmiştir.

DESIGNING A NEW THREE DIMENSIONAL MIG-MAG WELD SEAM FORM BY VOXELIZED PARABOLOID

SUMMARY

Keywords: Weld seam, virtual welding simulator, voxelization, marching cubes

In this study, a real-time and three-dimensional weld seam form was modeled for a low-cost virtual welding simulator developed for training welder candidates. Through this simulator, candidates can learn welding techniques in a safe environment without causing any work accidents and improve their skills by performing more applications than usual in a short time. In the developed simulator, special virtual reality devices such as Flock of Birds position and orientation sensor and head mounted display are used.

The simulation determines the weld bead shape and amount of penetration based on data from the Flock of Birds sensor device monitoring the position of the torch. When forming the weld bead shape, parabola was used as the basic bead shape unit due to the similarity of the weld bead slice with the parabola. The values of the height, width and penetration parameters of the basic weld bead shape that will form our weld seam were obtained from the weld seam experiments in the literature.

During the virtual welding process, the weld bead shape parameter values are calculated at specified time intervals using the feed-forward back-propagation artificial neural network. TrainLM (Levenberg-Marquardt) was used as the training function for network design. While determining the most appropriate transfer function, it was found that LogSig () function gave the best result. The number of hidden layers and the number of process elements (neurons) in each hidden layer were determined by trial and error method.

In the same time interval, the voxel map and the corresponding hash-based octree data structure are generated in real time. By using voxelized data, the triangular isosurfaces of the weld bead are reconstructed using the marching cubes algorithm. This results a more realistic weld seam appearance. This image and virtual scene are continuously sent to the head mounted display to maintain the sense of reality in the virtual environment.

Multi-threaded programming technique is also used to shorten the processing time in high resolution virtual scenes for voxelization and isosurface extraction processes. The isosurface extraction times for different number of threads are also shown.

BÖLÜM 1. GİRİŞ

Kaynak genellikle metal ve termoplastik materyalleri yüksek ısı kullanarak birleştirmek için kullanılan bir işlemdir. Günümüzde kaynak işlemi programlanabilir kaynak makineleri ile otomatik olarak güvenli bir şekilde gerçekleştirilebilmesine rağmen endüstride bu makinelerin kullanılamadığı birçok alanda tecrübeli kaynakçılara ihtiyaç duyulmaktadır. Kaynak işlemindeki birleştirmenin kalitesini kaynağı yapan kişinin el becerisi direk etkilemektedir. Bu yüzden kaynakçı operatörlerinin iyi bir eğitim sürecinden geçmesi gerekmektedir. Bu süreç uzun ve masraflı olmasının yanı sıra tehlikeli de olabilmektedir. Kaynak işleminde yüksek ısı ve elektrik akımlarının kullanılması eğitim ortamındaki tehlike faktörünü arttırmaktadır. Ayrıca kaynakçıların el becerilerini arttırmak için birçok defa kaynak işleminin gerçekleştirilmesi gerekmektedir. Bu işlemler materyal ve enerji kaybına sebep olmaktadır. Kaynak eğitimindeki masrafları azaltmak ve kullanıcılara daha güvenli bir eğitim ortamı sunmak için sanal kaynak simülatörleri kullanılmaktadır. Kaynak simülatörleri gerçek kaynak ortamını sanal ortamda taklit ederek kaynakçı adaylarının sanal ortamda kaynak işlemi gerçekleştirmelerini sağlamaktadır. Bu tez çalışmasında sanal bir kaynak simülatörü ve yeni bir kaynak dikiş modeli geliştirilmiştir.

1.1. Tezin Amacı

Bu tez çalışmasının amacı, kaynakçı eğitiminde gerçekleşebilecek kazalardan uzak güvenli bir ortam sağlayacak, malzeme israfını önleyen ve eğitimciler üzerindeki yükü azaltacak bir sanal kaynak simülatörü ile bu simülatörde kullanılacak yeni bir kaynak dolgu modeli geliştirmektir.

Kaynakçı eğitimi, yeni başlayan kaynakçı adayları için tehlikeli olabilecek bir ortamda gerçekleştirilmektedir. Kaynak işleminde kullanılan yüksek gerilim, materyallerin erimesi sonucu ortaya çıkan tehlikeli gazlar, ışınlar ve kıvılcımlar kaynakçı adayları için tehlikeli olabilmektedir. Ayrıca kaza geçiren bazı öğrenciler psikolojik sebeplerden dolayı eğitimlerine devam etmekte zorlanabilmektedir. Sanal kaynak simülatörleri sayesinde eğitim bu tür tehlikelerin bulunmadığı güvenli bir ortamda gerçekleştirilebilmektedir.

Kaynakçı eğitimi sırasında adaylar el becerilerini arttırmak için birçok kaynak işlemi gerçekleştirmektedir. Bu işlemler sırasında elektrik, temrin materyali, elektrot gibi malzemeler kullanılmaktadır. Eğitim uzadıkça harcanan materyal ve enerjinin masrafı da artmaktadır. Sanal kaynak simülatörleri sayesinde eğitim sırasında harcanan enerji azaldığı gibi materyal israfı da ortadan kalkmaktadır.

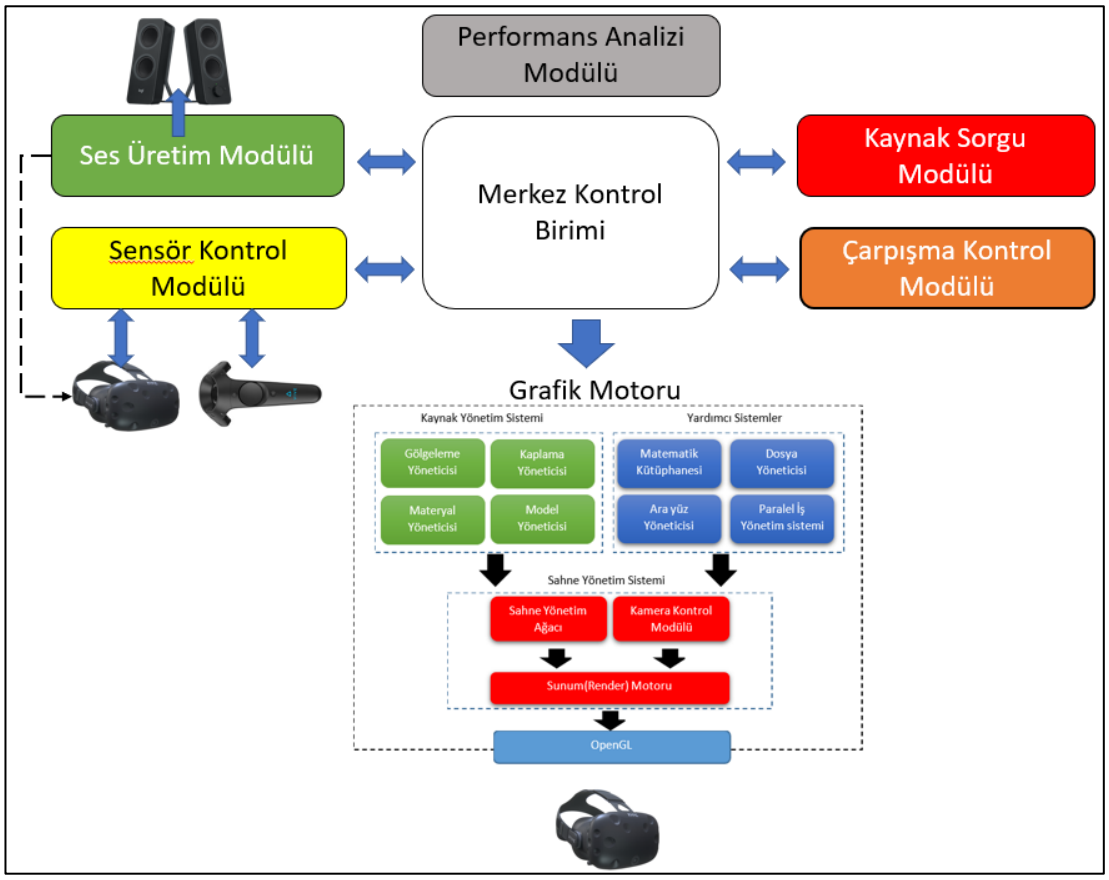
Kaynakçı adayları sanal kaynak simülatörlerini kullanarak eğitimcinin yardımı olmadan kaynak işlemi gerçekleştirebilmektedir. Simülatör içerisinde başarı analiz modülü sayesinde gerçekleştirdikleri kaynağın başarısını kontrol edebilmektedir.

1.2. Geliştirilen Sanal Kaynak Simülatörü

Bu tez çalışmasında kaynakçı eğitiminde kullanılmak üzere bir sanal kaynak simülatörü geliştirilmiştir. Geliştirilen simülatör aracılığıyla öğrenciler gerçek kaynak ortamının tehlikelerine maruz kalmadan güvenilir bir ortamda kaynak çalışmaları yapabilmektedir. Bu sayede kullanıcılar, birçok tekrar gerektiren el becerisini malzeme ve enerji israfı yapmadan geliştirilen simülatörü kullanarak gerçekleştirebilmektedirler.

Simülatör ile kaynak uygulaması yapılırken kullanıcının el hareketleri torç üzerindeki hareket ve oryantasyon algılayan sensörler tarafından takip edilmektedir. Sensörlerden alınan verilere göre simülatör, torcun kaynak materyaline ark oluşturacak kadar yaklaşıp yaklaşmadığı kontrol etmektedir. Ark oluşturacak mesafeye ulaşıldığında kaynak dolgusu sorgu birimi sensörlerden alınan verileri kullanarak çizilecek olan

kaynak dolgusunun genişlik, yükseklik ve nüfuziyet parametrelerini belirlemektedir. Ardından grafik motoru kaynak dolgu modelini oluşturup sanal ortam içerisine eklemektedir. Bu sayede kaynakçı yaptığı işlemlerin sonucunu gerçek zamanlı olarak görmektedir. Kaynak işlemi bittikten sonra performans analiz birimi sayesinde kaynakçı yaptığı kaynak işleminin başarısını ölçebilmektedir. Ayrıca kullanılan veri tabanı sayesinde daha önceden yaptığı kaynak denemelerindeki gelişmelerini de takip edebilmektedir. Simülatör, grafik motoru, sensör kontrol modülü, kaynak sorgu modülü, ses üretim modülü, çarpışma kontrol modülü ve performans analizi modülü olmak üzere altı temel kısımdan oluşmaktadır. Şekil 1.1.'de geliştirilen sanal kaynak simülatörünün temel yapısı verilmiştir.



Şekil 1.1. Sanal kaynak simülatörünün temel yapısı

Grafik motoru, OpenGL grafik kütüphanesi kullanarak simülatörün üç boyutlu sanal ortamını çizdirmek ile görevlidir. Grafik motoru kullanıcının sanal ortamla etkileşimini akıcı hale getirmek için saniyede en az 60 defa görüntü elde edebilmektedir. Böylece kullanıcılar yaptıkları işlemlerin sonuçlarını uygulama sırasında görebilmekte ve kendilerini sanal ortamın bir parçası olarak hissetmektedirler.

Sensör kontrol modülü, simülatörün kullandığı donanımsal cihazlar ile iletişimi sağlamaktadır. Bu cihazlar, hareket ve oryantasyon algılayıcısı HTC Vive yöneticisi, ve başa takılan ekrandır. Sensör kontrol modülü devamlı olarak simülatöre bağlı olan sensörleri kontrol eder ve onlardan gelen verileri merkez kontrol ünitesine gönderir. Sensör kontrol modülü simülatörün diğer birimlerinden farklı bir iplikte çalıştığı için cihazlar okunurken herhangi bir bekleme durumu oluşmamaktadır. Modül tasarlanırken, kullanılan cihazlar ile yazılım mimarisi arasında soyutlama tekniği kullanılmıştır. Bu sayede farklı sensörlerde simülatöre kolayca adapte edilebilmektedir.

Kaynak sorgu modülü kullanıcının el hareketlerine göre kaynak dolgusunun genişlik, yükseklik ve nüfuziyet parametrelerini elde etmek ile görevlidir. Kaynak uygulaması sırasında kullanıcının hareketleri devamlı olarak sorgu modülüne gönderilmektedir. Bu veriler, literatürdeki gerçek kaynak uygulamalarından elde edilmiş sonuçları kullanarak eğitilmiş bir yapay sinir ağına giriş parametesi olarak verilmektedir. Yapay sinir ağı sonuç olarak kaynak dolgusunun çizimi için gerek olan genişlik, yükseklik ve nüfuziyet parametrelerini vermektedir.

Çarpışma kontrol modülünün görevi kullanıcının el hareketi ile beraber sahne içerisinde hareket eden torcun sahnedeki diğer modellere yakınlığını takip etmektedir. Özellikle torcun ucundaki elektrotun kaynak materyaline olan uzaklığı devamlı olarak kontrol birimine gönderilmektedir. Eğer elektrot kaynak materyaline ark oluşturacak kadar yaklaşırsa kaynak işlemini başlatılır.

Performans anazili birimi, kaynak işlemi tamamlandıktan sonra devreye girmektedir. Bu birimin temel görevi kullanıcının gerçekleştirdiği kaynak işleminin başarısını ölçmektir. Performan analizi işlemi, kaynak tecrübesine sahip akademisyenlerin geliştirdiği uzman sistem tarafından yapılmaktadır [1]. Performans analizi modülü ayrıca kullanıcıların önceki kaynak işlemlerini de kaydetmektedir. Böylece kullanıcının zaman içerisindeki gelişimi de gözlemlenebilmektedir.

1.3. Önerilen Kaynak Dolgu Modeli

Bu tez çalışmasında vokselleştirilmiş paraboloid modeli kullanılarak, sanal sahne üzerine gerçek zamanlı bir şekilde eklenebilecek yeni bir kaynak modeli geliştirilmiştir. Yapılan literatür araştırması sonucu kaynak dolgusu kesit modelinin genelde parabole benzetildiği görülmüştür [2]. Fakat üç boyutlu sanal bir ortamda sadece iki boyutlu bir parabol şekli kaynak dolgu modeli elde etmek için yeterli olmamaktadır. Bu yüzden kaynak dolgu modeli olarak parabolün üç boyutlu formu olan paraboloid temel kaynak dolgu modeli olarak seçilmiştir.

Kaynak işlemi sırasında kaynak dolgu modelleri ard arda gelecek şekilde yerleştirilmesi gerekmektedir. Bu işlem sırasında dolgu modelleri arasındaki geçişleri yumuşatmak ve kaynak materyali altındaki nüfuziyeti göstermek için paraboloid modeli vokselleştirilmektedir. Vokselleştirilmiş model daha sonra yürüyen küpler algoritması kullanılarak çizime daha uygun üçgenlerden oluşan yüzeylere çevrilmektedir. Bu işlemler, kullanıcı farkında olmayacak şekilde arka planda yapılmaktadır. Kaynakçı adayı sadece kaynak dolgu modelini temsil eden üçgen yüzeyi görmektedir.

Kaynak modeli geliştirirken simülatörün gerçek zamanlı çalışması gerektiği göz önünde bulundurulmuştur. Gerçek zamanlı çizim uygulamalarında, sanal ortamı izleyen kullanıcıların gerçekçilik hissini kaybetmeyecek şekilde görüntünün tazelenmesi gerekir. Yeni kaynak dolgu modeli ile saniyede ekrana çıkartılan görüntü sayısı altmışın altına hiç düşmemektedir. Farklı çözünürlük oranları ile kaynak dolgusu oluşturma süreleri sonuçlar kısmında verilmiştir.

1.4. Tezin Organizasyonu

Tezin bundan sonraki bölümleri aşağıdaki gibi düzenlenmiştir:

Bölüm 2'de, günümüze kadar yapılmış olan kaynak simülatörleri, kaynak dolgusu için kullanılan şekiller ve bu şekillerin elde edilmesinde kullanılan parametrelerin elde edilmesi ile ilgili yapılan çalışmalar incelenmiştir. Bu çalışmaların tez çalışmamızda izlediğimiz yolu nasıl etkilediklerinden bahsedilmiştir.

Bölüm 3'de tez çalışmasında geliştirilen kaynak simülatörünün temel mimari yapısından bahsedilmiştir. Ardından simülatörün bir kaynak işlemi sırasında gerçekleştirdiği işlemler sırayla anlatılmış ve akış diyagramı verilmiştir. Simülatörde kullanılan sensörlerin özellikleri ve simülatör içerisindeki görevlerine de yer verilmiştir.

Bölüm 4'de geliştirilen simülatörün çizim ve görüntü işlemlerinden sorumlu birimi olan grafik motorundan bahsedilmiştir. Öncelikle grafik motorunun tanımı yapılıp çalışmamızdaki yeri anlatılmıştır. Grafik motorunun genel yazılımsal yapısının yanı sıra ekran kartı ile iletişim için kullanılan kütüphanelerden de yine bu bölüm içerisinde bahsedilmiştir.

Bölüm 5'de gerçek kaynak dolgu modelleri incelenerek, kaynak simülatörün de kullanılacak olan sanal modelin nasıl çıkartıldığı anlatılmıştır. Kaynak dolgu modelini simütör üzerinde gösteribilmek için uygulanan işlemlerden bahsedilmiştir. Son olarak kaynak dolgu modelinin elde edilmesinde kullanılan parametrelerin nasıl elde edildiği anlatılmıştır.

Bölüm 6'da bu çalışmada geliştirilen yeni kaynak dolgu formunun getirdiği avantajlardan bahsedilmiş ve ilgili sonuçlar tablolar halinde gösterilmiştir.

BÖLÜM 2. LİTERATÜR TARAMASI

Kaynak, dolgu çubuklu veya çubuksuz ısı uygulamasıyla aynı veya farklı iki metalin birleştirme işlemidir. Günümüz teknolojisinde kaynak kullanımı endüstrinin her dalında, mekanik endüstrisinde vb. kullanılmaktadır. Gaz alevi, elektrik arkı, lazer, elektron ışını vb. dahil olmak üzere birçok farklı enerji kaynağı kaynak için kullanılabilmektedir. Çoğu zaman endüstriyel bir işlem olmakla birlikte, kaynak açık hava, su altı ve uzayda dâhil olmak üzere birçok farklı ortamda gerçekleştirilebilmektedir.

Bu bölümde, kaynak işlemi için gerekli olan dikiş geometrisi modellemeleri hakkında geçmişte yapılan çalışmalardan bahsedilmiştir. Özellikle MIG kaynak teknolojisi üzerine yapay sinir ağı tekniğinin katkısı ve analize dayalı sonuçları incelenmiştir. Ark kaynağında kaynak özelliklerinin kalitesini kontrol edebileceğimiz matematiksel modeller bulunmasına karşın yapay sinir ağı girdi ve çıktı arasında bazı ilişkilerin olduğu yerlerde başarıyla kullanılmaktadır [3]. Artık günümüzde yapay sinir ağları, girdi parametresi ve son çıktının birbiriyle ilişkilendirilebildiği ve bunu sinir ağı tarafından verilen değerle karşılaştırılabildiği ve değeri optimize edebildiği çok kullanışlı bir araçtır.

Birçok araştırmacı, Taguchi, Response Surface yöntemi, tam faktöryel tasarımı, Box-Behnken tasarımı gibi çeşitli deney tasarım tekniklerini kullanarak çeşitli lazer türleri ile farklı tipte malzemeler üzerinde çalışmıştır [4]. Ayrıca elektrot boyutu, kaynak akımı, ark gerilimi, ark hareket hızı, kaynak pozisyonu, gaz akış hızı, koruyucu gaz bileşimi gibi çeşitli proses parametrelerinin çeşitli malzemeler üzerindeki etkisi ve çekme mukavemeti, sertlik gibi çıktı parametrelerine etkisi incelenmiştir.

Dutta ve Pratihar çalışmasında [5], geleneksel regresyon analizi ve sinir ağı tabanlı yaklaşımları kullanarak TIG kaynak sürecinin modellemesini yapmışlar ve geleneksel regresyon analizi ile karşılaştırmışlardır. Sinir ağı yaklaşımının geleneksel regresyon analizinden daha iyi olduğunu, nedenininde sinir ağı tabanlı yaklaşımın belirli bir aralıkta interpolasyon gerçekleştirebilmesi olduğunu belirtmişlerdir.

Pal çalışmasında [6], Darbeli MIG kaynağında çeşitli torç açıları için sensör sinyallerini kullanarak kaynak bağlantı mukavemetini incelemiş ve darbeli metaldeki düşük karbonlu çelik alın kaynağının gerilme özellikleri üzerindeki çeşitli torç açılarındaki darbe parametrelerinin etkisine odaklanmıştır. Kaynak bölgesi ile ısıdan etkilenen bölgenin arayüzü, kaynak mikro yapısındaki önemli farklılıklar nedeniyle en zayıf alan olarak bulunmuştur. Torç açısındaki değişikliğin kaynak dikiş karakteristiğini ve mikro yapısıwnı önemli ölçüde etkilediğini görmüştür. Tüm torç açıları için kaynak bağlantı gücünün ne şekilde etkilendiğini incelemiş ve ark basıklığı, ark gücü ve kaynak tepe sıcaklığının kaynak bağlantı kalitesi için yararlı olduğunu göstermiştir.

Planckaert çalışmasında [7], kısa ark mesafesinde MIG/MAG kaynak işleminin fiziksel modellemesindeki konuları araştırmış ve anahtarlamaları iki koruma koşulu tarafından kontrol edilen iki farklı sürekli duruma sahip bir hibrit model önermiştir. Önerilen modeli doğrulamak amacıyla, yüksek hızlı dijital video ile eşzamanlı olarak örneklenen besleme voltajı ve akım dahil olmak üzere farklı ölçümler yapılmıştır. Görüntü dizilerinden metal transferini temsil eden bazı ilgili miktarları çıkarmak için, aktif bir kontur algoritması geliştirilmiş ve test edilmiştir. Önerilen modelin, özellikle ark durumunda bir kaynak işleminin ana eğilimlerinin tahmininde etkinliği, deneysel veriler kullanılarak gösterilmiştir. Metal transferi sırasında modelin bazı sınırlamaları da vurgulanır ve daha sonra olası çözümler önerilmiştir.

Ates çalışmasında [8], gaz metal ark kaynağı parametrelerinin tahmini için yapay sinir ağlarına dayalı bir teknik sunmuştur. Modelin girdi parametreleri gaz karışımlarından oluşurken, YSA modelinin çıktıları sırasıyla çekme dayanımı, darbe dayanımı, uzama ve kaynak metali sertliği gibi mekanik özellikleri içermiştir. YSA denetleyicisi,

genişletilmiş delta-çubuk-delta öğrenme algoritması ile eğitilmiş, ölçülen ve hesaplanan veriler bir bilgisayar programı ile de simüle edilmiştir. Ates, düşük karbonlu çelik levhaların (15 x 150 x 450 mm) 180 A ve 28 V altında kaynaklandığı bir deney yapmıştır. Koruyucu gazın akış hızı 15 l/dk ve temas ucunun 15 mm'lik iş parçası mesafesine ayarlanmış, elektrot telinin çapı da 1,2 mm olarak kullanılmıştır. Modelin girdi parametreleri gaz karışımlarından oluşurken, YSA modelinin çıktıları sırasıyla çekme dayanımı, darbe dayanımı, uzama ve kaynak metali sertliği gibi mekanik özellikleri içermiştir. Çalışma, MIG yöntemi kullanılarak kaynaklı düşük alaşımlı çeliğin mekanik özelliklerinin hesaplanması için sinir ağlarının kullanım olasılığını göstermiştir. Sonuçlar, hesaplama sonuçlarının ölçülen verilerle iyi bir uyum içinde olduğunu da göstermiştir.

Pal çalışmasında [9], MIG kaynağı işleminde kaynak bağlantı gücü izlemeyi ele almıştır. Yanıt yüzey metodolojisi, kaynak deneylerini gerçekleştirmek için uygulanmış ve kaynaklı plakaların nihai çekme gerilimini tahmin etmek için çok katmanlı bir sinir ağı modeli geliştirilmiştir. Giriş olarak darbe gerilimi, arka zemin gerilimi, darbe süresi, darbe frekansı, tel besleme hızı ve kaynak hızı olmak üzere altı işlem parametresi ve kaynak akımı ve geriliminin ortalama karekök değerleri olmak üzere iki ölçüm kullanılmıştır. Modelin değişkenleri ve kaynaklı levhanın çekme gerilimi çıktı değişkeni olarak kabul edilmiştir. Ayrıca, çoklu regresyon analizi ile elde edilen çıktı, geliştirilen yapay sinir ağı modeli çıktısıyla karşılaştırmak için kullanılmıştır. Geliştirilen yapay sinir ağı modelinin öngördüğü kaynak dayanımının çoklu regresyon analizine göre daha iyi olduğu bulunmuştur.

Sathiya [10] 904 L süper östenitik paslanmaz çeliğin 3,5 kW soğutmalı levha lazer kaynağı üzerinde yapılan bir çalışmayı sunmuştur. Bu çalışmada ek yerleri, sabit bir akış hızında argon, helyum ve nitrojen gibi farklı koruyucu gazlarla kaynaklanmıştır. Süper östenitik paslanmaz çelik normalde yüksek miktarda Mo, Cr, Ni, N ve Mn içerir. Mekanik özellikler, iyi kaynaklı bağlantılar elde etmek için kontrol edilir. Eklemin kalitesi, kaynak parçası geometrisinin dikiş genişliği ve penetrasyon derinliği (DOP) gibi özellikleri incelenerek değerlendirilir. Bu yazıda, AISI 904 L SASS'den yapılan lazer kaynaklı alın bağlantılarının çekme dayanımı ve kaynak dikiş profilleri

incelenmiştir. Taguchi yaklaşımı, seçilen kaynak parametrelerini optimize etmek için istatistiksel bir deney tasarımı tekniği olarak kullanılmıştır. Gri ilişkisel analiz ve arzu edilirlik yaklaşımı, birden çok çıktı değişkenini aynı anda dikkate alarak girdi parametrelerini optimize etmek için uygulanmıştır. Optimize edilmiş parametreleri doğrulamak için her iki analiz için de doğrulama deneyleri yapılmıştır.

Sathiya diğer çalışmasında [11], lazer kaynağı girdi parametrelerinin kaynak bağlantısının kalitesinin belirlenmesinde çok önemli bir rol oynadığını belirtmiştir. İyi kaynaklı bağlantılar elde etmek için özellikle mekanik özelliklerin kontrol edilmesini önermiş ve çalışmasında, AISI 904L süper östenitik paslanmaz çelikten yapılan lazer kaynaklı alın derzlerinin penetrasyon derinliği, kaynak genişliği ve çekme dayanımı gibi kaynak dikiş geometrisini incelenmiştir. Deneysel tasarımı gerçekleştirmek için tam faktöryel tasarım kullanmıştır. Yapay sinir ağı, ışın gücü, hareket hızı ve odak konumu gibi lazer kaynağı girdi parametreleri ile üç farklı koruyucu gazdaki (argon, helyum ve diğer) üç yanıt penetrasyon, kaynak genişliği ve çekme gerilimi arasındaki ilişkiyi kurmak için MatLab yazılımında geliştirilmiştir. Oluşturulan modeller, proses parametrelerini genetik algoritma kullanarak optimize etmek için kullanılmıştır. Üç farklı gaz için optimum çözümler ve bunların karşılıkları elde edilmiş ve genetik algortima elde edilen optimize edilmiş parametreleri doğrulamak için doğrulama deneyi de yapılmıştır.

Achebo çalışmasında [12], çeliğin nihai gerilme mukavemetine karşılık kaynak akımı, voltaj, hız ve zaman gibi girdi parametrelerinin seçiminde, Taguchi Metodu yardımıyla optimizasyon sağlandığını göstermiştir. Taguchi Metodu uygulanarak yapılan analizden 240A kaynak akımı, 2.0 dakika kaynak süresi, 0.0062 m/s kaynak hızı ve 33V kaynak voltajı için optimum işlem parametresi önerilmiştir. Bu optimum parametrelerin, mevcut işlem parametrelerinin Sinyal/Gürültü oranında 2.32dB ve çekme gerilimine göre 1,11 kat iyileştirmeye sahip olduğu bulunmuştur. Bu çalışma, Taguchi Yöntemi'nin uygulanması için adım adım bir yaklaşımı açıklamaktadır.

Nagesh çalışmasında [13], elektrot besleme hızı, ark gücü, ark voltajı, ark akımı, ark uzunluğu gibi girdi parametrelerini seçerek ve kaynak dikişi yüksekliği, dikiş

genişliği, nufuziyet derinliği, nufuziyet alanı ve ark hareket hızı gibi çıktı parametrelerinin optimizasyonu için yapay sinir ağı yöntemini kullanarak kaynak dikiş geometrisini ve penetrasyonu tahmin etmiştir. Deneysel sonuçlar, önceden ısıtılmış plakaların veya düşük ark ilerleme hızının veya yüksek ark gücünün kullanımının daha iyi füzyon sağladığı sonucuna varmıştır. Ark hareket hızındaki artışla birlikte hem kaynak dikiş yüksekliği hem de genişliği azalır, ancak yüksek ark hızı oranına sahip daha düz bir dikiş yapmak için yüksekliğindeki azalma nispeten daha fazladır. Ark uzunluğunu sabit tutan elektrot besleme hızındaki artışla penetrasyon artmıştır.

Correia çalışmasında [14], Genetik algoritma kullanarak MIG kaynak parametresinin optimizasyonunu sunmuştur. Optime yakın arama, MIG kaynak işleminin girdileri ve çıktıları arasındaki modelleme denklemlerinin önceki ve bilgisine dayalı olarak bir sonraki deneyi tahmin eden genetik algoritma ile adım adım gerçekleştirilmiştir. Genetik algortima, nispeten az sayıda deneyle optimuma yakın koşullar oluşturmayı başarmış ancak genetik algortima tekniği ile optimizasyon, kuşak sayısı, popülasyon boyutu, vb. parametrelerin iyi bir şekilde ayarlanması gerektirdiği görülmüştür. Aksi takdirde, arama alanının yetersiz bir şekilde genişlemesi riski bulunduğu sonucuna varılmıştır.

Khuder ve arkadaşları çalışmalarında [15], MIG punto kaynağı kullanarak, benzer olmayan metallerin kaynak birleşimlerinde kaynak işlem parametresinin etkisini incelemişlerdir. Bu araştırmada kaynak için seçilen temel malzeme, östenitik paslanmaz çelik tipi AISI 316L ve karbon çeliğidir. Bu farklı metallerin kaynağında kullanılan dolgu metali E80S-G'dir ve koruyucu gaz olarak CO_2 kullanılmıştır. Deney, giriş parametresi olarak tel besleme süresi, besleme süresi ve kaynak akımı dikkate alınarak yapılmıştır. Bu parametrelerin nokta çapı ve kesme kuvveti üzerindeki etkisi deney yapılarak tahmin edilmiştir. Sonuçta, punto kaynağı boyutunun ve kesme kuvvetinin artan kaynak akımı ile arttığı, kaynak süresinin artmasıyla kesme kuvvetinin azaldığı sonucuna varmışlardır. Ayrıca, artan kaynak akımı ve kaynak süresinin de kaynak bölgesinin çapını artıracağını ve kesme kuvvetini azaltacağını bulmuşlardır.

Kumar çalışmasında [16], Yapay Sinir Ağı ve Genetik Algoritma kullanarak MIG kaynak parametrelerinin optimizasyonu üzerinde çalışmalar yapmıştır. Bu araştırma çalışmasında, 304 kalite paslanmaz çelik ve 316 kalite gibi farklı malzemelerin kaynağı sırasında kaynak gerilimi, kaynak hızı ve kaynak akımı gibi kaynak parametrelerinin nihai çekme gerilmesi üzerindeki etkisini tahmin etmek için yapay sinir ağı yöntemini kullanarak matematiksel model yapmıştır. Çıktı parametresinin değerini optimize etmek için kullanılan genetik algoritma kullanılmış ve analizden maksimum nihai gerilme mukavemetinin 110 A kaynak akımında, 18 V kaynak geriliminde ve 43.362 cm/dk hareket hızında karşılandığı sonucuna varılmıştır. Ayrıca Yapay Sinir Ağının diğer regresyon modeli olarak başarılı bir şekilde entegre edildiğini göstermiştir.

Hooda çalışmasında [17], AISI 1040 orta karbonlu çelik bağlantının MIG kaynağının gerilme mukavemetini tahmin etmek için bir yanıt yüzey modeli geliştirmiştir. Bu araştırmada kaynak gerilimi, akım, tel hızı ve gaz akış hızı girdi parametresi olarak ele alınmıştır. Deney, yüzey merkezli kompozit tasarım matrisi ile tasarlanmış ve bu deneyden hem enine hem de maksimum akma dayanımı için kaynak gerilimi 22,5 V, tel hızı 2,4 m/dk ve gaz akış hızı 12 l/dk gibi proses parametresinin optimum değerlerinin olduğu sonucuna varılmıştır. Burada mevcut akım değerleri sırasıyla 190 A ve 210 A'dir.

Balasubramanian çalışmasında [18], sürekli akım ve darbeli akım tekniğinin etkisi altında gaz metal ark kaynağı ve gaz tungsten ark kaynağı ile üretilen yüksek mukavemetli alüminyum alaşımlı derzleri incelemiştir. Darbeli akım gaz metal ark kaynağı bağlantıları, diğer kaynaklı bağlantılara göre yüksek mukavemet değerleri ve yüksek bağlantı verimliliği üretmiştir. İnce taneciklerden dolayı ana metal ve ısıdan etkilenen alan bölgeleri kaynak metaline göre yüksek sertlik değerleri üretmiştir. Darbeli akım gaz tungsten ark kaynak bağlantıları yüksek yükseklik değerleri ve sürekli akım gaz metal ark kaynak bağlantıları düşük sertlik değerleri üretmiştir.

Patel çalışmasında [19], Taguchi metodu ve Gray Relational Analizi ile MIG kaynağı ve TIG kaynağı için kaynak dikişi sertliği üzerindeki etkilerini araştırmak için kaynak

akımı, tel çapı ve tel besleme hızı parametreleri değerlendirmiştir. Çalışmadan kaynak akımının MIG ve TIG kaynağı için en önemli parametre olduğu sonucuna varılmıştır. Gray Relational Analizi optimizasyon tekniğinin kullanılmasıyla, optimum parametre kombinasyonunun kaynak akımı, 100 A, tel çapı 1,2 mm ve tel besleme hızı MIG kaynağı için 3 m/dk bulunmuştur.

Ghazvinloo çalışmasında [20], robotik MIG kaynağı için kaynak hızı, gerilim ve akım etkisi altında yorulma ömrü, darbe ve dikiş nufuziyet özelliklerini analiz etmiştir. Çalışmada 2,35 mm ve 10 mm kalınlığında 60 derece V oluklu plakaları 1 mm çapında dolgu malzemesi kullanılarak kaynaklanmıştır. İşlem sırasında kaynak parametreleri olan kaynak hızı, voltaj ve akım değiştirilmiştir. Artan gerilim ve akım yorulma ömrünü kısaltmış ancak kaynak hızı yorulma ömrünü uzatmıştır. Azalan kaynak hızı ve artan akım voltajı, darbe enerjisini iyileştirmiştir. Dikiş nufuziyetinin kaynak akımına bağlı olarak direk etkilenen büyüklük olduğu da belirtilmiştir.

Aktepe çalışmasında [21], 155 mm topçu mühimmatı üretimindeki kullanılan demir oranı ve MIG kaynak işlemi için yapay sinir ağı modeli kullanmıştır. Amaç, gerekli çıktı düzeylerine sahip girdilerin belirlenmesi ve işlemin hata için iyileştirilmesi ile kaynak işleminin iyileştirilmesi ve hatalı ürün oranının en aza indirilmesidir. Ağ modelinde kullanılan 22 giriş parametresi ve 3 çıkış parametresi vardır. Model, verilen girdi parametrelerine göre çıktı parametrelerinin değerlerini tahmin etmiştir. Ek olarak, kalite özelliklerine uygun bir kaynak işlemi için girdi parametrelerinin optimal değerleri de bulunmuştur.

Ibrahim çalışmasında [22], robotik gaz metal ark kaynağı yaparak 6 mm kalınlığında yumuşak çelik metal üzerine farklı parametrelerin kaynak penetrasyonu, mikroyapısal ve sertlik ölçümüne etkilerini araştırmıştır. Bu çalışmada tercih edilen değişkenler ark gerilimi, kaynak akımı ve kaynak hızıdır. Ark gerilimi ve kaynak akımı sırasıyla 22, 26 ve 30 V ve 90, 150 ve 210 A olarak ve kaynak hızı da 20, 40 ve 60 cm/dk olarak seçilmiştir. Kaynak işleminden sonra her numune için penetrasyon, mikroyapı ve sertlik ölçülerek etkisi incelenmiştir. Sonuç olarak, kaynak akımının parametre değerini artırmanın penetrasyon derinliği değerini artırdığı görülmüştür. Bunun

dışında, ark gerilimi ve kaynak hızı, penetrasyon derinliğinin değerini etkileyen diğer bir faktör olduğu da belirtilmiştir.

Shoeb çalışmasında [23], yüksek mukavemetli çelik kaynağında kaynak hızı, voltaj ve gaz akış hızı gibi çeşitli kaynak parametreleri değiştirilmiş ve bu parametrelerin nufuziyet, genişlik ve yükseklik gibi kaynak dikiş geometrisi üzerindeki etkileri incelenmiştir. Matematiksel denklemler faktöriyel tekniği kullanılarak geliştirilmiştir. Gerilim arttıkça yüksekliğin azaldığı ancak nufuziyetin artığı ve kaynak hızının nufuziyete etkisinin olmadığı belirtilmiştir.

Mishra çalışmasında [24], kaynak akımı, kaynak gerilimi ve kaynak hızı parametrelerinin kaynak sırasında AISI 1020 çeliğinin penetrasyon derinliği üzerindeki etkisini incelemiştir. Deneyi planlamak, verileri elde etmek ve kaynak parametrelerini ve süreci optimize etmek için Taguchi tekniğine dayalı bir deney planı kullanılmıştır. Kaynak parametrelerinin optimizasyonu için Taguchi ortogonal dizisi, sinyal-gürültü oranı ve varyans analizi kullanılmıştır. Penetrasyon analizinde etkinliği bulmak için deneysel değerler ile öngörülen değerler arasındaki farkı elde etmek için uyum testleri yapılmıştır.

Chaki çalışmasında [25], lazer kaynak işleminin modellenmesi için Yapay Sinir Ağı ve Genetik Algoritmanın hibrit bir modelini sunmuştur. Bu model, ilgili işlem parametreleri ile penetrasyon derinliğinin tahmini ve optimizasyonu için kullanılmıştır. Bu amaç için geliştirilen program, başlangıçta Bayesian regülasyonlu geri yayılımlı sinir ağı kullanarak optimize edilmiş bir ağ mimarisi kurar. Bu eğitimli ağ, daha sonra işlem için optimum parametreleri bulmak için genetik algoritma ile birlikte kullanılmıştır. CO_2 lazer-MIG hibrit kaynakta proses girdi parametrelerinin (ark gücü, iş parçası yüzeyinden odak mesafesi, torç açısı ve lazer ile kaynak torçu arasındaki mesafe) etkisini incelemek için yayınlanmış literatürden elde edilen deneysel bir veri seti kullanılmıştır. 5005 Al-Mg alaşımı için penetrasyon derinliği üzerinde çalışılmış ve bu çalışmada genetik algoritma – yapay sinir ağı modelinin eğitimi, testi ve optimizasyonu amacıyla kullanılmıştır. Sonuçlar, modelin çıktıyı oldukça iyi bir doğrulukla % 0.7198'lik ortalama mutlak hatayla tahmin edebileceğini

ve 100.09 sn'lik ihmal edilebilir bir hesaplama süresi ile işlem parametrelerini optimize edebileceğini göstermiştir. Önerilen yaklaşım, operasyonel parametrelerin tahmini ve optimizasyonu için makul doğrulukta çok değişkenli karmaşık bir problemde uygulama için öngörülmüştür. Bayes düzenlemesi yöntemiyle geri yayılımlı ağ kullanılarak eğitilen 4-7-1 ağının en iyi tahmin kabiliyetini gösterdiği bulunmuştur ve optimizasyon sırasında maksimum 3.84 mm penetrasyon derinliği elde edilmiştir.

Prakash çalışmasında [26], girdi değişkenleri olarak gerilim, akım vekaynakl hızını kullanarak düşük karbonlu çeliğin üzerindeki penetrasyon etkisini incelemiştir. Taguchi ortogonal dizisine dayalı deneylerin tasarımında varyans analizi kullanılarak parametrelerin optimum koşulla etkisini belirlenmiştir. Araştırma sonuçlarına göre, düşük karbonlu çelik üzerine kaynak yapıldığında en önemli etkiye sahip olan parametrenin kaynak akımı olduğu ve bunu kaynak gerilimi ve tel hızının izlediği belirtilmiştir. Ayrıca en uygun koşul değerlerinin kaynak akımı için 250A, kaynak gerilimi için 20V ve tel hızı için de 2.2 birim olduğu belirtilmiştir.

Schneider çalışmasında [27], TIG-MIG/MAG hibrit kaynak işleminin kaynak dikiş geometrisi üzerindeki etkisini incelemiştir. Deneyleri gerçekleştirmek için Taguchi metodolojisini (sağlam tasarım yöntemi) kullanan deneysel bir tasarım kullanılmıştır. Deneyler, her biri üç kopya olmak üzere toplam 81 test numunesi olmak üzere 27 deney içeren bir ortogonal matrise göre gerçekleştirilmiştir. Faktörlerin (MIG/MAG koruyucu gaz tipi, MIG/MAG voltajı, MIG/MAG tel beslemesi, TIG gaz akış hızı, TIG'nin elektrik akımı yoğunluğu ve kaynak hızı) her biri üç seviye ile değiştirilmiştir. Penetrasyon, ısıdan etkilenen bölge, dikiş genişliği ve dikiş yüksekliği analiz edilen yanıt değişkenleri olarak belirlenmiştir. Sonuçlar, penetrasyonun tel besleme miktarından, MIG/MAG koruyucu gaz türünden, MIG/MAG voltajından ve kaynak hızından önemli ölçüde etkilendiğini göstermiştir. Isıdan etkilenen bölge, MIG/MAG voltajından, MIG/MAG koruyucu gaz türünden, kaynak hızından ve TIG'in elektrik akımı yoğunluğundan etkilenmiştir. Çalışmada tel beslemesi haricinde tüm faktörlerin genişlik üzerinde etkisi olduğu belirtilmiştir. Dikiş yüksekliği, MIG/MAG tel beslemesinden ve TIG'in elektrik akımı yoğunluğundan önemli ölçüde etkilenmiştir. Prosesin optimizasyonu yapılmış, böylece her çıktı değişkeni için kullanılması

gereken faktörlerin değerleri belirtilmiştir ve optimizasyon kaynak test numuneleri ile de onaylanmıştır.

Sudarshan çalışmasında [28], SPSS yazılımına istatistiksel bir yaklaşım uygulayarak, yumuşak çelik üzerine MIG kaynak işlemini optimize etmiştir. Nihai gerilme mukavemetini incelemek için seçilen parametreler V-alın açıları, kaynak akımı ve kaynak voltajıdır. V açıları olarak 30^0, 45^0 ve 60^0; akım değerleri olarak 80A, 100A ve 110A; ve gerilimdeğerleri olarak da 17V, 19V ve 20V alınmıştır. Akma dayanımı ve nihai çekme dayanımı nihai çekme dayanımının, akım ve gerilimin artmasıyla arttığı gösterilmiştir. Ayrıca, gerilimin 20 Volta kadar yükselmesi ile başlangıçta nihai gerilme mukavemetinin arttığını ve daha sonra gerilim değeri arttıkça azaldığı da belirtilmiştir.

Kanti çalışmasında [29], darbeli gaz metal ark kaynağı işleminde kaynak dikiş geometrisinin tahmini için geri yayılımlı bir sinir ağı modeli sunmuştur. Model deneysel verilere dayandırılmış ve modeli geliştirmek için de giriş parametreleri olarak plakanın kalınlığı, darbe frekansı, tel besleme hızı, tel besleme hızı / ilerleme hızı oranı ve tepe akımı kullanılmıştır. Çıktı parametreleri olarak penetrasyon derinliği ve dışbükeylik indeksi dikkate alınmıştır. Geliştirilen model daha sonra deneysel sonuçlarla karşılaştırılmış ve sinir ağı modelinden elde edilen sonuçların kaynak dikiş geometrisini tahmin etmede doğru olduğu belirtilmiştir.

Sudhakaran çalışmasında [30], penetrasyon derinliğini tahmin etmek ve proses parametrelerini optimize etmek için benzetilmiş tavlama algoritmasını kullanarak bir sinir ağı modeli geliştirilmiştir. Çalışma için seçilen işlem parametreleri kaynak akımı, kaynak hızı, gaz akış hızı ve kaynak tabancası açısıdır. Seçilen çıktı parametresi penetrasyon derinliğidir. Deneyler, 125 çalıştırma ile kesirli faktöriyel kullanılarak deney tasarımına dayalı olarak gerçekleştirilmiştir. Deneysel veriler kullanılarak, ileri beslemeli geri yayılım sinir ağı modeli geliştirilmiş ve Levenberg-Marquardt algoritması kullanılarak eğitilmiştir. 4-15-1 ağına dayalı ağ modelinin penetrasyon derinliğini daha doğru tahmin ettiği bulunmuştur. Optimizasyon yapmak için proses

parametrelerini penetrasyon derinliği ile ilişkilendiren matematiksel bir model de geliştirilmiştir.

Sreeraj çalışmasında [31], düşük karbonlu çelik plakalar üzerinde en iyi kaynağı yapmak için girdi parametreleri olarak kaynak akımı, kaynak hızı, tabanca açısı, temas ucu-çalışma mesafesi kullanmıştır. Ark Kaynağı deneyleri tam çoğaltma tekniği ile merkezi bileşik döndürülebilir tasarıma dayalı olarak gerçekleştirilmiş ve çoklu regresyon yöntemi kullanılarak matematiksel modeller geliştirilmiştir. Geliştirilen modeller yeterlilik ve anlamlılık açısından kontrol edilmiştir. Yapay Sinir Ağı kullanılarak parametreler tahmin edilmiş ve tahmin edilen ile gerçek değerler arasındaki hata yüzdesi hesaplanmıştır. İşlem parametrelerinin dikiş geometrisi üzerindeki doğrudan ve etkileşim etkileri grafik olarak sunulmuştur.

Iqbal çalışmasında [32], yüksek mukavemetli düşük alaşımlı çeliğin gaz tungsten ark (GTA) kaynağında kaynak dikiş geometrisini tahmin etmek için yapay sinir ağı uygulamıştır. Kaynak işlemi parametrelerini kaynak dikiş geometrisi ile ilişkilendirmek için geri yayılım sinir ağı algoritması takip edilmiştir. Kaynak akımı ve kaynak hızının, dikiş geometrisi üzerinde oldukça önemli olduğu vurgulanmıştır. Kaynak voltajının etkisi orta derecede önemliyken, gaz akış hızının etkisinin de önemsiz olduğu belirtilmiştir.

BÖLÜM 3. GELİŞTİRİLEN SANAL KAYNAK SİMÜLATÖRÜ

Sanal simülatörlerin temel amacı, gerçek hayatta öğrenilmesi tehlikeli veya pahalı pratikleri sanal ortamda öğretebilmektir. Kaynakçı eğitimide aynı şekilde tehlikeli ve pahalıya mal olan bir işlemdir. Bu yüzden kaynakçı adaylarının eğitimini sanal kaynak simülatörlerinde başlatmak, hem iş kazası riskini hemde malzeme israfından kaynaklanan eğitim masraflarını azaltmaktadır.

Bu tez çalışmasında geliştirilen kaynak simülatörü, kullanıcılarına gerçek kaynak işlemine yakın hisler uyandıracak şekilde tasarlanmıştır. Kullanıcının el ve kafa hareketleri çeşitli sensörler aracılığıyla yakalanmakta ve 3B sanal ortam bu hareketlere göre güncellemektedir. Geliştirilen kaynak simülatörünün bileşenleri Şekil 3.1.'de gösterilmiştir.

Şekil 3.1. Sanal kaynak simülatörünün bileşenleri

Kullanıcılar sanal ortamı başlarına taktığı HMD(Başa takılan ekran) aracılığıyla görmektedir. Simülatörde HTC firması tarafından üretilen HMD kullanılmaktadır. Bu ürün içerisinde başın oryantasyonu ve konumunu algılayacak sensörler bulunmaktadır. Sensörlerden alınan veriler simülatör yazılımı aracılığıyla takip edilmekte ve HMD tarafından kullanıcıya gösterilen 3B sahne güncellenmektedir.

Simülatör gerçek bir torcu temsil etmesi için HTC Vive yönetici cihazını kullanmaktadır. Bu cihaz kullanıcının el hareketlerini ve elin oryantasyonudaki değişimleri takip etmektedir. Yönetici cihazından gelen verilere göre sanal ortamdaki torcun koordinatı ve oryantasyonu değişmektedir. Kullanıcı, elini hareket ettirdiğinde sahnedeki torcunda aynı yönde hareket ettiğini görmekte ve böylece kaynak işlemini kendisinin yaptığını hissetmektedir.

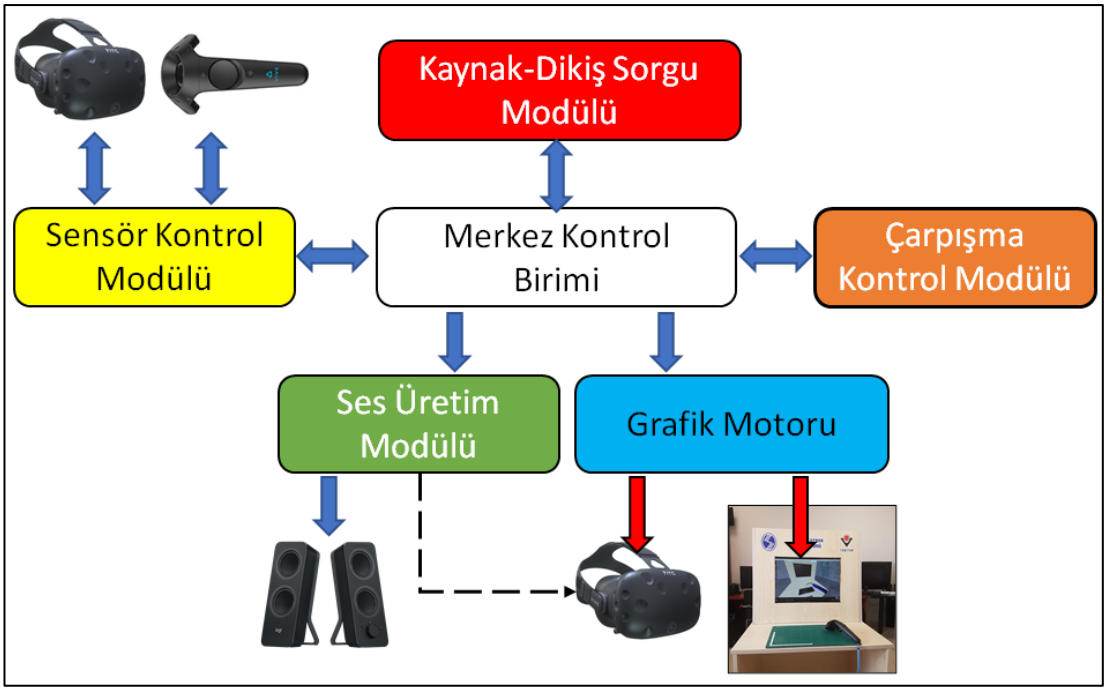
Kullanıcının sanal ve gerçek ortamda birebir aynı modeller üzerinde çalışması gerçeklik hissini arttırmaktadır. Kaynak işlemi için gerçek bir masanın 3B modeli kullanılmıştır. Şekil 3.2.'de kaynak masası gösterilmektedir. Kaynak masası üzerinde 3B(Üç Boyutlu) görüntü verebilen bir monitör bulunmaktadır. Kaynak işlemi gerçekleşirken izleyiciler bu monitör üzerinden kaynakçı adayının performansını takip edebilmektedir.

Gerçek Masa 3B Masa Modeli

Şekil 3.2. Kaynak masası ve 3B modeli

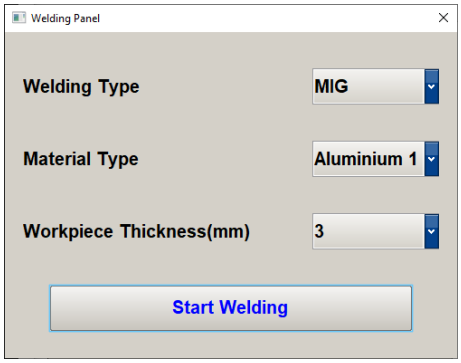
3.1. Simülatörün Mimarisi

Geliştirilen kaynak simülatörünün yazılım mimarisi Şekil 3.3.'deki gibidir. Merkez kontrol birimi simülatörün yönetiminden sorumludur. Simülatör çalıştırıldığında kontrolü merkez kontrol birimi almaktadır. Yazılım içerisindeki sensör kontrol modülü, kaynak dikişi sorgu modülü ve çarpışma modüllerini kullanarak 3B sahnenin oluşturulması için gereken bilgileri grafik motoru ve ses üretim modülüne aktarmaktadır. Grafik motoru ve kaynak sorgu modülü sırasıyla dördüncü ve beşinci bölümlerde detaylı bir şekilde anlatılmıştır.



Şekil 3.3. Geliştirilen sanal kaynak simülatörünün yazılım mimarisi

Simülatör çalıştırıldığında merkez kontrol birimi yapılacak kaynak işleminin özelliklerinin belirlenmesi için kullanıcıya bir giriş formu sunmaktadır. Şekil 3.4.'de giriş formu gösterilmektedir. Form içerisinde ilk olarak kaynak türü seçilmektedir. İkinci seçenekte birleştirilecek olan materyal türü verilmektedir. Son olarak birleştirilecek olan materyalin kalınlığı belirlenmektedir. Kullanıcı "Start Welding" butonuna bastığında kontrol birimi simülatörü çalıştırmaktadır.



Şekil 3.4. Simülatör giriş paneli

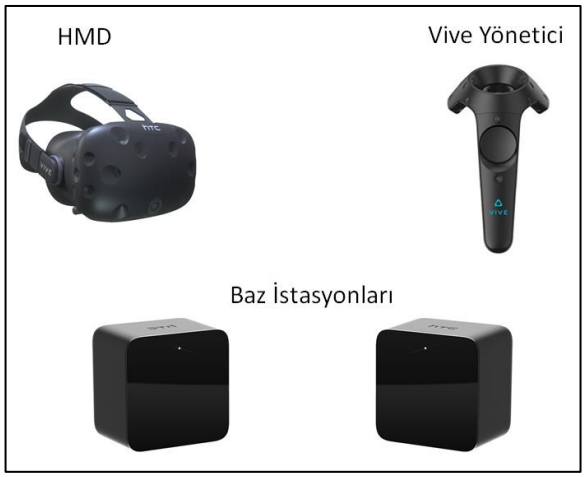
3.2. Sensör Kontrol Birimi

Sanal kaynak simülatöründe kullanıcının el ve baş hareketlerini takip edebilmek için HTC Vive sanal gerçeklik platformu kullanılmıştır. Bu platform bir adet HMD, iki adet yönetici ve sanal ortamın sınırları ile kullanıcının konumunu belirleyen iki adet kızıl ötesi baz istasyondan oluşmaktadır. HMD, her bir göz için 1080x1200 çözünürlüğe sahip iki adet OLED(organic light-emitting diode) panel barındırmaktadır. Simülatörün 3B ortamı grafik motoru aracılığıyla bu ekranlara yansıtılmaktadır. Başa geçen ekranın dış kabında bir düzine kızıl ötesi sensör bulunmaktadır. Bu sensörler kızıl ötesi istasyonlar tarafından başın konumunu belirlemek için kullanılmaktadır. HMD ayrıca birer adet ivmeölçer, cayroskop ve mesafe sensörü barındırmaktadır. Bu sensörler aracılığıylada kullanıcın başını döndürme açısı da tesbit edilebilmektedir.

Vive yöneticileri, kullanıcının simülatöre komut gönderebilmesi için birden fazla giriş aracına sahiptir. İki aşamalı bir tetik ve sıkma butonları bulunmaktadır. Ayrıca kullanıcıya titreşim ile geri besleme yapabilmektedir. Kaynak işlemi başladığında bu titreşim mekanizması simülatör tarafından kullanılmaktadır. Yöneticilerin üzerinde 24 adet kızıl ötesi sensör bulunmaktadır. Bu sensörlerin yaydığı ışınlar kızıl ötesi istansyonlar tarafından yakalanmakta ve yöneticinin ortamdaki konumu belirlenmektedir.

Kızıl ötesi baz istasyonları gerçek ortamda sanal ortam için kullanılacak bir izleme ortamı oluşturmak için kullanılmaktadır. Şekil 3.5.'de HTC vive setine ait araçlar

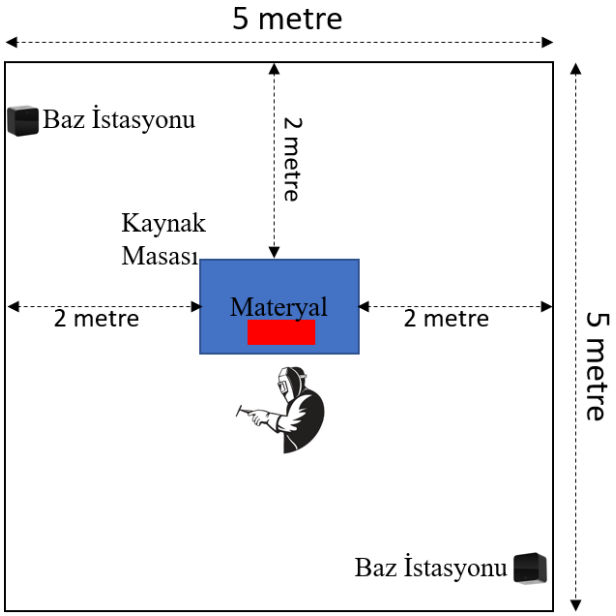
gösterilmektedir. Baz istasyonları saniyede 60 adet kızıl ötesi sinyal yaymaktadır. Yayılan sinyaller Vive yöneticisi ve HMD'de bulunan sensörler tarafından yakalanmaktadır. Böylece Vive yöneticilerin ve HMD cihazının izleme ortamındaki konumu hesaplanmaktadır.



Şekil 3.5. HTC Vive araçları

Sensörlerin verimli çalışabilmesi için kaynak işlemi 5 metre x 5 metre bir alan içerisinde gerçekleştirilmektedir. Şekil 3.6.'da bu alanın planı verilmiştir. Baz iztasyonları şekilde de görüldüğü gibi kaynak alanının uç köşelerine yerden enaz 2 metre yüksekliğe yerleştirilmişlerdir. İstasyonların bakış yönleri kaynak alanının merkezine doğru çevrilmiştir. Bu sayede htc vive yöneticisi ve HMD cihazları ile daha iyi iletişim gerçekleştirilmektedir. Kaynak masasının da konumu şekilde görüldüğü gibi merkezdedir. Kaynak işlemi için kullanılacak olan materyalde kaynak masası üzerine yerleştirilmektedir. Böylece sanal ortam ile gerçek ortamın ölçüleri birbirini tutmuş olacaktır.

Sensör kontrol biriminin görevi HTC Vive araçlarından gelen hareket ve oryantasyon bilgileri yakalayıp merkezi kontrol birimine göndermektir. Sensör kontrol biriminin tasarımında farklı sensörlerde kullanılabileceği göz önünde bulundurulmuş ve yazılım mimarisi gerekli soyutlama mekanizmaları üzerinden geliştirilmiştir.

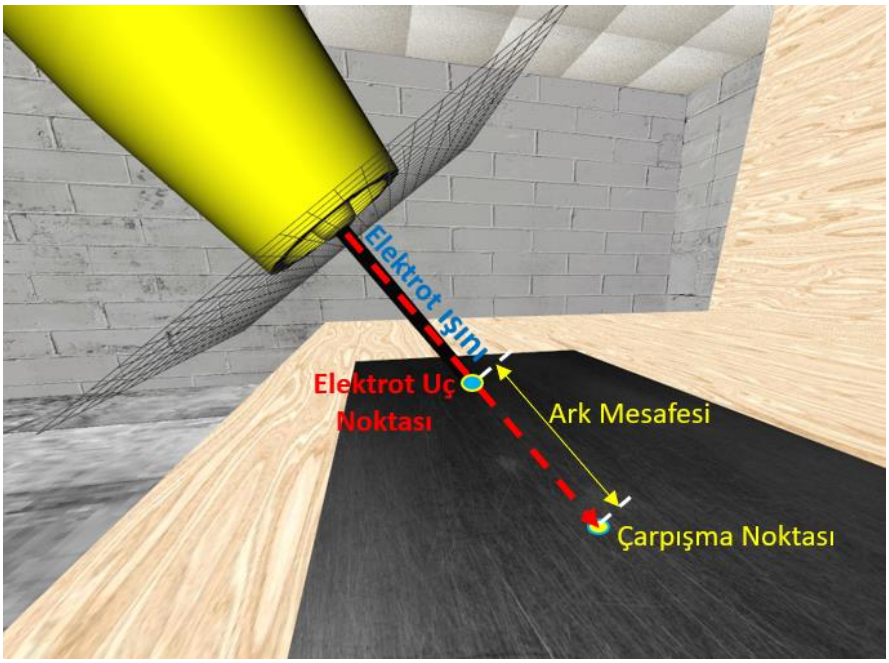


Şekil 3.6. Sanal kaynak işleminin gerçekleştirildiği ortam.

3.3. Çarpışma Kontrol Birimi

3B sahne içerisinde hareket halinde olan modellerin başka modeller veya sahneyi oluşturan sabit şekillerle temas ettiği veya aralarındaki mesafenin tespit edilebilmesi için çarpışma testleri kullanılmaktadır. Kaynak simülatörü içerisinde sadece torç hareket halinde olacağı için çarpışma testleri de torç ile sabit sahne arasında gerçekleşecektir. Çarpışma kontrol biriminin görevi simülatör çalışırken, her bir çizim öncesinde gerekli çarpışma testlerinin yapılmasını sağlamak ve sonuçları merkezi kontrol birimine göndermektir.

Torcun ucundan elektrot adı verilen silindir şeklinde bir materyal çıkmaktadır. Kaynak işleminin başlamasını elektrodun kaynak materyali ile arasındaki mesafe belirlemektedir. Elektrot ile kaynak materyali arasındaki mesafeye ark mesafesi adı verilmektedir. Ark mesafesi hesaplanırken torcun ucundan Şekil 3.7.'deki gibi elektrot ile aynı yönde temsili bir ışın çıkartılmaktadır. Bu ışın ile materyal arasında bir çarpışma testi uygulanır. Çarpışma noktası hesaplandıktan sonra elektrotun uç noktası ile çarpışma noktası arasındaki mesafe hesaplanmaktadır. Hesaplama yapıldıktan sonra sonuç merkezi kontrol birimine gönderilmektedir. Merkezi kontrol birimi de kaynak işleminin başlayıp başlamayacağına karar verip gerekli işlemleri yapacaktır.



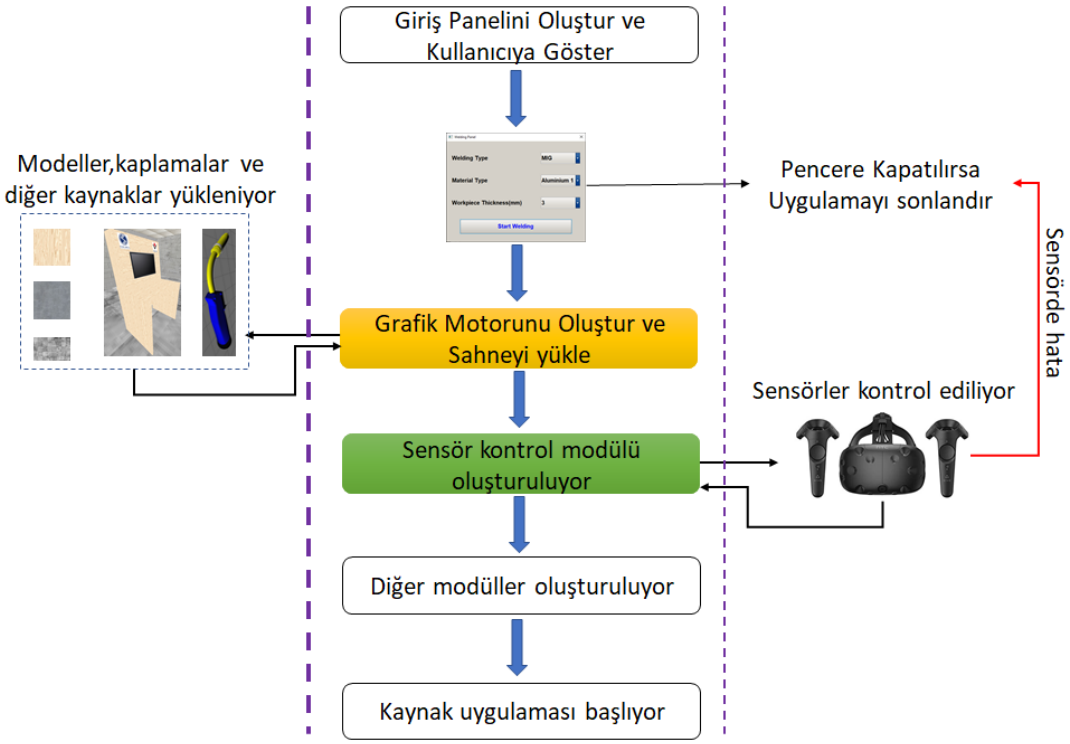
Şekil 3.7. Ark mesafesinin hesaplanması

3.4. Ses Üretim Modülü

Sanal gerçeklik uygulamalarında amaç kullanıcılarda gerçeklik hissi uyandırmaktır. Bunun için gerçek hayattaki seslerinde benzetilmesi gerekmektedir. Ses üretim modülünün görevi kaynak işlemi başladığında kullanıcılara gerçek bir kaynak işleminde ortaya çıkan seslerin benzerlerini üretmektir. Sanal ses üretimi için OpenAL (Open Audio Library) kütüphanesi kullanılmaktadır. OpenAL sayesinde simülatör üç boyutlu ses üretebilmekte ve üretilen sesler kullanıcının kulaklıklarının yanı sıra kaynak masası üzerinde bulunan harici hoparlörlerdende dışarı verilmektedir. Ses üretim modülü emirleri merkezi kontrol biriminden almaktadır. Merkezi kontrol birimi kaynak işlemi başldığında ses üretim modülüne kullanılacak ses efeklerini ve sürelerini belirtmektedir.

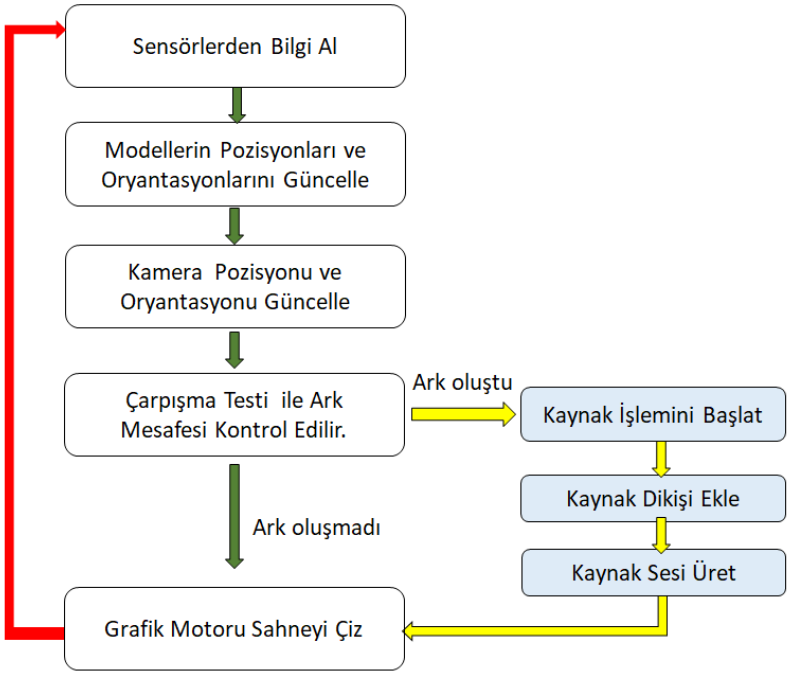
3.5. Merkezi Kontrol Birimi

Merkez kontrol birimi sanal kaynak simülatörünün beyni olarak görev yapmaktadır. Simülatör yazılımı içerisinde bütün modülleri kontrol edip aralarındaki iletişimi sağlamaktadır. Simülatör çalıştırıldığında merkez kontrol birimi, Şekil 3.4.'de görünen uygulama ayar panelini kullanıcının karşısına çıkarmaktadır. Kullanıcı uygulamayı başlattığında seçilen ayarlara göre merkez kontrol birimi sahnenin oluşturulması için grafik motorunu kullanmaktadır. Öncelikle sahneyi oluşturan modeller sisteme yüklenir,ardından sensör modülü çalıştırılarak htc araçlarının doğru bir şekilde çalışıp çalışmadığı kontrol edilmektedir. Kaynak işlemi başlamadan önce diğer modüllerde merkez kontrol birimi tarafından oluşturulup kullanıma hazır hale getirilmektedir. Şekil 3.8.'de sanal kaynak simülatörünün başlatılması sırasında gerçekleşen işlemlerin akış diyağramı gösterilmiştir. Sensörlerde oluşacak hatalar sistemin bir hata mesajı vererek kapatılmasına neden olacaktır. Kaynak uygulaması başlatıldığında merkezi kontrol biriminin görevi devam etmektedir.



Şekil 3.8. Sanal kaynak simülatörünün başlatılması sırasında gerçekleşen işlem aşamaları

Kaynak uygulaması başladığında grafik motoru sonsuz bir çizim döngüsüne girer. Bu döngünün her dönüşünde 3B sanal ortamın anlık bir görüntüsünü oluşturmaktadır. Döngü yeteri kadar hızlı döner ise sanal ortamın görüntüsü akıcı bir şekilde oluşturulmakta ve ilgili ekranlara gönderilebilmektedir. Simülatörün gerçeklik hissini arttırmak için uygulamanın gerçek zamanlı çalıştırılabilmesi gerekir. Gerçek zamanlı çizim uygulamalarının saniyede 60 veya daha fazla görüntü elde etmesi gerekmektedir [33]. Sanal ortamın anlık görüntüsünü elde ederken sensörlerden gelen veriler ve çarpışma testlerinin sonuçları da kullanılmaktadır. Geliştirilen kaynak simülatöründe gerçek zamanlı bir çizim elde edebilmek için çarpışma testi, grafik motoru ve kaynak sorgu modelinde optimizasyonlar yapılmıştır. Sonuçlar bölümünde simülatörün performans sonuçlarına ait tablolar bulunmaktadır.



Şekil 3.9. Sanal kaynak simülatörünün sahne oluşturma döngüsü

Şekil 3.9.'da kaynak uygulaması başladığında çalıştırılan sanal ortamı oluşturma döngüsü verilmiştir. Bu döngü simülatör uygulaması kullanıcı tarafından kapatılana kadar devam edecektir. Döngünün her bir turu sonunda 3B sanal ortamın anlık görüntüsü oluşturulacak ve ekrana gönderilecektir. Döngüde ilk olarak kullanıcının elindeki ve başındaki sensörlerin konum ve oryantasyon bilgileri alınmaktadır.

Ardından bu bilgiler kullanılarak sahnedeki modellerin koordinatları güncellenmektedir. Simülatörde birden fazla model kullanılabilmektedir. Fakat genelde kaynak işleminde kullanılacak olan torcun konumu ve oryantasyonu güncellenmektedir. Daha sonra kullanıcının sahneyi izlediği sanal kameranın koordinatı ve oryantasyonu kullanıcının başına takılan HMD cihazındaki sensörlerden alınan veriye göre güncellenmektedir. Kullanıcı bakış açısını değiştirdiğinde sanal ortamdaki baktığı noktada buna göre değişecektir. Bir sonraki adımda sahnedeki torç ile materyal arasında çarpışma testi uygulanmaktadır. Bu test sonucunda elektrodun ucu ile materyal arasındaki mesafe hesaplanmaktadır. Bu mesafe ark oluşturmaya yetecek bir boyutta ise kaynak işlemi başlatılmaktadır. Ark mesafesi uygulama çalıştırılmadan önce kullanıcı tarafından değiştirilebilmektedir. Kaynak işlemi başladığında önce kaynak işleminden çıkacak ses üretilmektedir arından kaynak dikişi üretilip sahneye eklenmektedir. Kaynak dikişinin üretilmesi için gereken işlemler beşinci bölümde detaylı olarak anlatılmaktadır. Kaynak dikişine ait modeller sahneye eklendikten sonra grafik motoruna sahneyi çizmesi komutu verilir. Sahne çizimi gerçekleştikten sonra döngü tekrar başa döner. Döngünün bitmesi için kullanıcının uygulamayı kapatması gerekmektedir.

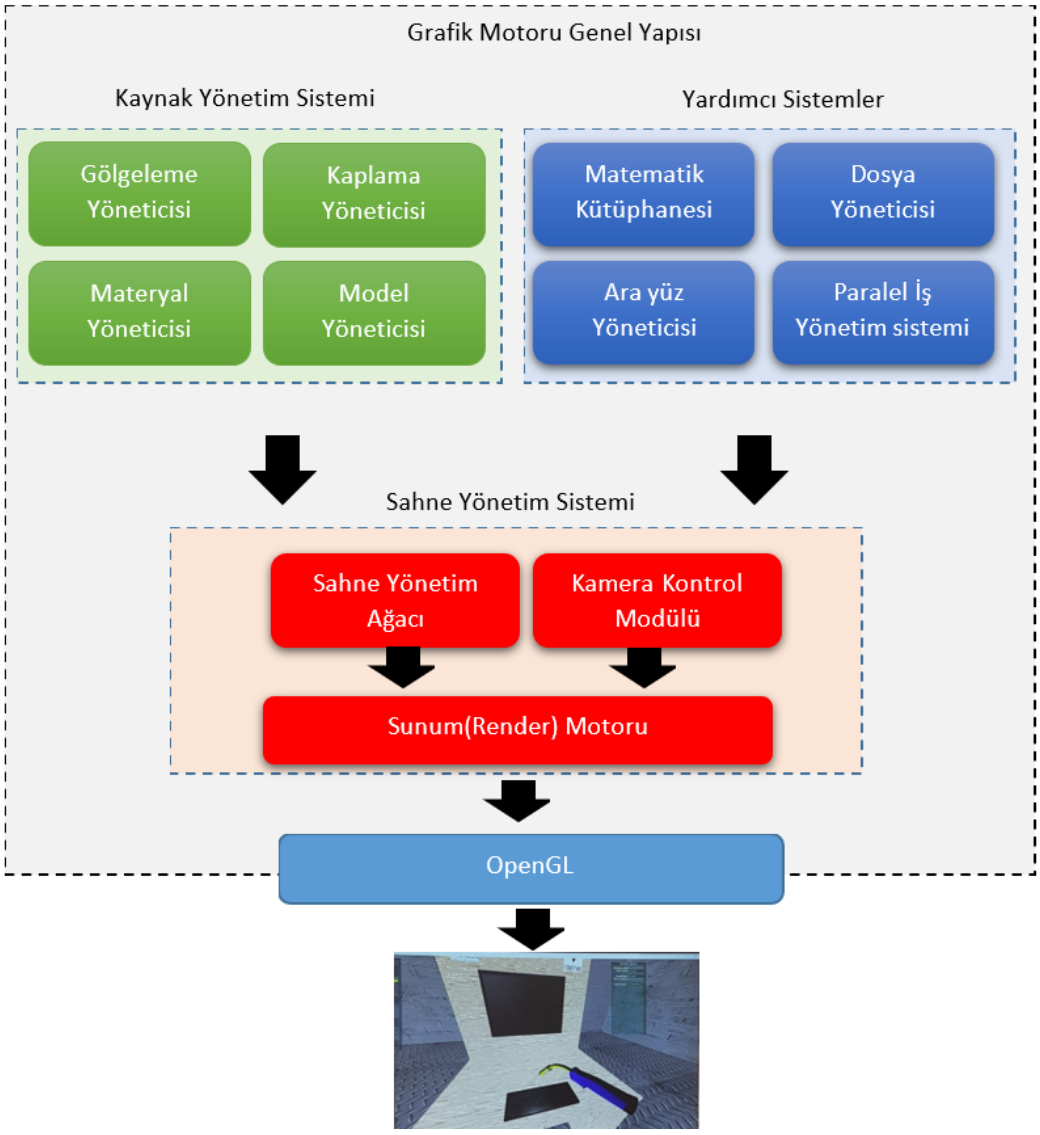
BÖLÜM 4. GRAFİK MOTORU

Son yıllarda 3B(üç-boyutlu) grafik uygulamaları büyük popülerlik kazanmıştır. Özellikle doksanlı yıllar içerisinde kullanımı artan bu uygulamalar geçen yıllar içerisinde bilgisayarların iş gücünün artması ile her geçen gün daha etkileyici sonuçlar vermeye başlamıştır. 3B uygulamaların daha gerçekçi görüntüler elde etmesi ve daha etkileyici efektler yapılması arka plandaki yazılım organizasyonunu daha karmaşık hale getirmektedir. Yazılımın karmaşık hale gelmesi 3B uygulamaların geliştirilme süresini de arttırmaktadır. Bu sebeple programcıların daha kolay ve daha hızlı 3B uygulama geliştirebilmesi için grafik motoru adı verilen kütüphaneler geliştirilmektedir [34]. Bu kütüphaneler aracılığıyla uygulama geliştiren programcılar 3B uygulama için gerekli olan birçok karmaşık işlemden arındırılmış olmakta ve çok daha hızlı uygulama geliştirebilmektedir.

Bu tez çalışması içerisinde de ilk olarak bir grafik motoru tasarlanmıştır. Daha sonra grafik motorunun kütüphaneleri kullanılarak sanal kaynak simülatörünün görüntüleri elde edilmiştir. Grafik motoru kaynak simülatörünün en önemli modüllerinden birisidir. Şekil 4.1.'de grafik motorunu oluşturan birimlerin genel yapısı gösterilmektedir.

Grafik motoru çizim işlemleri için OpenGL (Open Graphics Library) kütüphanesini kullanmaktadır. Programlama dili olarak C++ dili seçilmiştir. Grafik motoru, kaynak yönetim sistemi, yardımcı sistemler ve sahne yönetim sistemi olmak üzere üç temel kısımdan oluşmaktadır. Kaynak yönetim sistemi grafik motorunun kullanacağı nesnelerin oluşturulması ve sistemin kullanacağı hafıza alanının en iyi performansı verecek şekilde yönetilmesini sağlayamak için tasarlanmıştır. Yardımcı sistemler grafik motorunun kullanacağı kaynakların sisteme yüklenmesinden, yapılacak olan işlemlerin merkezi işlemci çekirdeklerine paylaştırılmasına kadar çeşitli yardımcı görevleri üstlenmektedir. Sahne yönetim sistemi grafik motorunun oluşturacağı 3B

ortamın yönetilmesi ve çizilmesi ile görevlidir. Grafik modülleri yaptıkları işlemlerden sahne yönetim sistemini haberdar etmektedir. Böylece sahneyi oluşturan her öğe sahne yönetim sistemi tarafından dinamik olarak yönetilmektedir.



Şekil 4.1. Grafik motoru genel yapısı

4.1. OpenGL Kütüphanesi

OpenGL, bir API (Application programming Interface) yani bir yazılım programlama arayüzüdür. Grafik donanımı ile iletişime geçmek için kullanılan bir yazılım kütüphanesidir. OpenGL kütüphanesi 500'ün üzerinde farklı komut barındırmaktadır [35]. Bu komutlar etkileşimli üç-boyutlu bilgisayar grafik uygulamalarının

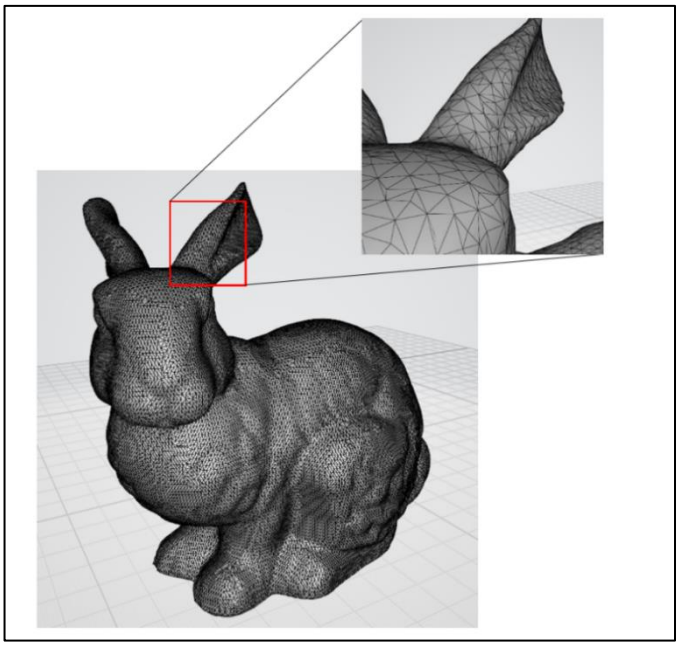
geliştirilmesini sağlayacak nesneler, resimler ve çeşitli işlemleri tanımlamak için kullanılmaktadır. OpenGL donanımdan bağımsız olarak geliştirilmiştir. Sadece yapılacak işlemlerin isimlerini ve işlem sonuçlarını belirlemektedir. Kullanılacağı donanıma göre bu işlemlerin içeriği değişebilmektedir. Bu sayede OpenGL çok farklı platformlarda sorunsuzca çalışabilmektedir.

4.1.1. OpenGL kullanılmasının nedeni

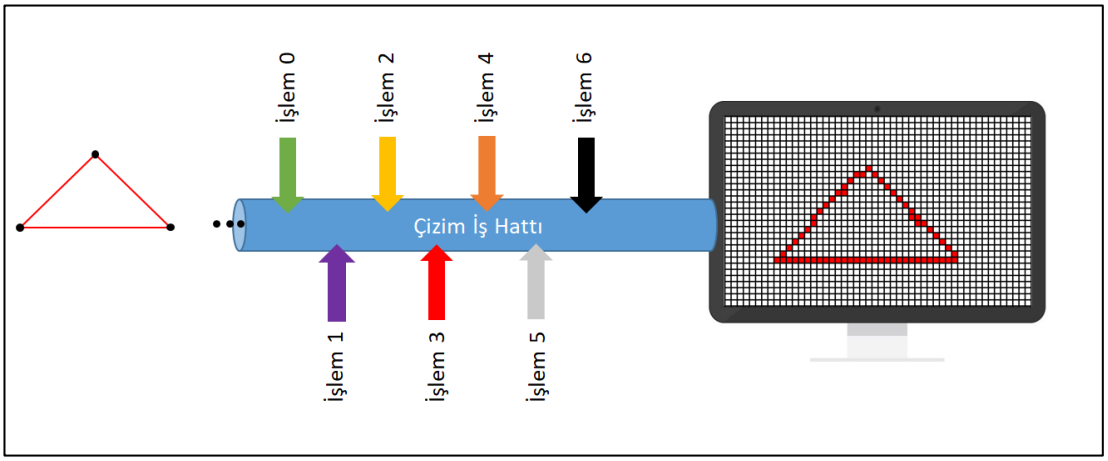
OpenGL kütüphanesi grafik donanımına erişimde kullanılan tek kütüphane değildir. Microsoft firmasının geliştirmiş olduğu DirectX kütüphanesi de aynı amaçla kullanılmaktadır. Fakat DirectX sadece Microsoft firmasına ait platformlarda çalışmaktadır. Sanal kaynak simülatörünün platformdan bağımsız olabilmesi veya başka platformlara kolay adapte edilebilmesi için OpenGL grafik kütüphanesi seçilmiştir.

4.1.2. OpenGL çizim iş hattı

Günümüzde 3B ortamlar birçok üçgenin birleştirilmesi ile oluşturulmaktadır [36]. Şekil 4.2.'de 3B tavşan modelinin üçgenlerden oluşturulmuş hali gösterilmiştir. OpenGL kütüphanesinin temel görevi nokta bilgilerini programcıdan aldığı üçgenleri, grafik donanımına çizdirmektir. Bu işlemin gerçekleşmesi sırasında kullanıcıdan gelen veriler sırayla çeşitli işlemlere tabi tutulmaktadır. Noktalara ait verilere uygulanan işlemlerin bütününe OpenGL çizim iş hattı denilmektedir. Şekil 4.3.'de çizim iş hattının temsili görüntüsü verilmektedir. Çizim iş hattındaki işlemler uygulamadan uygulamaya farklılık gösterebilmektedir. Bu işlemler gölgeleme yöneticisi bölümünde daha detaylı açıklanmıştır.



Şekil 4.2. 3B modellerin üçgenler ile gösterimi



Şekil 4.3. OpenGL çizim iş hattının temsili gösterimi

4.2. Kaynak Yönetim Sistemi

Kaynak yönetim sistemi içindeki birimlerin görevi grafik motorunun ihtiyaç duyduğu kaynakların, hafıza üzerinde yerleşimini ve organizasyonunu yönetmektir. Bu birim, gölgelendirme yöneticisi, kaplama yöneticisi, materyal yöneticisi ve model yöneticisi olmak üzere dört temel modülden oluşmaktadır. Sistem içerisindeki modüller ihtiyaç duyduklarında yardımcı sistemlere ait modülleri de kullanabilmektedir. Aynı şekilde

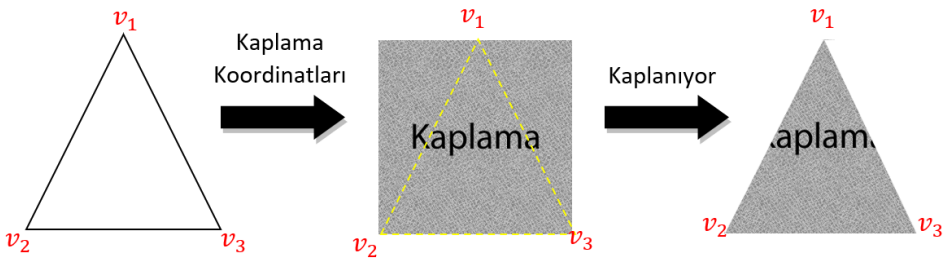
sahne yönetim modülü de 3B sanal ortamı oluşturabilmek için kaynak yönetim sisteminin modüllerini kullanmaktadır.

4.2.1. Kaplama yöneticisi

Kaplama 3B grafik uygulamalarında yüzeylerin üzerine yerleştirilen (giydirilen) resimlerdir. Kaplamalar gerçek dünyadan alınan resimler olabileceği gibi çeşitli grafik uygulamaları ile yazılımsal olarak da elde edilebilmektedir. Kaplama yöneticisinin görevi resimleri sabit diskten alıp OpenGL tarafından kullanılabilecek biçimde hafıza ünitesine yerleştirmek ve istenildiğinde sahne yönetim sistemine resmi devretmektir. Bu işlem gerçekleştirilirken resim bilgilerinin hafızada birden fazla defa kopyalanmasına engel olunmaktadır.

4.2.1.1. Kaplama giydirme

Kaplama giydirme işlemi 3B yüzeyler üzerine gerçek dünyadan alınmış veya dijital olarak elde edilmiş resimlerin yerleştirilmesidir. Şekil 4.4.'de bir üçgen yüzeyine kaplama giydirme işleminin gösterimi verilmektedir.

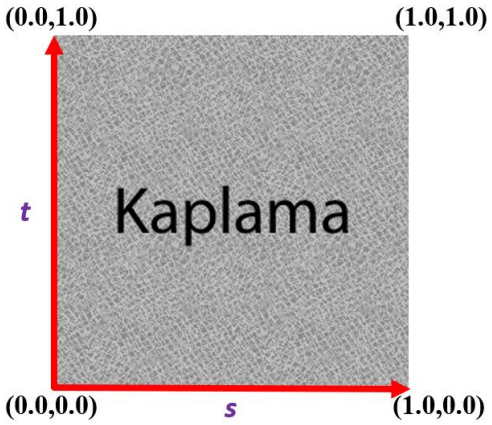


v_1 Kaplama koordinatları (s,t) = (0.5,1.0);
v_2 Kaplama koordinatları (s,t) = (0.0,0.0);
v_3 Kaplama koordinatları (s,t) = (1.0,0.0);

Şekil 4.4. Bir üçgene kaplama giydirme işleminin gösterimi

Kaplama giydirme işlemi gerçekleştirilirken yüzeyin her bir noktasının kaplamanın hangi noktasına denk geleceğinin belirlenmesi gerekir. Şekil 4.5.'de OpenGL kütüphanesinin kullandığı kaplama koordinat sistemi gösterilmektedir. Kaplama

koordinat sisteminin t ve s adında iki ekseni bulunmaktadır ve orijin noktası resmin sol alt köşesinden başlamaktadır.



Şekil 4.5. OpenGL Kaplama Koordinat Sistemi

Çizilecek olan yüzey OpenGL iş hattının sonunda iki boyutlu projeksiyon düzlemine yerleştirilmektedir. Bu aşamada kaplamanın yüzey üzerine düşen renk değerleri ilgili pikseller üzerine yerleştirilir. Bu işlemin adımları Şekil 4.4.'de gösterilmiştir.

4.2.1.2. Kaplama sınıfı (Texture)

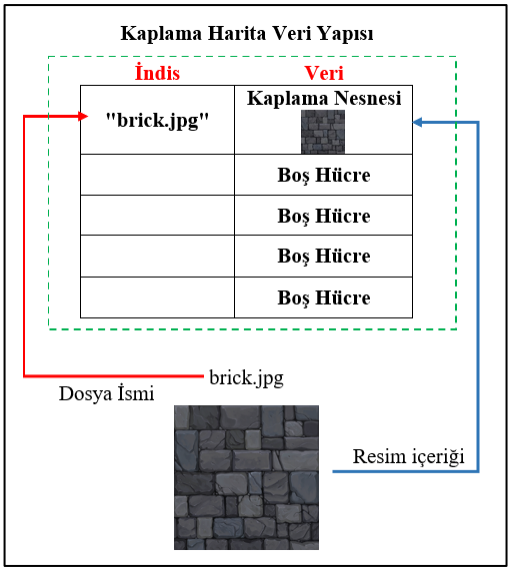
Her bir kaplamayı temsil etmesi için Texture isimli bir sınıf oluşturulmuştur. Sisteme yüklenen her bir kaplama için bu sınıftan bir nesne oluşturulmaktadır. Sınıf içerisinde kaplamanın çizim anında uygulanması için gerekli yapılar ve metotlar geliştirilmiştir. Bu sınıf, kaplama yönetim sınıfına ve OpenGL kütüphanesine uygun olacak şekilde dizayn edilmiştir.

4.2.1.3.Kaplama yönetim sınıfı (TextureManager)

Kaplama resimlerinin sisteme yüklenmesinden kaplama yönetim sınıfı sorumludur. Bu sınıf tekil (Singleton) tasarım deseni kullanılarak tasarlanmıştır. Sanal kaynak simülatörünün çalışma süresince bu sınıftan sadece bir tane nesne oluşturulabilmektedir. Sınıfın tekil olarak tasarlanmasının temel sebebi yazılım

sistemin her noktasından aynı nesneye erişilmesi ve yüklenecek kaplamaların tek bir nesne tarafından kontrol edilmesidir.

Kaplama yönetim sistemi her bir kaplama için kaplama sınıfından bir nesne oluşturmakta ve bu nesneler bir C++ harita (map) veri yapısının içerisinde saklanmaktadır. Harita veri yapısındaki elemanlara erişmek için indis olarak yüklenecek olan resmin ismi kullanılmaktadır. Veri yapısının hücrelerinde ise resim içeriğini barındırması için oluşturulan bir kaplama nesnesinin adresi bulunacaktır. Kaplama harita veri yapısının temel yapısı Şekil 4.6.'da gösterilmektedir.

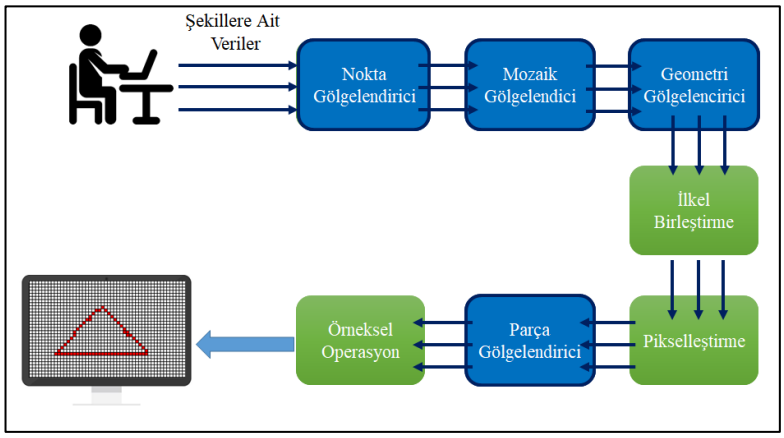


Şekil 4.6. Kaplama harita veri yapısının temel yapısı

Programcı tarafından yeni bir kaplamanın yüklenmesi halinde öncelikle resmin daha önceden sisteme yüklenip yüklenmediği kaplama yönetim nesnesi tarafından kontrol edilmektedir. Bunun için kaplama nesnelerinin saklandığı harita veri yapısı kontrol edilmektedir. Eğer aynı resim kullanılarak daha önceden bir kaplama oluşturulmuş ise ilgili kaplama nesnesi getirilmektedir. Aksi durumda yeni bir kaplama nesnesi oluşturulup önce haritaya eklenmekte ardında programcıya döndürülmektedir.

4.2.2. Gölgelendirici yöneticisi

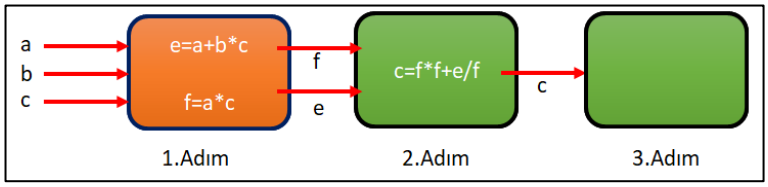
OpenGL çizim iş hattı içerisinde çizilecek şekle ait noktalar ardışıl olarak birçok işlemden geçmektedir. Bu işlemlerin bazıları kullanıcılar tarafından programlanabilir iken bazıları da grafik kartı üreticileri tarafından sabit olarak tasarlanmaktadır. Çizim iş hattındaki programlanabilir işlemlere gölgelendirici (shader) adı verilmektedir. OpenGL çizim iş hattındaki sabit ve programlanabilir iş parçaları işlem sırasına göre Şekil 4.7.'de gösterilmektedir. Buna göre çizilecek şekil için kullanıcıdan alınan nokta verileri öncelikle nokta, ardından mozaik ve geometri gölgelendiricilerine gönderilmektedir. Buradan çıkan sonuçlar sırasıyla ilkel birleştirme ve pikselleştirme birimlerine gönderilmektedir. Bu iki birim kullanıcılar tarafından programlanamaz. Pikselleştirme aşamasından elde edilen veriler programlanabilir olan parça gölgelendirici aşamasına verilmektedir. Parça gölgelendiricisi ekrana çıkarılacak olan renk değerlerini sabit bir iş parçası olan örneksel operasyona devreder. Buradan çıkan sonuçlar da monitöre gönderilecek görüntünün oluşmasını sağlamaktadır.



Şekil 4.7. OpenGL çizim iş hattındaki sabit ve programlanabilir iş parçalarının diyagramı

Çizim iş hattındaki her bir iş parçası kendisinden önceki iş hattından aldığı verileri işleyerek bir sonraki iş parçasına devretmekle görevlidir. Şekil 4.8.'de bu işleme bir örnek verilmiştir. Örnekte kullanıcıdan gelen veriler ilk olarak 1.adım tarafından alınmaktadır. Bu iş parçası gelen verilere çeşitli işlemler yapıp yeni sonuçlar elde etmektedir. Elde edilen sonuçlar bir sonraki adıma gönderilmektedir. 2.adım kendisine 1.adımdan gelen veriler üzerine işlemler yapıp yeni bir veri elde etmekte ve bu veriyi

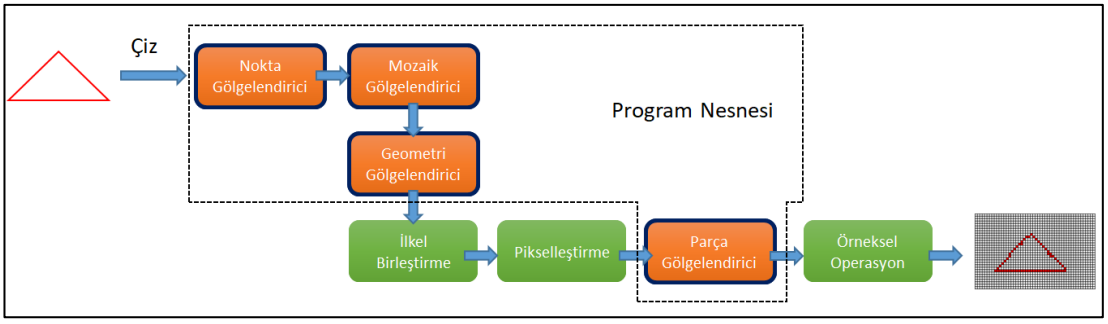
3.adıma göndermektedir. 3.adım da aynı şekilde gelen veriye işlemler gerçekleştirip sonucu bir sonraki adıma gönderebilir.



Şekil 4.8. İş parçaları arasındaki veri iletişimi

4.2.2.1. Program nesneleri

Program nesneleri OpenGL çizim iş hattının programlamabilir kısımlarını temsil etmektedir. Her bir çizim çağrısı yapıldığında OpenGL aktif olan Program nesnesini çizim iş hattı olarak kullanmaktadır. Şekil 4.9.'da program nesnesi ile çizim iş hattı arasındaki ilişki gösterilmektedir.

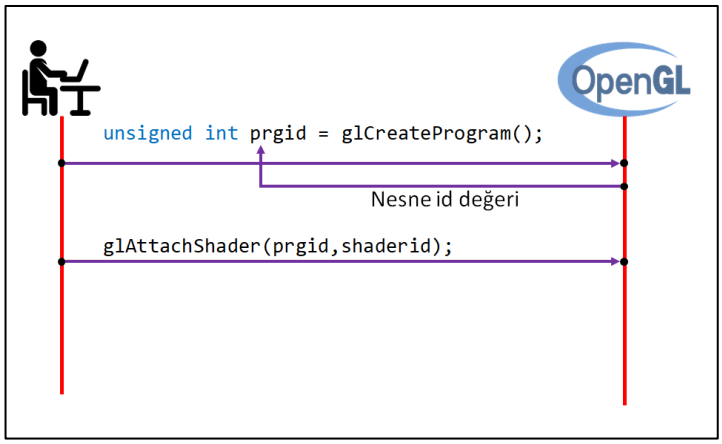


Şekil 4.9. Program nesnesi oluşturan gölgelendiriciler

Program nesnelerinin oluşturulması ve çizim iş hattına yüklenmesi programcıların görevidir. Programcı bir program nesnesini aktif ettikten sonra OpenGL kendisine gelen bütün noktaları bu programa ait gölgelendiricilere yollamaktadır. OpenGL uygulamalarında birden fazla program nesnesi kullanılabilmektedir. Bir program nesnesi aktif edildikten sonra başka bir program aktif edilene kadar çizim iş hattı olarak görev yapmaktadır.

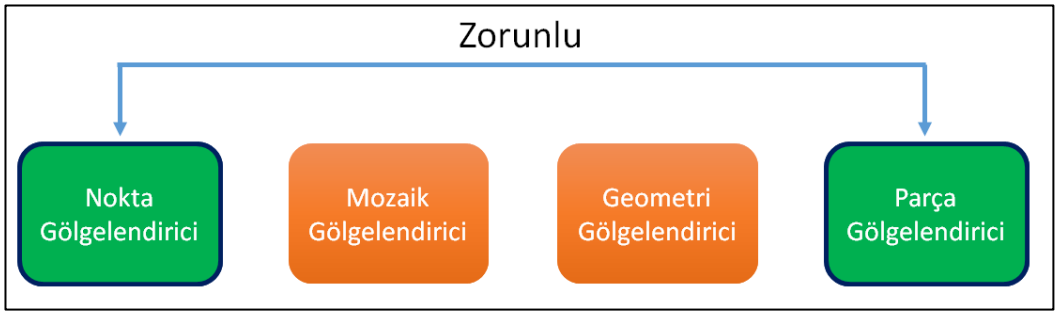
Program nesnelerini oluşturmak için programcının OpenGL kütüphanesine istekte bulunması gerekmektedir. Bu işlem Şekil 4.10.'da gösterilmektedir. Öncelikle

"glCreateProgram" fonksiyonu çağrılarak OpenGL'den bir program nesnesi oluşturulması istenmektedir. Buna karşılık olarak OpenGL oluşturduğu nesnesinin id değerini programcıya getirmektedir. Ardından bu id değeri kullanılarak program nesnesi üzerine bir gölgelendirici program eklenmiştir.



Şekil 4.10. Program nesnesinin OpenGL tarafından oluşturulması

Program nesnesi OpenGL iş hattındaki programlanabilir dört adımı temsil etmektedir. Bu adımların programcı tarafından hazırlanması ve program nesnesine bağlanması gerekir. Şekil 4.11.'de programlanabilir adımlar gösterilmektedir. Bir program nesnesine nokta gölgelendirici ve parça gölgelendiricinin yüklenmesi zorunludur. Diğer iki adım göz ardı edilebilir.



Şekil 4.11. Bir program nesnesine yüklenmesi zorunlu olan ve olmayan gölgelendiriciler

Grafik motoru içerisinde OpenGL program nesnelerini temsil etmek için ShaderProgram isimli bir sınıf oluşturulmuştur. ShaderProgram sınıfının görevi kullanılacak program nesnelerinin, grafik motoru kullanıcıları tarafından daha kolay

kontrol edilebilecek bir ünite içerisine yerleştirmektir. Böylece kullanıcılar program nesnelerini çok daha kolay inşa edip kullanabilmektedir. ShaderProgram sınıfının prototipi Şekil 4.12.'de gösterilmektedir.

ShaderProgram sınıfı loadShaderFromFile metodu aracılığıyla gölgelendirici programlarına ait kaynak kod dosyalarını okuyup derledikten sonra program nesnesine eklemektedir. Gölgelendiricilerin türüne göre saklandıkları kaynak kod dosyalarının uzantıları da değişmektedir. Uzantı çeşitleri Tablo 3.1.'de gösterilmiştir. Grafik motoru mozaik gölgelendiricisini kullanmamaktadır. Dosya uzantıları gölgelendiricileri birbirinden ayırabilmek için kullanılmıştır.

Tablo 3.1. Gölgelendiricilerin saklandığı kaynak kod dosyalarının uzantıları

Gölgelendirici Türü	Dosya Uzantısı
Nokta Gölgelendiricisi	.vert
Geometri Gölgelendirici	.geo
Parça Gölgelendiricisi	.frag

```
class ShaderProgram
{
public:
	ShaderProgram(const string& programName);

	void   loadShaderFromFile( GLenum shaderType,
	                           const string& fileName);
	void   linkShaderProgram();
	int    getProgramId() const;
	string getProgramName()     const;
	void   use();
	void   unUse();

	void   bindAllAttributes();
	void   bindAtrributes(unsigned int slot,
	                      string attributeName);
	void   addUniformVariable(string variableName);

	void   sendUniformMatrix4fv(string variableName,
	                            float* matrix);
	void   sendUniformMatrix3fv(string variableName,
	                            float* matrix);
	void   sendUniformVec3(string variableName,
	                            float *vec3);

	void   setGlobalUniformManager(UniformManager* uniformManager);

	UniformManager*     createUniformManager();

	UniformManager*     getGlobalUniformManager();

private:

	int           m_ProgramId;

	string        m_ProgramName;

	UniformManager*     m_GlobalUniformManager;


	map<string, unsigned int> m_UniformLocations;

	map<unsigned int, string> m_Attributes;

	friend ShaderManager;
};
```

Şekil 4.12. ShaderProgram sınıfının prototipi

4.2.2.2. GLSL (OpenGL gölgelendirici dili)

OpenGL çizim iş hattının belirli aşamaları kullanıcılar tarafından programlanabilmektedir. Programlama dili olarak da GLSL (OpenGL Shader Language) kullanılmaktadır. GLSL dili gramer olarak C diline benzemektedir. Şekil 4.13.'de Nokta gölgelendiricisi olarak kullanılacak temel bir GLSL kaynak kodu gösterilmektedir. GLSL dili geliştirildiği günden bu yana birçok değişimden geçmiştir. Grafik motoru tasarlanırken dilin 4.3 sürümü kullanılmıştır.

```
layout(location = 0) in vec3 position;

void main()
{
      gl_Position = vec4(position, 1.0);
}
```

Şekil 4.13. Temel bir GLSL kaynak kodu

4.2.2.3. Nokta gölgelendiricisi

Çizilecek şekle ait olan noktalar ilk olarak nokta gölgelendiricisine verilmektedir. Bu gölgelendiricinin en temel görevi noktaları yerel konumdan alıp 3B ortamdaki konumlarına taşımaktır. Bu işlem için programcı tarafından yazılıp OpenGL çizim iş hattına yüklenmiş bir gölgelendirici program kullanılmaktadır. Şekil 4.14.'de basit bir nokta gölgelendirici kodu gösterilmektedir.

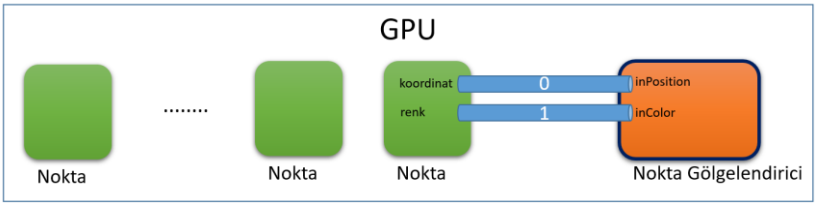
Kaynak kod içerisindeki ilk iki satır kullanıcının gönderdiği verileri temsil etmektedir. "inPosition" değişkeni gönderilen noktanın koordinat verisini tutarken "inColor" değişkeni renk değerini tutmaktadır. Çizdirilecek şekle ait her bir nokta için nokta gölgelendirici kodu bir kez çalıştırılmaktadır. Programcı tarafından gönderilen noktaların içerisindeki verilerin gölgelendiriciye bildirilmektedir. Bu işlem temsili olarak Şekil 4.15.'de gösterilmektedir. Şekle göre nokta yapısı iki veri barındırmaktadır. Bunlar koordinat ve renk verileridir. Gölgelendirici içerisinde de bu

verileri temsil edecek aynı boyuta sahip iki değişken oluşturulmuştur. Değişkenlerin konumları da nokta yapısındaki karşılıkları ile aynı olmak zorundadır.

```
layout (location = 0) in vec3 inPosition;
layout (location = 1) in vec4 inColor;

uniform mat4 uMtxTransform;

out vec4 color;

void main()
{
    gl_Position = uMtxTransform*vec4(inPosition, 1.0);
    color = inColor;
}
```

Şekil 4.14. Basit bir nokta gölgelendirici kodu

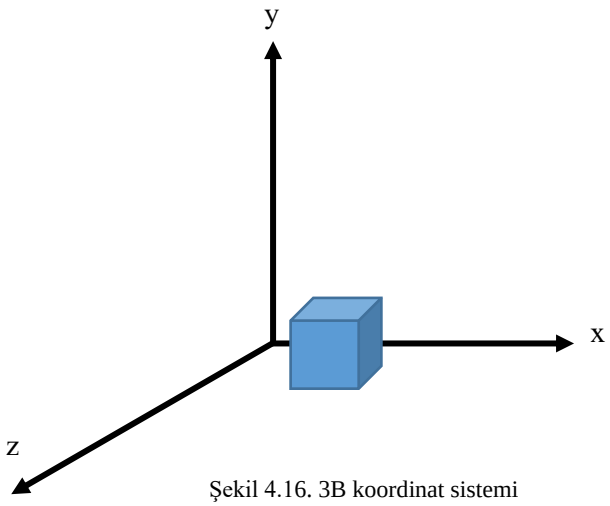
Şekil 4.14.'de nokta gölgelendirici kaynak kodunda öncelikle noktaya ait veriler iki değişkene bağlanmaktadır. Ardından şekle ait noktaları 3B dünya içerisindeki konumlarına yerleştirmek için "uMtxTransform" isimli dönüşüm matrisi ile çarpmaktadır. Elde edilen sonuç bir sonraki gölgelendiriciye gönderilecek "gl_Position" değişkenine atanmaktadır.



Şekil 4.15. Nokta gölgelendiriciye gönderilen verilerin çekilmesi

4.2.2.3.1. 3B koordinat sistemi

3B koodrinat sistemi aynı noktadan çıkan birbirine dik üç eksenden oluşmaktadır. Şekil 4.16.'da 3B koordinat sistemi eksenleri ve örnek bir nesne gösterilmektedir. Nesneler gerçek hayatta, genişlik, yükseklik ve derinlik olmak üzere üç boyuta sahiptir. Bu boyutlar eksenler ile de temsil edilmektedir. Boy y-ekseni, genişlik x-ekseni ve derinlik z-ekseni olarak kabul edilmektedir.



Şekil 4.16. 3B koordinat sistemi

4.2.2.3.2. 3B dönüşüm işlemleri

Grafik motoru üç farklı dönüşüm tekniği kullanmaktadır. Bunlar döndürme, ölçekleme ve ötelemedir. Her bir dönüşüm işlemi için 4x4 kare matris oluşturulmaktadır. Şekil 4.16.'da eksenlere göre döndürme dönüşüm matrisleri gösterilmektedir. Matrisler döndürme açısının sinüs ve kosinüs değerleri kullanılarak oluşturulmaktadır.

X Eksenine Göre

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha & 0 \\ 0 & sin\alpha & cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Y Eksenine Göre

$$\begin{bmatrix} cos\alpha & 0 & sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -sin\alpha & 0 & cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Z Eksenine Göre

$$\begin{bmatrix} cos\alpha & -sin\alpha & 0 & 0 \\ sin\alpha & cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.17. 3B uzayda eksenlere göre döndürme matrisleri

Ölçekleme ve öteleme işlemleri, döndürmenin aksine tek matris ile temsil edilebilmektedir. Şekil 4.17.'de bu matrisler gösterilmektedir. Ölçekleme matrisinde ölçekleme miktarı eksenlere göre belirtilmelidir. Bu sayede şekillerin her bir boyutu farklı bir ölçeklemeye tabi tutulabilmektedir. Öteleme işlemi şeklin 3B uzayda bir konumdan başka bir konuma taşınması olarak tanımlanabilir. Öteleme matrisi tek bir form ile temsil edilebilmektedir. Matris yapılacak ötelemenin eksenlerdeki büyüklüğüne göre hesaplanmaktadır. Bu sayede şekil istenilen eksende hareket ettirilebilmektedir.

Ölçekleme

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

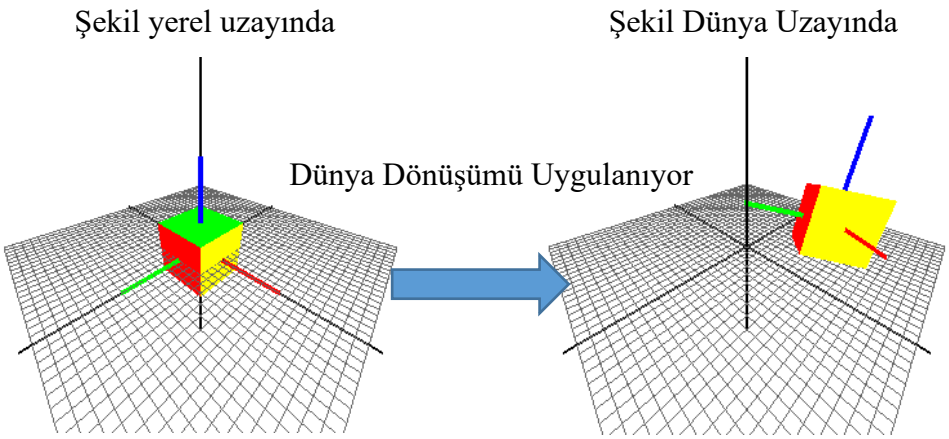
Öteleme

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.18. 3B uzayda ölçekleme ve öteleme matrisleri

4.2.2.3.3. Dünya dönüşüm matrisi

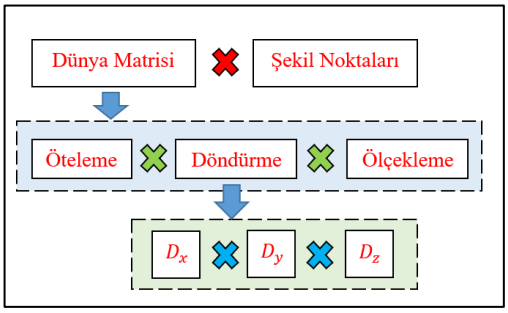
3B nesneler tasarım aşamasında koordinat sisteminin merkezinde tasarlanmaktadır. Nesnenin içinde bulunduğu bu uzaya yerel uzay adı verilmektedir. Şekil 4.18.'de yerel uzayda tasarlanmış bir küp şekli gösterilmektedir. Dönüşüm işlemi uygulanmamış bütün şekiller koordinat sisteminin merkezinde bulunmaktadır. Programcı şekilleri yükledikten sonra 3B ortam içerisinde bir konuma yerleştirmek istediğinde şekle ait noktalara çeşitli dönüşüm işlemleri uygulamak zorundadır. Bu dönüşüm işlemlerinin birleştirilmesine dünya dönüşüm matrisi adı verilmektedir. Dünya dönüşüm matrisi şeklin 3B dünya içirisindeki konumunu ve oryantasyonunu temsil etmektedir.



Şekil 4.19. Bir şeklin yerel uzayından dünya uzayına geçişi

Dünya dönüşüm matrisi elde etmek için şekle ait olan döndürme, ölçekleme ve öteleme matrisleri çarpma işlemi kullanılarak birleştirilir. Bu işlem Şekil 4.19.'da gösterilmektedir. Matrislerin çarpım sırası önemlidir. 3B bir şeklin dönüşüme

uğrayabilmesi için şekle ait bütün noktaların dünya matrisi ile çarpılması gerekmektedir.

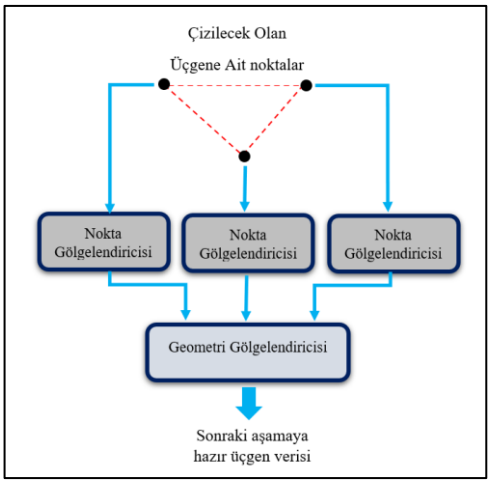


Şekil 4.20. Dünya dönüşüm matrisinin elde edilme adımları

Şekil 4.14.' deki nokta gölgelendiricisinde "uMtxTransform" matrisi çizilecek şekle ait olan dünya matrisini temsil etmektedir. Bu matrisin değeri programcı tarafından Şekil 4.19.'daki sırayla hesaplanmakta ve çizim başlamadan önce gölgelendiriciye gönderilmektedir.

4.2.2.4. Geometri gölgelendiricisi

Geometri gölgelendiricisi çizilecek şekli oluşturan en küçük geometrik birimler için çağrılmaktadır. Tasarlanan grafik motoru çizim birimi olarak üçgen kullandığı için motor içerisindeki geometri gölgelendiricileri her bir üçgen için bir defa çağrılmaktadır. Buna göre bir üçgen için üç nokta olduğundan üç defa nokta gölgelendiricisi çağrılırken, bu birimlerden çıkan pozisyon değerleri bir geometri gölgelendiricisinde işlem görecektir. Şekil 4.20.'de bir üçgen şekli için devreye giren nokta ve geometri gölgendiricileri göstermektedir. Burada çizilecek olan üçgene ait noktalar aynı programa sahip nokta gölgelendiricilerine gönderilmektedir. Çıkan sonuçlar da tek bir geometri gölgelendiricisine aktarılmaktadır. Geometri gölgelendiricisi bu verileri kullanarak tek bir üçgen elde edebileceği gibi birden fazla üçgen de üretebilmektedir. Grafik motorunda geometri gölgelendiricisi modellerin tel kafeslerini elde etmek, voksel haritaları oluşturmak gibi işlemlerde kullanılmaktadır.

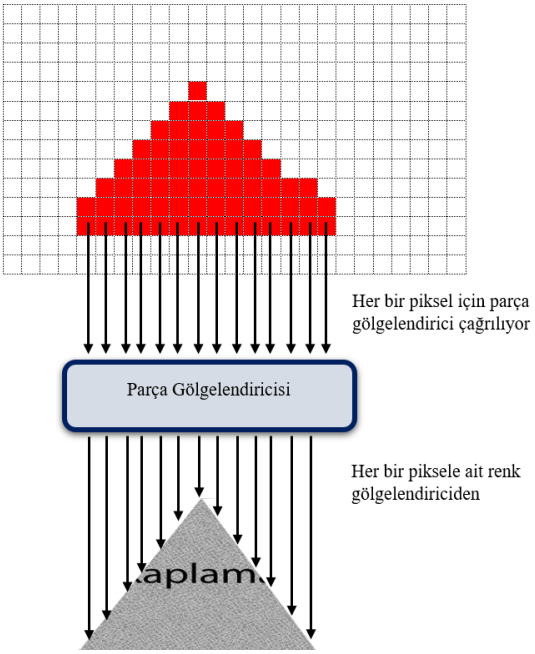


Şekil 4.21. Geometri gölgelendiricisinin nokta gölgelendiricisi ile iletişimi

4.2.2.5. Parça gölgelendiricisi

Parça gölgelendiricisi çizilecek şeklin monitörde kaplayacağı piksellerin renklerini belirlemek için kullanılmaktadır. Her bir piksel için bir defa çağrılmaktadır. Buna göre bir üçgen için üç nokta gölgelendirici, bir geometri gölgelendirici ve oluşacak pikselleri için çok daha fazla sayı parça gölgelendiricisi çağrılacaktır. Parça gölgelendiricisinde kaplama giydirme işleminin yanı sıra, ışık hesaplamaları, kenar belirleme gibi işlemler gerçekleştirilmektedir.

Şekil 4.21.'de parça gölgelendiricisinin çizdirilecek bir üçgene ait pikseller üzerinde çağrılması gösterilmektedir. Parça gölgelendiricisi piksellerin konumlarını bilmemektedir. Piksellerin adedi ve konumları önceki aşamalarda hesaplanmaktadır. Çizim iş hattı piksellerin konum ve sayıları hesaplandıktan sonra her bir piksel için bir defa parça gölgelendiricisini çağırmaktadır. Parça gölgelendiricisi sonucunda üçgenin piksel renkleri çıkacaktır.



Şekil 4.22. Parça gölgelendiricinin çizilecek üçgene ait pikseller için çağrılması

Temel bir parça gölgelendirici Şekil 4.22.'de verilmiştir. Burada "in" ile işaretlemiş "texCoord" değişkeni değerini çizim iş hattının bir önceki aşamasından almaktadır. Bu değişken parça gölgelendiricinin çağrıldığı pikselin alacağı rengin kaplama dosyasındaki koordinatını tutmaktadır. "fragColor" değişkeni parça değişkenin çıkış olarak belirlediği değişkendir. Piksel rengi bu değişken içerisine atanacaktır. "main" fonksiyonu içerisinde texture fonksiyonu "texCoord" değişkeni ile kaplamadaki rengi getirmektedir. Bu renk "fragColor" değişkenine atanmaktadır. Bu işlem bütün piksellere uygulandığında her piksel kaplamadaki kendine düşen rengi alacaktır.

```
out vec4 fragColor;

in vec2 texCoord;

uniform sampler2D ourTexture;

void main()
{
    fragColor = texture(ourTexture,texCoord);
}
```

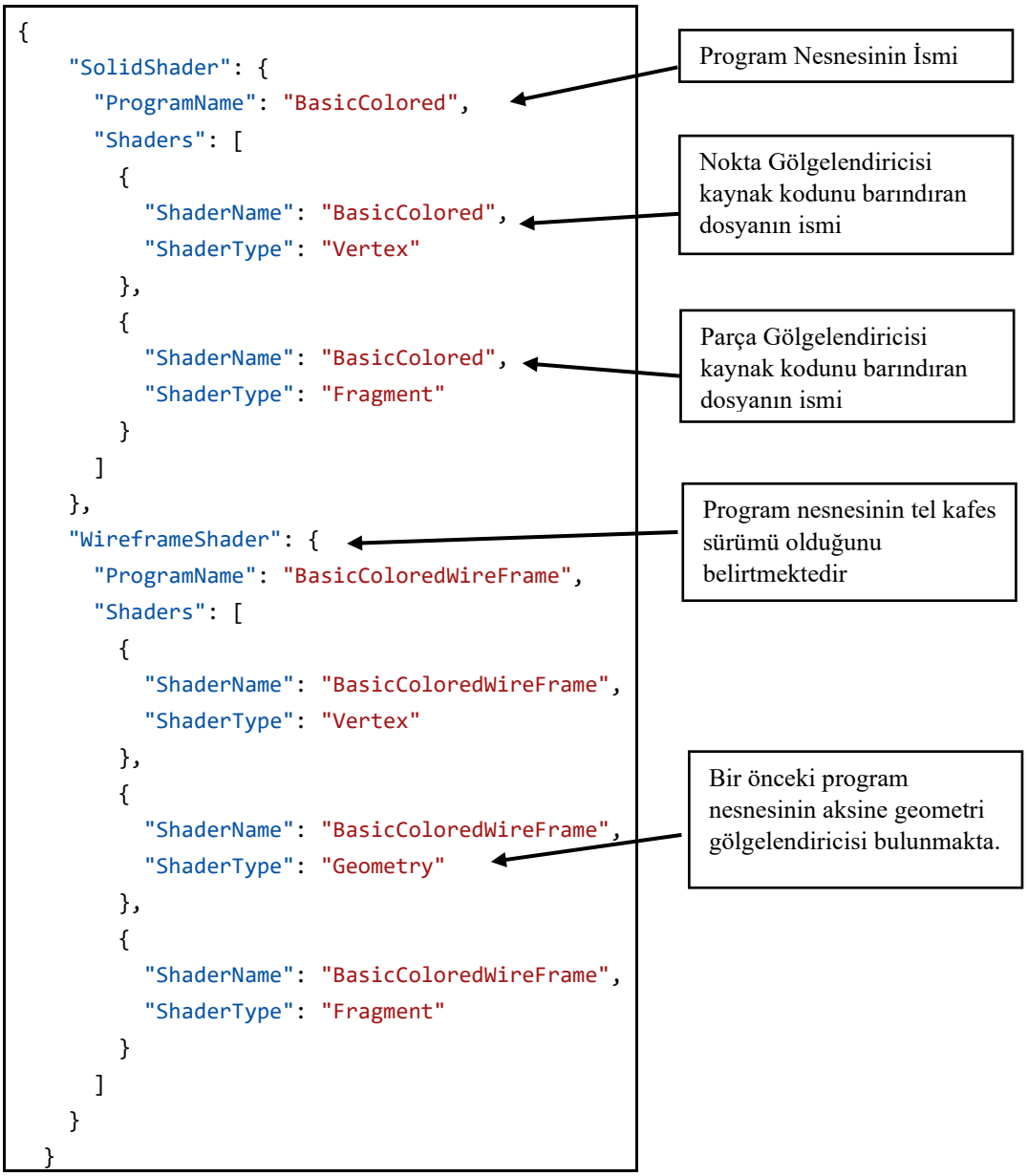
Şekil 4.23. Kaplama giydirme işlemi yapan temel bir parça gölgelendirici

4.2.2.6. Gölgelendirme yönetici sınıfı

Günümüzdeki çoğu 3B grafik uygulamalarının çalışması sırasında farklı gölgelendirici programlar kullanılmaktadır [37]. Farklı modeller farklı efeklere sahip olabilmektedir. Uygulamaların karmaşıklığı arttıkça kullanılacak gölgelendirici programlarda farklılık gösterecektir. Bu çalışmada tasarlanan grafik motoru kullanıldığı uygulamalar için çok sayıda gölgelendirici kombinasyonlarını çalıştırabilmektedir. Gölgelendirme sınıfı farklı kombinasyonlara sahip olacak çizim iş hatlarını yönetmek için tasarlanmıştır. Yönetici görevi gördüğü için tekil tasarım deseni kullanılarak tasarlanmıştır. Çizim iş hattında kullanılacak gölgelendiricileri sabit diskten okumak ve uygulamanın içerisinde aktif etmeye kadar bütün işleri organize etmektedir.

Çizim iş hatları program nesneleri tarafından temsil edilmektedir. Gölgelendirme yönetici sınıfı oluşturduğu her bir çizim iş hattı için bir program nesnesi oluşturmakta ve nesneyi içerisinde barındırdığı harita veri yapısında tutmaktadır. Yönetici sınıfı program nesnesini oluşturacak gölgelendirici kaynak kod dosyalarını grafik motoru için geliştirilmiş olan bir ayar dosyasından okumaktadır. Şekil 4.23.'de çizim iş hattı için kullanılacak bir ayar dosyası gösterilmektedir.

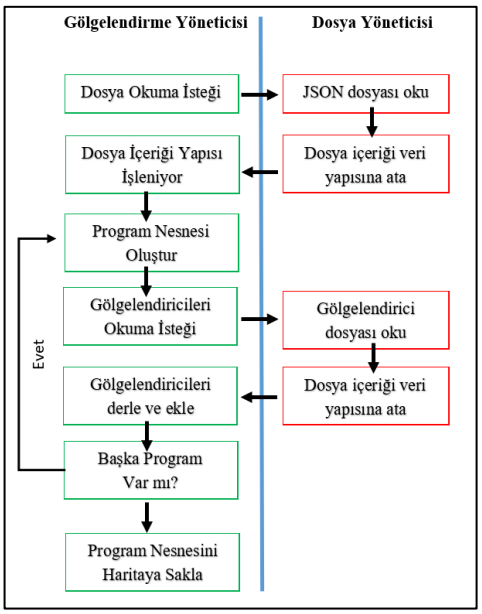
Çizim iş hattı ayar dosyası iki temel kısımdan oluşmaktadır. Bunlar "SolidShader" ve "WireFrameShader" şeklinde adlandırılmıştır. SolidShader çizilecek şeklin katı bir model olarak ekrana çıkartılmasını sağlayacak olan gölgelendiricileri belirtmektedir. WireFrameShader ise şeklin tel kafes formunda çizilebilmesi için gerekli olan gölgelendirici kombinasyonlarını belirtmektedir. Dosya JSON (Javascript Object Notation) formatıyla düzenlenmiştir. Dosya okuma ve gölgelendirme yöneticinin anlayacağı veri yapısına dönüştürme işlemi dosya yönetimi birimi tarafından yapılmaktadır.



Şekil 4.24. Çizim iş hattını oluşturan gölgelendiricileri belirten ayar dosyası örneği

Ayar dosyasındaki “ProgramName” parametresinin aldığı değer çizim iş hattı için oluşturulacak program nesnesinin ismini temsil etmektedir. Kaplama yöneticisinde olduğu gibi gölgelendirme yöneticisi de program nesnesini harita veri yapısına bu isimle indislemektedir. ProgramName parametresi ardından çizim iş hattında kullanılacak gölgelendiricilerin özellikleri verilmektedir. Her gölgelendirici “ShaderType” parametresine sahiptir. Buradaki değer gölgelendiricinin türünü belirtmektedir. “Shadername” parametresi gölgelendiricinin kaynak kodunun saklandığı dosyanın ismini belirtmektedir. Yönetici bu ismi ve gölgendirici türünü

kullanarak okunacak dosyaya karar vermektedir. Okuma işlemi dosya yöneticisine yapılan bir istekle gerçekleştirilmektedir. Dosya yöneticisi JSON formatında yazılmış ayar dosyasını okuyup istenilen gölgelendirici isimlerini sırayla gölgelendirme yöneticisine getirmekte ve bu isimler kullanılarak öncelikle bir program nesnesi oluşturulmaktadır. Ardından bu nesneye ait olan gölgelendiriciler okunup derlenerek program nesnesi içerisine eklenmektedir. Bu işlemler Şekil 2.24.'deki diyagramda gösterilmektedir. Yükleme işlemi yapılırken grafik motorunun iki modülü arasındaki iletişim de diyagramda görülmektedir.



Şekil 4.25. Çizim iş hattı ayar dosyasının sisteme yüklenme aşamaları ve ilgili modüller

4.2.2.6.1. Genel değişkenler

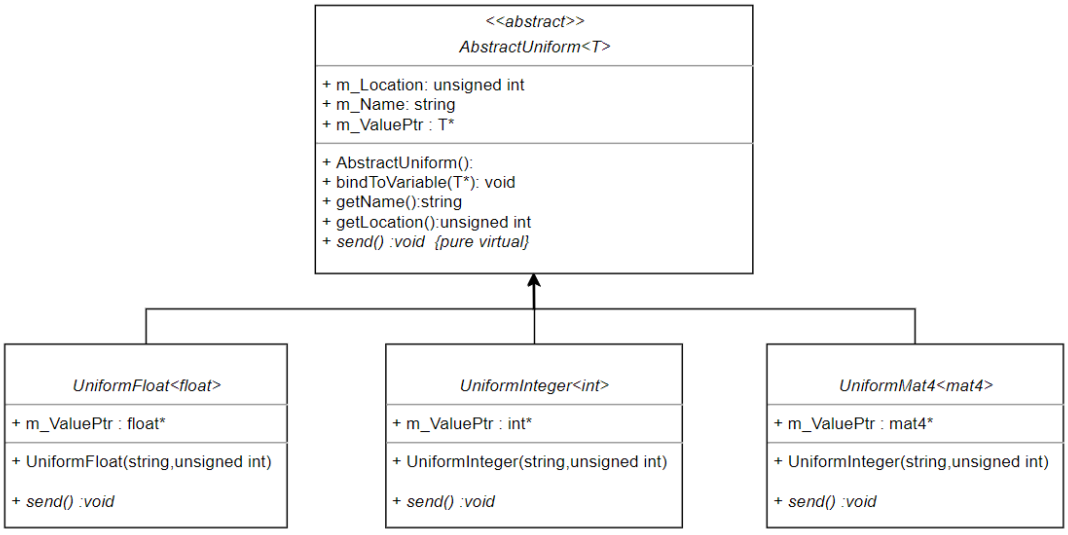
Gölgelendirici programlar içerisinde genel değişken adı verilen değerleri programcı tarafından gönderilen global değişkenler bulunmaktadır. Genel değişkenler "uniform" anahtar kelimesi ile tanıtılmaktadır. Genel değişkenler çizim iş hattında global olarak hareket etmektedirler. İş hattının bir adımında tanımlı olan değişken diğer iş adımlarından da erişelebilir. Genel değişkenlere gölgelendirici program içerisinde değer atanamaz ve değerleri çizim iş hattı başlamadan önce programcı tarafından gönderilmelidir.

Program nesneleri oluştururken, nesneye ait olan gölgelendiricilerin genel değişkenleri tesbit edilmekte ve OpenGL aracılığıyla bir iletişim yolu kurulmaktadır.

4.2.2.6.2. Genel değişken yöneticisi

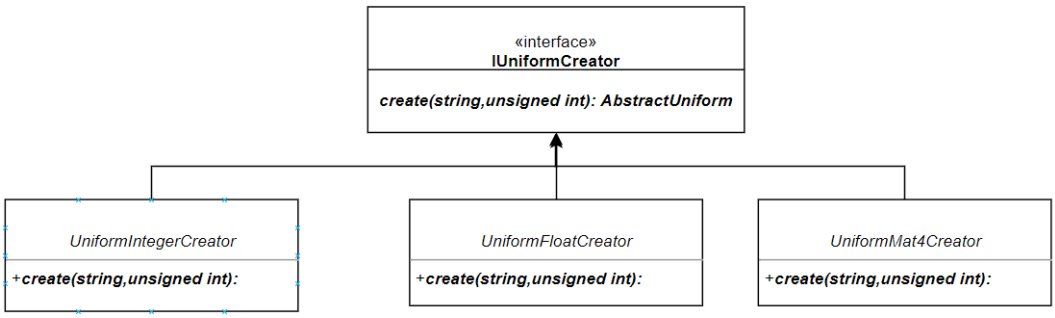
Gelen değişkenler veri türüne göre farklılık göstermektedir. Veri türündeki farklılık iletişim fonksiyonunun da farklı olmasına sebep olmaktadır. Bu yüzden grafik motoru içerisinde genel değişkenleri temsil etmek için tek bir sınıf kullanmak yerine veri türüne göre farklı sınıflar geliştirilmiştir. Bu sınıfların veri türleri farklı olmasına rağmen eylemleri birbirine benzediği için sınıfların kalıtım alacağı bir arayüz sınıfı geliştirilmiştir.

AbstractUniform sınıfı genel değişkenlerin ebeveyn sınıfıdır. Bu sınıf kalıtım alacak sınıfların temel veri tipleri bilinmediği için şablon olarak tasarlanmıştır. T harfi veri türünü temsil etmektedir. Kalıtım alınırken T veri türü belirtilmektedir. Şekil 4.25.'de genel değişken sınıflarının UML kalıtım diyagramı verilmiştir. Şekilde sadece üç sınıf verilmiştir fakat fabrika tasarım deseni sayesinde kullanıcılar istedikleri yeni sınıfları da sisteme dahil edebilmektedir.



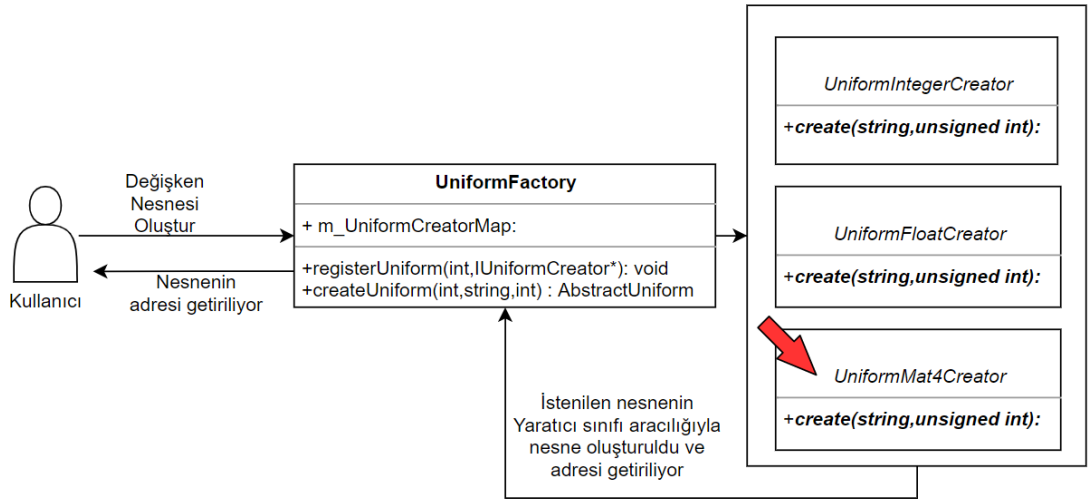
Şekil 4.26. Genel değişken sınıfları UML diyagramı

Genel değişken sınıflarını oluşturmak için fabrika tasarım deseni kullanılmıştır. Bu sayede gölgelendiriciler içerisinde bulunan genel değişkenlerin türü ne olursa olsun ilgili nesneler fabrika sınıfı tarafından üretilebilecektir. Fabrika tasarım deseninin çalıştırılabilmesi için üretilecek olan sınıflara ait yaratıcı fabrika sınıflarının da oluşturulması gerekmektedir. Yaratıcı fabrika sınıflarının UML diyagramları Şekil 4.26.'da gösterilmektedir. Yaratıcı fabrika sınıfları IUniformCreator isimli bir arayüz sınıfından kalıtım almaktadır. Bu arayüzün tek bir soyut fonksiyonu bulunmaktadır. Bu fonksiyon arayüzden kalıtım alan her sınıfta tanımlanmak zorundadır.



Şekil 4.27. Yaratıcı fabrika sınıfları UML diyagramı

Şekil 4.27.'de kullanıcıların bir genel değişken nesnesi oluşturma isteği sırasında gerçekleşen işlemlerin diyagramı gösterilmiştir. UniformFactory sınıfı içerisinde bütün genel değişken sınıfları için bir yaratıcı nesne barındırılmaktadır. Kullanıcı createUniform metodu aracılığıyla fabrika nesnesine ihtiyacını bildirmektedir. Fabrika nesnesi bu isteğe karşılık gelen yaratıcı nesneye yönlendirilmektedir. Şekilde yönlendirilen yaratıcı nesne kırmızı ok ile gösterilmiştir. Yaratıcı nesne görevli olduğu genel değişken nesnesi oluşturup adresini fabrika nesnesine döndürmektedir. Fabrika nesnesi de bu adresi kullanıcıya getirmektedir. Fabrika nesnesi yaratıcı nesneler için harita türünde bir veri yapısı tutmaktadır. Bu veri yapısı grafik motoru çalıştırıldığında UniformFactoryRegister isimli bir sınıf tarafından gerçekleştirilmektedir.



Şekil 4.28. Genel değişken nesnelerinin oluşturulması diyagramı

4.2.3. Model yöneticisi

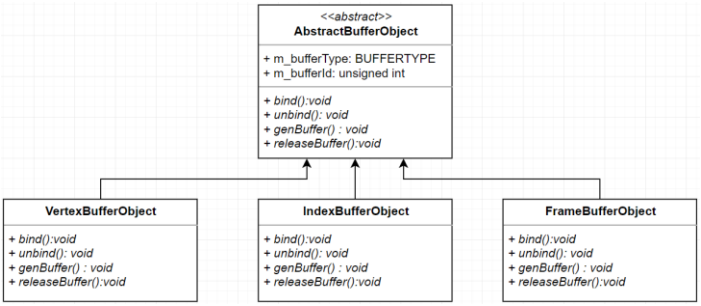
3B dünyada modeller üçgenlerden oluşmaktadır. Modellerin üçgen sayısı arttıkça inşa edilmeleri, hafızada kapladıkları alan ve çizim işlemleri de karmaşıklaşmaktadır. Grafik motoru içerisinde bu işlemleri gerçekleştirmek için model yöneticisi adında bir yazılım birimi oluşturulmuştur. Model yöneticisi modeli oluşturan üçgenlere ait bilgilerin dosyalardan okunması, gerekli tampon belleklerinin bu bilgiler ile doldurulması, modele bağlı olan kaplama ve program nesnelerinin yüklenmesi ve oluşturulan modelin sunum motoruna bağlanması gibi işlemlerden sorumludur.

4.2.3.1. 3B modellerin yönetilmesi ve yardımcı araçlar

Grafik motorunun sanal ortamı oluşturmak için kullanacağı 3B modellerin sisteme yüklenmesi, kontrolü ve çizilebilmesi için çeşitli araçlara ihtiyaç duyulmaktadır. OpenGL kütüphanesi bu işlemler için yeterli olsa da direk OpenGL kütüphanesini kullanmak programcıların işlemlerini daha karmaşık ve zor hale getirmektedir. Bu yüzden 3B modellerin sanal ortam içerisinde daha rahat kullanılabilmesi için bazı yazılım araçları geliştirilmiştir.

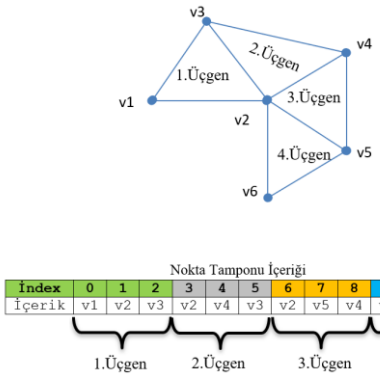
4.2.3.1.1. Nokta tamponu ve sınıfı

OpenGL bir modeli çizebilmek için modele ait olan üçgenleri oluşturan noktalara ihtiyaç duymaktadır. Programcı bu noktaları bir OpenGL nesnesi olan nokta tamponu içerisine yerleştirmelidir. OpenGL nokta tamponu ardışıl bir hafıza bölgesinden ibarettir. OpenGL nokta tamponu oluşturma, veri yükleme ve gerekli ayarların yapılması işlemleri daha kolay kullanılabilecek hale getirmek için model yönetim birimi içerisinde VertexBufferObject isimli bir sınıf oluşturulmuştur. Model yönetim sistemi içerisinde benzer özelliklere sahip sınıflar bulunduğundan VertexBufferObject sınıfı AbstractBufferObject adında soyut bir sınıftan kalıtım almaktadır. Şekil 4.28.'de tampon belleklerini kontrol eden sınıfların UML diyagramı gösterilmektedir.



Şekil 4.29. Tampon belleklerini kontrol eden sınıfların UML diyagramı

Modelleri oluşturan üçgenlere ait noktaların bilgileri nokta tampon belleğinde ardışıl olarak yerleştirilmektedir. Şekil 4.29.'da üçgenlerden oluşan bir modelin noktalarının hafızaya nasıl yerleştirildiği gösterilmiştir.



Şekil 4.30. Üçgenlere ait noktaların nokta tampon belleğine yerleştirilmesi

Noktalara ait veriler şekle uygulanacak işlemlere ve efektlere göre değişiklik göstermektedir. Şekil 4.30.'da üç farklı nokta veri yapısı gösterilmektedir. Burada Vertex3Color yapısında noktalar sadece koordinat ve renk bilgisine sahip olacaktır. Vertex3Tex yapısında ise noktalar koordinat ve kaplama koordinatı bilgilerine sahiptir. Son olarak Vertex3LightTex yapısı üç boyutlu ortamda ışık hesabının yapıldığı durumlarda devreye girmektedir. Bu yapı içerisinde noktanın koordinatı, normal vektörü ve kaplama koordinatları bulunmaktadır.

```
class Vertex3Color
{
public:
    glm::vec3 pos;

    glm::vec4 color;
};
```

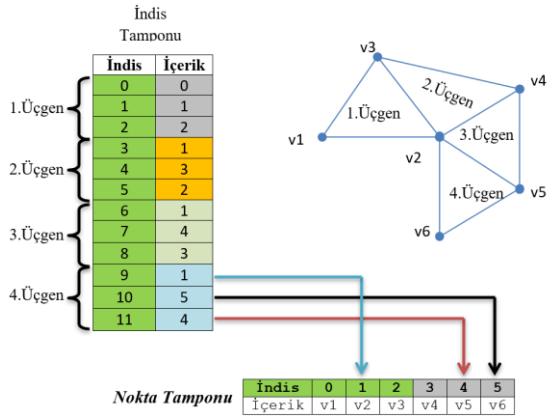
```
class Vertex3Tex
{
public:
    glm::vec3 pos;

    glm::vec2 tex;
};
```

```
class Vertex3LightTex
{
public:
    glm::vec3 pos;
    glm::vec3 normal;
    glm::vec2 tex;
};
```

Şekil 4.31. Farklı efektler ile kullanılacak nokta sınıfları

4.2.3.1.2. İndis tamponu ve sınıfı

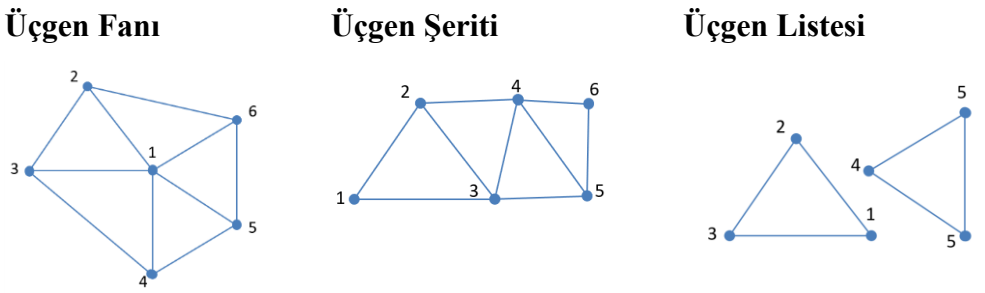
İndis tamponu, üçgeni oluşturan noktaların nokta tamponundaki sıralarını (indislerini) tutmaktadır. Bu sayede nokta tamponunda sadece özellikleri farklı noktaların tutulmasını sağlayacaktır. Çizim yapılırken indis tamponundaki nokta sıraları kullanılarak şekle ait üçgenler çizilmektedir. Şekil 4.31.'de indis tamponu ile ona karşılık gelen nokta tamponunun hafıza görüntüsü verilmiştir. İndis tamponu her bir üçgeni oluşturan noktaların nokta tamponundaki konumlarını yani indislerini tutmaktadır. Örneğin 4.üçgeni oluşturan noktaların indisleri sırasıyla 1, 5 ve 4 değerine sahiptir. Bu indis değerleri indis tamponunda 9, 10 ve 11. hücrelere yerleştirilmektedir. İndis tamponları bir nokta tamponuna bağlı olarak çalışmaktadır.



Şekil 4.32. Örnek indis tampon içeriği ve buna karşılık gelen nokta tampon içeriği

Model yöneticisi içerisinde indis tamponunu temsil edilmesi için IndexBufferObject isimli bir sınıf oluşturulmuştur. Bu sınıf içerisinde indis tamponunun kullandığı nokta tamponunun da kimlik bilgileri tutulmaktadır. Bu sayede sadece IndexBufferObject nesnesi aracılığıyla üçgenler çizilebilmektedir.

İndis tamponunda kullanılan indis değerleri üçgenlerin çizilme yöntemine göre değişmektedir. Şekil 4.32.'de üçgenlerin temel çizilme şekilleri gösterilmektedir. Üçgen listesinde her bir üçgen için üç indeks bilgisi kullanılmaktadır. Bu çizilme yöntemi ile her türlü 3B model temsil edilebilmektedir fakat hafızada kapladığı alan diğerlerine göre daha fazladır. Üçgen şeriti ve üçgen fanı zemin tasarımı gibi sadece belirli modeller üzerine uygulanabilmektedir. Üçgen şeritinde her bir üçgene ait indislerin son ikisi sıradaki üçgenin ilk iki indisi olmaktadır. Grafik motoru içerisinde kullanılan 3B modeller üçgen şeriti ve fanı ile temsil edilemeyecekleri için indis tampon sınıfında üçgen listesi kullanılmıştır.



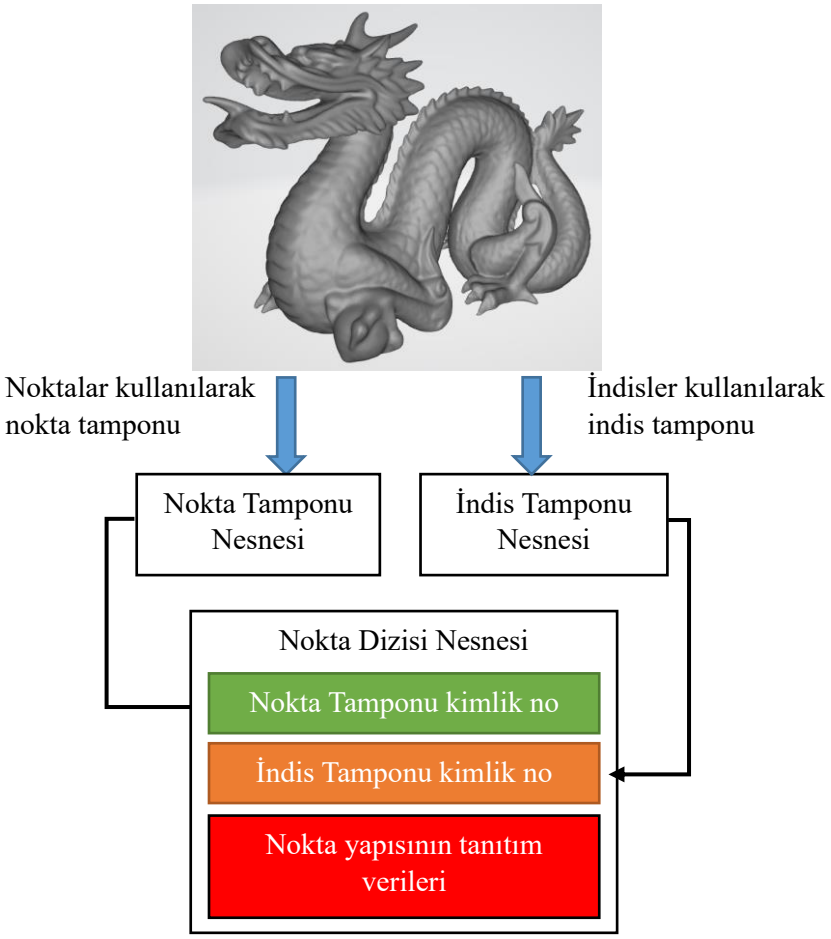
Şekil 4.33. Üçgenlerin temel çizilme şekilleri gösterilmektedir

4.2.3.1.3. Nokta dizisi nesnesi

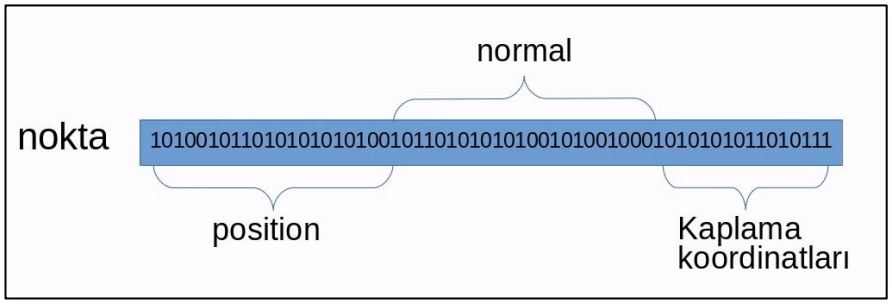
OpenGL 3B modellerin çizilmesinde kullanılan nokta ve indis tamponlarını gölgelendiriciler ile bağlamak için nokta dizisi nesnelerini kullanmaktadır. Bu nesneler çizim için kullanılacak olan nokta ve indis tampon nesnelerinin kimlik değerlerini tutmanın yanı sıra çizim için kullanılacak olan nokta yapısının içeriğini gölgelendiricilere iletmekle de görevlidir. Çizim esnasında her 3B modelin bir nokta dizisi nesnesi bulunması gerekir. Nokta ve indis tamponları nokta dizisi nesnesine bağlandıktan sonra programcı tarafından kullanım dışı kalmaktadır.

Şekil 4.33.'de 3B bir modeli oluşturan nokta ve indis tamponlarının nokta dizisi nesnesi tarafından temsil edilmesi gösterilmektedir. Nokta dizisi nesnesi nokta ve indis tamponlarına ait bir kimlik numarasını içerisinde barındırmaktadır. Çizim sırasında bu numaraları kullanarak 3B modeli çizdirebilmektedir.

Nokta verileri, gölgelendiricilere bir bit akışı olarak gönderilmektedir. Gölgelendiricilerin bu bitlerin içerisinde ayıklama yapabilmesi için nokta veri yapısının özelliklerini bilmesi gerekmektedir. Bu amaçla nokta dizisi içerisinde kullanılacak nokta yapısına ait bilgiler bulunmaktadır. Bu bilgilere nokta özelliği adı verilmektedir. Nokta veri yapısı içerisinde bulunan her bir veri bir özellik tarafından temsil edilmektedir. Nokta özellikleri aracılığıyla gölgelendiriciler kendilerine gelen nokta verisini okuyabilmektedir. Şekil 4.34.'de bir nokta verisinin temsili bit dizilişi ve bu bitlerin noktanın özelliklerine bölünmesi gösterilmiştir. Bu nokta yapısında ilk olarak koordinat bilgisi ardından ışık hesaplamasında kullanılacak olan normal bilgisi ve son olarak da kaplama koordinat bilgisi yer almaktadır.



Şekil 4.34. 3B modelin nokta dizisi nesnesi ile temsil edilmesi



Şekil 4.35. Nokta özelliklerinin temsili gösterimi

Nokta dizisi nesneleri nokta içerisindeki verilerin özelliklerini glVertexAttribPointer isimli fonksiyonu kullanarak bildirmektedir. Şekil 4.35.'de bu fonksiyonun Vertex3Color isimli bir nokta yapısına ait bir özellik için çağrılması gösterilmiştir. Nokta yapısı içerisindeki her bir özellik için glVertexAttribPointer fonksiyonunun çağrılması gerekir. Fonksiyonun ilk parametresindeki değer nokta gölgelendiricisinin

bu özelliği alacağı indisi temsil etmektedir. Gölgelendirici bu değeri kullanarak gelen noktanın hangi özelliği ile bağlantı kuracağını belirtmektedir.

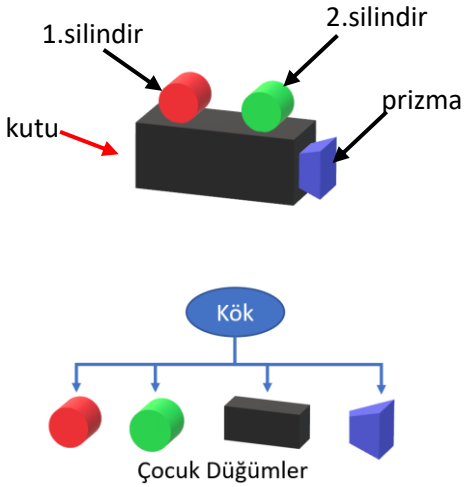
```
glVertexAttribPointer(0,    //özelliğin nokta dizisindeki indisi
                      3,    //özellikte kaç tane veri bulunuyor
                      GL_FLOAT, //her bir verinin türü
                      GL_FALSE, //normalize edilmelimi
                      sizeof(Vertex3Color),  //noktanın toplam boyutu
                      (void*)0); //özelliğin nokta içerisindeki konumu
```

Şekil 4.36. Nokta yapısına ait bir özellik tanıtımı

Nokta dizileri nesnesini teslim etmek için VertexArrayObject isimli bir sınıf geliştirilmiştir. Bu sınıf nokta ve indis verilerini kullanarak gerekli OpenGL çağrılarını yapmakta ve kullanıcıları daha kolay kontrol edilecek bir arayüz sağlamaktadır. Nokta dizileri nesnelerini daha iyi kontrol edebilmek ve kaynak israfını önlemek için de VertexArrayObjectManager isimli bir sınıf tasarlanmıştır. Bu sınıf grafik motorunun oluşturduğu her modele karşılık bir VertexArrayObject nesnesi oluşturmakta ve model ismine bağlı olarak bu nesneleri bir harita veri yapısında saklamaktadır.

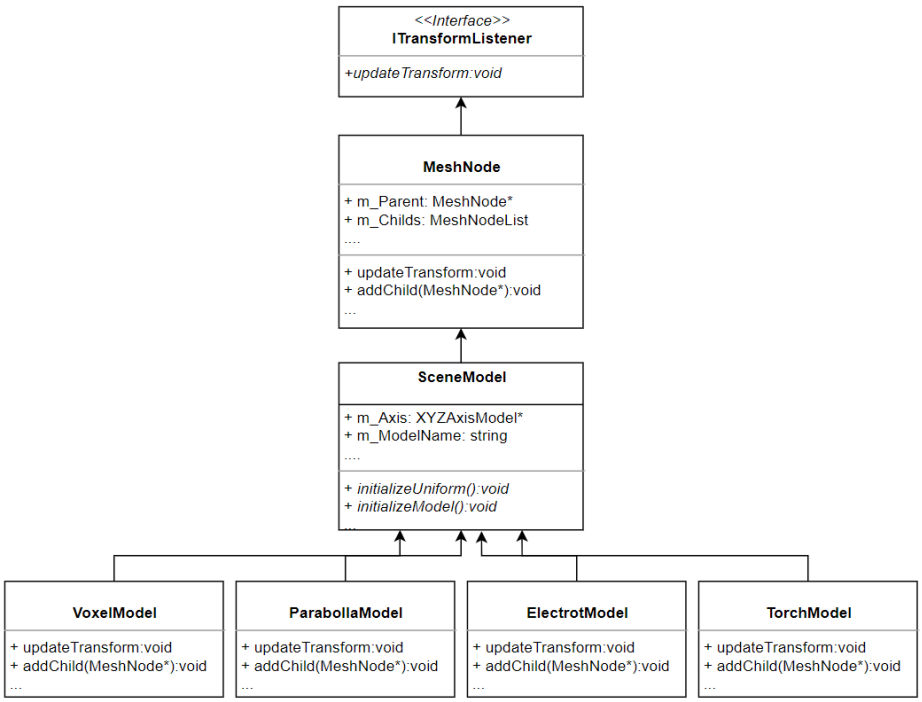
4.2.3.2. Model hiyerarşisi ve sınıfı

3B modeller tek bir parçadan oluşabileceği gibi birden fazla küçük modelin birleşiminden de oluşturulabilmektedir. Örneğin aşağıdaki model dört ayrı modelin birleşmesinden oluşmaktadır. Şekil 4.36.'da hiyerarşik yapıya sahip bir 3B model örneği gösterilmektedir. Model dört birimden oluşmaktadır. Bu tür hiyerarşik yapıya sahip modeller ağaç veri yapısı kullanılarak temsil edilmektedir. Her bir birim ağaç içerisinde bir düğüm ile temsil edilecektir. Bütün birimleri kontrol edecek bir kök düğüme de ihtiyaç duyulacaktır. Kök düğüm içerisinde bir model barındırmayacak, sadece alt düğümleri bir arada tutmak için kullanılacaktır.



Şekil 4.37. Hiyerarşik yapıya sahip bir 3B model örneği

Model hiyerarşisinde çocuk düğümlerin dünyadaki konum ve oryantasyonları ebeveynlerine bağlı olarak değişmektedir. Örneğin ebevyn +x ekseninde 2 birim hareket ederse çocuk düğümlerde aynı şekilde 2 birim hareket edecektir. Fakat çocuk düğümlerin hareketleri ebeveyni etkilememektedir. Çocuk ve ebeveyn düğümler arasındaki bu ilişkiler kurabilmek için UML diyagramı Şekil 4.37.'de gösterilen sınıflar tasarlanmıştır.

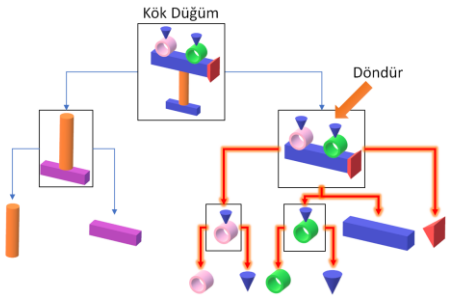


Şekil 4.38. 3B model hiyerarşisine ait sınıfların UML diyagramı

Her 3B modeline ait alt birimler MeshNode sınıfı tarafından temsil edilmektedir. Bu sınıf ayrıca kendi türünden çocuk nesneleri de barındırabilmektedir. Ayrıca ebeveyninin adresini de barındırmaktadır. Böylece modele ait ağaç yapısındaki düğümler arasında dolaşmak imkânı doğmaktadır. Sisteme yeni bir model ekleyebilmek için öncelikle yeni modeli temsil edecek bir sınıf tasarlamak gerekecektir. Tasarlanacak sınıfın grafik motoru içerisine yüklenebilmesi için SceneModel sınıfından kalıtım alması gerekmektedir. Grafik motoru SceneModel içerisinde initializeUniform ve initializeModel sanal fonksiyonlarını kullanarak modelin başlangıç ayarlarını gerçekleşmesini sağlamaktadır.

4.2.3.3. Model dönüşüm sistemi

3B modeller hiyerarşik bir yapıya sahip oldukları için modelin dönüşüm işlemlerinin de bu hiyerarşinin yapısına uygun bir şekilde gerçekleşmesi gerekmektedir. Dönüşüm bütün modele uygulanabileceği gibi sadece bir düğüme de uygulanabilir. Örneğin kök düğüme dönüşüm uygulandığında aynı dönüşümün çocuk düğüme de uygulanması gerekmektedir. Dönüşüm işlemlerinin hiyerarşinin başından en alttaki düğümlere kadar ulaşabilmesi için updateTransform sanal fonksiyonu kullanılmaktadır. Bu fonksiyon çağrıldığı düğümün bütün çocuklarına ait updateTransform fonksiyonlarını da çağırmaktadır. Böylece dönüşümden bütün çocukları etkilenmektedir. Şekil 4.39.'da 3B bir modelin bir alt düğümüne döndürme işlemi uygulanmaktadır. Dönüşümün uygulandığı düğüm ile aynı seviyede bulunan düğümler dönüşümden etkilenmemektedir. Dönüşüm işlemi uygulandığı düğüme ait bütün çocuklara seviye seviye uygulanmaktadır. Hiyerarşik yapısı sayesinde 3B modelleri oluşturan parçalar bağımsız şekilde hareket ettirilebilmektedir.



Şekil 4.39. 3B bir model hiyerarşisindeki bir düğüme dönüşüm uygulanması

Dönüşüm işleminin ebeveyn düğümden çocuklara aktarılması sırasında ebeyn düğümün dönüşüm matrisinin çocuk düğümlerin dönüşüm matrisleri ile de çarpılması gerekmektedir. Bu işlem sayesinde ebeveyn düğüme uygulanan dönüşüm işlemi çocuk düğümlere de aktarılmış olacaktır. updateTransform fonksiyonu içerisinde dönüşüm matrislerinin aktarılması Şekil 4.39.'da gösterilmektedir.

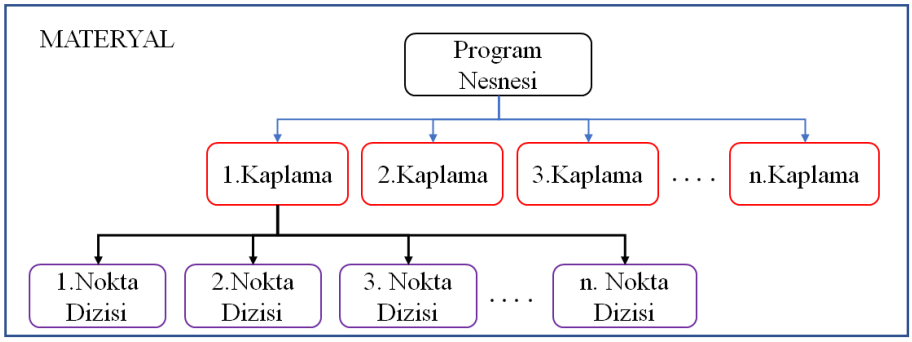
```
void MeshNode::updateTransform()
{
	std::stack<MeshNode*> nodeList;

	nodeList.push(this);

	getTransform()->updateWorldMatrix();

	while(nodeList.empty()!=true)
	{
		MeshNode* activeNode = nodeList.top();

		glm::mat4 transLocal, transWorld,result,last;

		nodeList.pop();

	    glm::mat4 matrixWorld = activeNode->getTransform()->getMatrix();

		for (auto child : activeNode->getChildList())
		{

		  child->getTransform()->setCombinedWorldMatrix(matrixWorld);
		  child->getTransform()->updateWorldMatrix();
		}

	}
}
```

Şekil 4.40. updateTransform fonksiyonun içeriği

4.2.4. Materyal yöneticisi

OpenGL kütüphanesi ile çizim yaparken en iyi performansı elde etmek için grafik donanımın olabildiğince az durum değişikliğine girmesi amaçlanmaktadır. Program nesnesinin değiştirilmesi, kaplama değiştirilmesi ve nokta dizisinin değiştirilmesi durumlarında grafik donanımı durum değişikliklerine gitmektedir. Her bir durum değişikliği belleğin boşaltılması gibi ek süre gerektiren işlemlerle sonuçlanmaktadır. Grafik donanımın durum değişikliğini en aza indirmek için çizilecek olan 3B modeller sahip oldukları özelliklere göre gruplara ayrılmaktadır.

Materyal çizim yapılacak bir grup 3B şekli temsil etmek için kullanılmaktadır. Her materyal aktif bir gölgelendirici program nesnesine sahiptir. Birden fazla kaplama barındırabilmektedir. Her kaplama da birden fazla nokta dizisi nesnesi tarafından kullanılabilmektedir. Şekil 4.40.'da bir materyal içerisinde barınan elemanların arasındaki ilişki gösterilmektedir. Grafik motoru çizim yaparken sanal ortamı kullanılan materyallere göre ayırmaktadır. Her materyali de içerisindeki kaplamalara göre ayırmaktadır. Aktif olan kaplamayı da kaplamayı kullanan nokta dizilerine göre ayırmaktadır. Böylece kaplama ve program nesneleri arasındaki geçiş en düşük seviyeye indirilecektir. Materyal yöneticisinin kullanımı sahne yönetim modülü içerisinde bulunan ve sahnenin çizilmesi ile görevli olan sunum motoru tarafından yapılmaktadır.



Şekil 4.41. Materyal nesnesi içeriği

4.3. Yardımcı Sistemler

Grafik motorunun ihtiyaç duyduğu sanal ortamın oluşturulması ile direk ilişkisi olmayan görevleri yardımcı sistemler modülü gerçekleştirmektedir. Modül matematik kütüphanesi, arayüz yöneticisi, dosya yöneticisi ve paralel iş yönetim sistemi olmak üzere dört birimden oluşmaktadır.

4.3.1. Matematik kütüphanesi

3B çizim işlemleri gerçekleştirirken çok sayıda matematiksel işlemler gerçekleştirilmektedir. Bu işlemler için gerekli olan fonksiyonları yeniden yazmak yerine hız ve güvenirliği test edilmiş hazır kütüphaneler kullanmak daha iyi sonuçlar

vermenin yanı sıra grafik motorunun geliştirme süresinin kısalmasını sağlamıştır. Bu sebeple grafik motoru içerisinde GLM (OpenGL Matematics) kütüphanesi kullanılmıştır. GLM kütüphanesi GLSL standartlarına göre geliştirilmiştir. Böylece gölgelendiriciler ile haberleşmede GLM yapıları sorunsuz bir şekilde kullanılabilmektedir. GLM matematiksel işlemleri intel ve amd işlemcilerinde paralel yaptıracak şekilde tasarlanmıştır.

4.3.2. Arayüz yöneticisi

Sanal kaynak simülatörünün kullanımını kolaylaştırmak için çeşitli arayüz araçlarına ihtiyaç duyulmaktadır. Bu amaçla AntTweakBar isim OpenGL kütüphanesi üzerine yazılmış bir arayüz sistemi grafik motoruna entegre edilmiştir. Arayüz elemanlarının oluşturulmasını ve yönetilmesini kolaylaştırmak için GuiModelController isimli bir sınıf tasarlanmıştır. Şekil 4.42.'de simülatörün çalışma anında devreye giren iki arayüz paneli gösterilmektedir. Kullanıcılar buradaki buton ve seçim araçlarını kullanarak simülatördeki modelleri manipüle edebilmektedir. Ayrıca yapılan kaynak işlemi ile alakalı bilgiler gerçek zamanlı olarak ekrana çıkartılabilmektedir.

Şekil 4.42. Örnek arayüz panelleri

4.3.3. Dosya yöneticisi

Simülatör yazılımı içerisinde sabit disk içerisinde bulunan verilerin okunması ve işlenmesi önemli bir görev teşkil etmektedir. Okunan veriler büyük bir çoğunluğu grafik motoru tarafından kullanılacağından dolayı dosya yükleme işlemleri için ayrı bir birim geliştirilmiştir. Dosya yönetici birimi, model yüklemeden, gölgelendirici kaynak kodlarının yüklenmesine kadar grafik motorunun çalışması için önemli görevleri üstlenmektedir.

4.3.3.1. Gölgelendirici kaynak kodlarının yüklenmesi

Gölgelendiricilere ait kaynak kodlar yazı dosyaları içerisinde saklanmaktadır. Grafik motoru ihtiyaç duyduğunda bu kodları dosya yöneticisi aracılığıyla okumaktadır. Dosya yöneticisinin görevi ismi verilen gölgelendiricinin dosyasını açmak ve içerisindeki veriyi tek bir karakter dizisi içerisine yerleştirip grafik motoruna teslim etmektir.

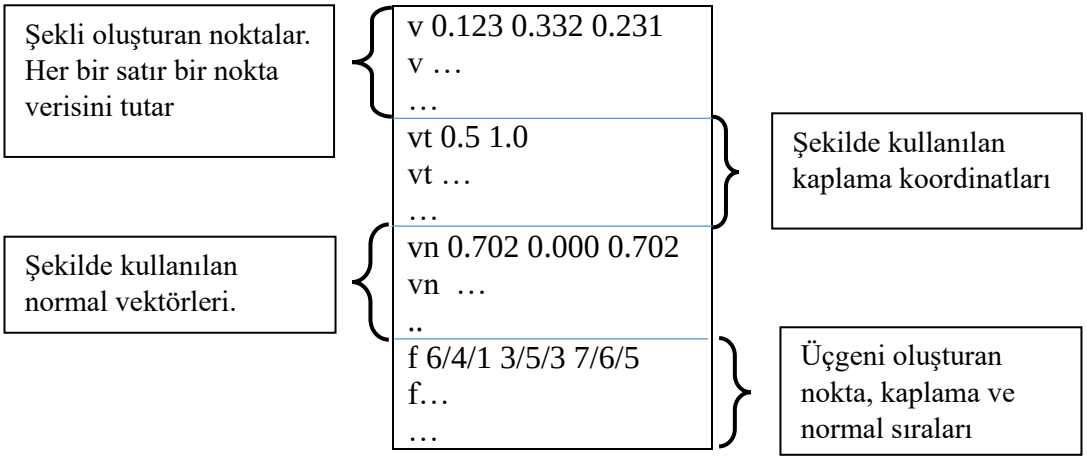
4.3.3.2. Kaplama dosyalarının okunması

Modellere ait kaplama dosyalarının yüklenmesi için SOIL isimli resim yükleme kütüphanesi kullanılmaktadır. Bu kütüphane okunan resim dosyasının piksel değerlerini ve piksel özelliklerini getirmektedir. SOIL kütüphanesi dosya yöneticisi içersinde bulunan TextureLoader sınıfı tarafından kontrol edilmektedir. Bu fonksiyon kendisinden istenilen dosyayı SOIL kütüphanesi kullanarak okumakta ve gelen verileri grafik motorunun kullanabileceği bir yapıya çevirerek kullanıcıya döndürmektedir.

4.3.3.3. Model dosyalarının okunması

Grafik motoru, stl ve obj olmak üzere üç farklı 3B model dosya formatı okumaktadır. Bu işlemler için assimp adlı bir kütüphane kullanılmaktadır. Assimp kütüphanesi çok sayıda 3B dosya formatı desteklemektedir. Böylece grafik motoru başka dosya formatlarına da kolayca adapte edilebilecektir.

Torç ve kaynak masası gibi statik 3B modeller Pro-Engineer adlı 3B tasarım programında dizayn edilmişlerdir. Pro-Engineer programının modelleri kaydetmek için kullandığı dosya formatları grafik motorunun ihtiyaç duymayacağı detaylara sahiptir. Grafik motoru modellerin sadece üçgen bilgisine ihtiyaç duyduğu için daha basit bir dosya formatına ihtiyaç duyulmuştur. Bu yüzden Pro-Engineer ile. igs dosya formatı ile kaydedilen dosyalar daha kolay kaydedilebilecek olan .obj dosya formatına çevrilmiştir. Obj dosya formatının üçgen bilgilerini barındıran kısmının içeriği Şekil 4.43.'de gösterilmiştir. Dosyada ilk olarak şekli oluşturan nokta bilgileri yer almaktadır. Her bir satır bir noktanın x, y ve z eksenlerindeki konumunu belirtmektedir. Satır v karakteri ile başlamaktadır. Nokta bilgilerinden sonra kaplama koordinat bilgileri yer almaktadır. Her bir satır bir kaplama koordinatı belirtmektedir. Satırlar vt harfleri ile başlamakta ve kaplamaya ait s ve t koordinatlarını barındırmaktadır. Kaplama ardından normal bilgisini barındıran satırlar gelmektedir. Normal bilgileri de okunduktan sonra şekli oluşturan üçgenlerin bilgileri yer almaktadır. Üçgen oluşturan nokta, kaplama ve normal bilgisi / karakteri ile ayrılmaktadır. Obj dosya formatı modele ait hiyerarşi bilgisini barındırmanın yanı sıra materyal bilgilerinide barındırabilmektedir.



Şekil 4.43. Obj dosya formatının içeriği

Model dosyaları okunduktan sonra model yöneticisinin işleyebileceği veri yapıları içerisine yerleştirilmekte ve işlem bittiğinde bu veri yapılarına ait adres bilgisi kullanıcıya döndürülmektedir.

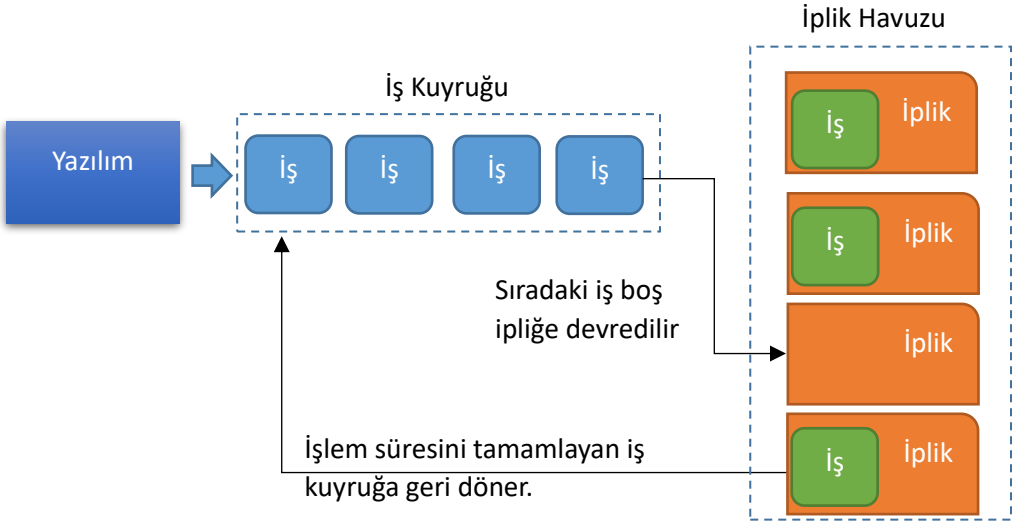
4.3.4. Paralel iş yönetim merkezi

Günümüzde bilgisayarlar birden fazla çekirdeğe sahip işlemciler tarafından kontrol edilmektedir. Geliştirilecek yazılımların da bu çekirdekleri verimli bir şekilde kullanması uygulamaların performanslarını arttıracaktır. Paralel iş yönetim merkezi birimi grafik motoru ve simülatöre ait diğer birimlerin de çalıştıkları bilgisayarın var olan çekirdeklerini eş zamanlı çalışacak şekilde kontrol etmek için tasarlanmıştır.

İplik kütüphanesi olarak C++ standart kütüphanesi kullanılmaktadır. C++ içerisinde yer alan thread sınıfını kapsayan Thread isimli bir sınıf tasarlanmıştır. Bu sınıfa yapacağı işlemler Work isimli bir nesne olarak verilmektedir. Work nesneleri de gerçekleşecek işlemi bir fonksiyon olarak almaktadır. Kullanıcı yapılacak olan işlemleri bir fonksiyon içerisine yerleştirmeli ve oluşturduğu bir Work nesnesine bu fonksiyonun adresini vermesi gerektedir.

İpliklerin kullanımını daha verimli hale getirmek için bir iplik havuzu oluşturulmaktadır. İplik havuzları içerisinde aktif iplik nesneleri barındırmaktadır. İplik nesnelerin hangi işi gerçekleştireceğine iş kuyruğu karar vermektedir. İş kuyruğu yapılacak işlerin önceliklerine göre sırayla yerleştirildiği bir kuyruk yapısıdır. İplik havuzu herhangi bir iplik boşa çıktığında sıradaki işi boş ipliğe yerleştirmektedir. İplik içerisinde işlem süresi dolan iş nesneleri görevlerini bitirmişler ise serbest bırakılırlar aksi durumda tekrardan iş kuyruğuna yerleştirilirler. Şekil 4.44.'de iplik havuzunun çalışma prensibi gösterilmektedir.

Paralel iş yönetim merkezi sayesinde grafik motoru merkezi işlemciyi çok daha verimli kullanmaktadır. Özellikle voksel haritasının oluşturulması ve ilerleyen küplerin gerçekleştirilmesinde çok iplikli programlama daha iyi sonuçlar vermektedir. Sonuçlar bölümünde bu konu daha detaylı incelenecektir.



Şekil 4.44. İplik havuzunun çalışma prensibi

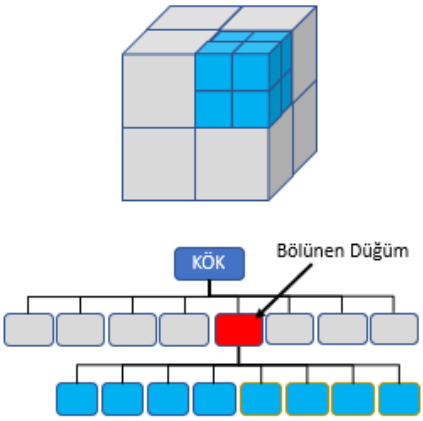
4.4. Sahne Yönetim Sistemleri

Gerçek zamanlı 3B uygulamalarında sahne elemanlarını tek bir yapı içerisinde kontrol edilmesi çarpışma testi, fizik hesaplamaları ve görüş alanı dışında kalan nesnelerin çıkartılması gibi yüksek hesap gücü gerektiren işlemlerin daha kolay optimize edilmesini sağlamaktadır [38]. Sahne kontrolünde kullanılan yapılar, 3B sanal ortamı çeşitli parçalara bölerek hesaplamaların yapılacağı model sayısını azaltmayı hedeflemektedirler.

4.4.1. Sahne yönetim ağacı

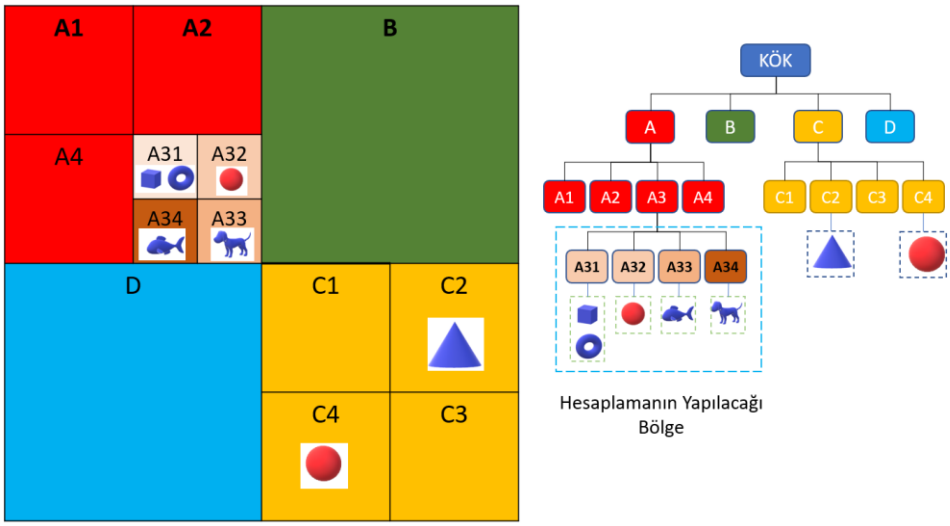
Grafik motoru 3B sahne yönetimi için sekizli ağaç veri yapısını kullanmaktadır. Bu ağaç yapısında her bir düğümün tam sekiz çocuğu bulunmaktadır. 3B sahneyi dörtlü bir ağaca yerleştirebilmek için sahne sekiz eşit parçaya bölünmektedir. Her bir parça bir düğümü temsil etmektedir. Elde edilen parçalar da yine sekiz parçaya bölünerek ağaç içerisinde yeni çocukları temsil edebilmektedir. Şekil 4.45.'de 3B sahne bir küp olarak düşünülmektedir ve sahne 3 eksende sekiz eşit parçaya bölünmüştür. Kök düğüm bütün sahneyi temsil ederken kök düğümün çocukları sekize bölünen sahnenin parçalarını temsil etmektedir. Mavi renk ile boyanmış olan sahne parçası tekrardan

sekiz eşit parçaya bölünmüştür. Bu parçayı temsil eden düğümün sekiz çocuk düğümüde bu küçük parçaları temsil edecektir.



Şekil 4.45. 3B sahnenin sekiz eşit parçaya bölünmesi ve sekizli ağaç yapısı ile temsil edilmesi

Bir bölümün bölünüp bölünmeyeceği içerisindeki model sayısına bağlıdır. Model sayısı arttıkça tekrardan bölme işlemi gerçekleştirilir. Sahnenin parçalara ayrılması çarpışma testi gibi hesaplamaların sahnede bütün modeller ile yapılması yerine olası çarpışmanın olacağı alan ile sınırlı kalmasını sağlamaktadır. Sahnenin sekizli alana bölünmesini iki boyutlu düzlem üzerinde göstermenin zorluğundan dolayı Şekil 4.46.'de bir sahnenin dörtlü ağaca dönüştürülmesine örnek verilmiştir. Şekilde de görüldüğü gibi sahnenin belirli kısımları dörde bölünmeye devam ederken diğer kısımlarda bölünme durmuştur. B ve D bölümleri içerisinde bir model olmadığı için ağaçtaki düğümlerinin de çocukları olmayacaktır. A34 parçasında bulunan balık modelinin başka bir model ile çarpışıp çarpışmadığını hesaplayacak olan matematiksel işlem sadece A3 düğümünün çocukları üzerinde çalışacaktır. Sahnedeki diğer modeller ile aynı bölgede bulunmadığı için çarpışma testinin yapılması gereksizdir. Bu optimizasyon sayesinde çarpışma testleri çok daha az sürede sonuçlanacaktır.

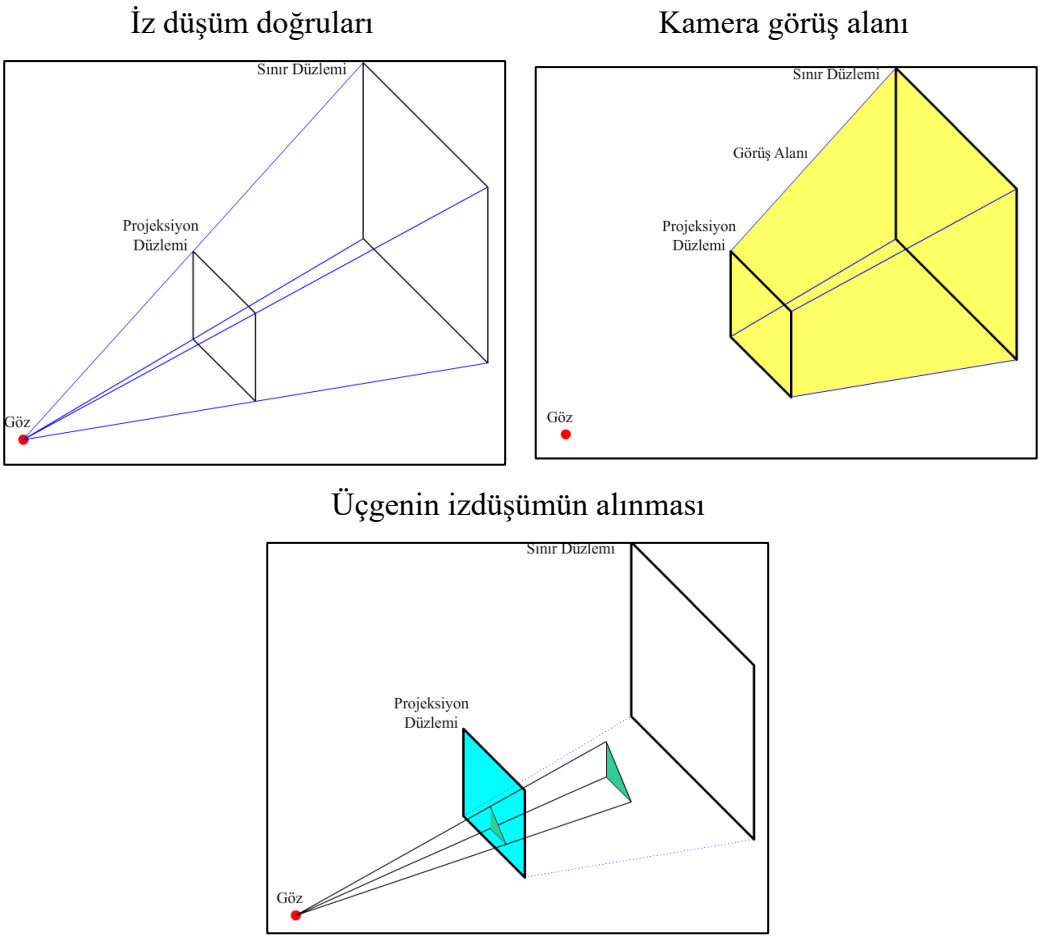


Şekil 4.46. 3B sahnenin 4 eşit parçaya bölünmesi ve dörtlü ağaç yapısı ile temsil edilmesi

Sahne oluşturulurken modeller sırayla sekizli ağaç içerisine yerleştirilmektedir. Ağaç yapısı kendisine verilen modelin merkez noktası ve kapladığı hacme bakarak ağaç içerisinde hangi düğüme yerleştireceğine karar vermektedir. Ağaçtaki her düğüm birden fazla modeli barındırabilmektedir. Çarpışma testi, fiziksel hesaplamalar ve kamera görüş alanında kalan modellerin bulunması sekizli ağaç yapısı aracılığıyla gerçekleştirilmektedir.

4.4.2. Kamera kontrol modülü

Kullanıcılar sanal ortamı grafik motoru tarafından oluşturulan bir sanal kamera aracılığıyla görecektir. Sanal kamera, 3B ortamın bir düzlem üzerine izdüşümünü almaktadır. 3B uzayda iz düşüm üç boyutlu noktaların iki boyutlu düzlem üzerine düşürülmesidir. Şekil 4.47.'da 3B sanal ortamdaki bir şeklin projeksiyon düzlemi üzerine iz düşümü gösterilmiştir. İz düşüm işlemi fotoğraf çekme işlemine benzemektedir. Şekildeki göz kameranın bulunduğu nokta olarak kabul edilmektedir. Sınır düzlemi kameranın görebileceği maksimum uzaklık iken projeksiyon düzlemi izdüşümün yapılacağı düzlemi temsil etmektedir. Bu iki düzlem arasında kalan alan kamera görüş alanı olacaktır. Görüş alanında bulunan bütün şekillerin izdüşümleri alınarak 3B dünyanın kamera tarafından alınan görüntüsü elde edilmektedir.



Şekil 4.47. 3B sanal ortamın bir düzlem üzerindeki izdüşümünün alınması

İz düşüm işlemini gerçekleştirmek için görüş alanındaki bütün şekiller iz düşüm dönüşümüne tabi tutulması gerekmektedir. Bu dönüşüm için 4x4 boyutlarında bir iz düşüm matrisi oluşturulmaktadır. İz düşüm matrisinin oluşturulması Şekil 4.48.'da gösterilmiştir. Burda r iz düşüm düzleminin genişlik ve yükseklik oranını temsil ederken Θ açısı kameranın görüş açısını temsil etmektedir. n harfi iz düşüm düzleminin kameraya uzaklığını temsil ederken f harfi ufuk veya sınır düzleminin kameraya uzaklığını temsil etmektedir.

$$\text{İz Düşüm Matrisi} = \begin{bmatrix} \frac{\cot\frac{\theta}{2}}{r} & 0 & 0 & 0 \\ 0 & \cot\frac{\theta}{2} & 0 & 0 \\ 0 & 0 & \frac{n+f}{n-f} & \frac{2nf}{n-f} \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Şekil 4.48. İz düşüm matrisi

İz düşüm dönüşümü yapılırken kameranın koordinat sisteminin merkezinde negatif z eksenine baktığı varsayılarak hesaplanmaktadır. Kameranın farklı bir koordinatta ve oryantasyonda olması iz düşüm dönüşümünü etkilemektedir. Bu yüzden iz düşüm dönüşümü yapılmadan önce sanal kameranın koordinat düzleminin merkezine çekilmesi ve negatif z eksenine bakacak şekilde döndürülmesi gerekmektedir. Sanal kamerada 3B dünya içerisinde bir nesne olduğu düşünülürse konumu ve oryantasyonu bir dünya matrisi tarafından temsil edilebilmektedir. Bu matris Şekil 4.49.'de gösterilmektedir. Kamerayı dünyanın merkezine çekmek için görüntü matrisi ile çarpmamız gerekmektedir. Bu matris kameranın dünya matrisinin tersi alınarak hesaplanmaktadır. Kameranın negatif z eksenine bakacağı düşünüldüğünde görüntü matrisi Şekil 4.49.'deki formu almaktadır. Matristeki sağ, yukarı ve bakış ifadeleri kameranın yön vektörlerini yani oryantasyonunu temsil etmektedir. Pozisyon ise kameranın bulunduğu koordinat bilgilerini temsil etmektedir.

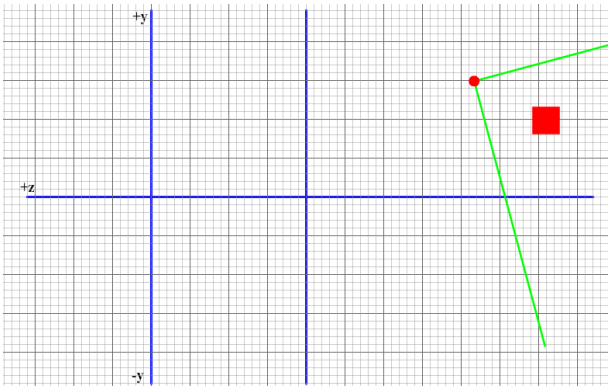
$$\text{Kameranın Dünya Matrisi} = \begin{bmatrix} Sag_x & Yukarı_x & Bakiş_x & Pozisyon_x \\ Sag_y & Yukarı_y & Bakiş_y & Pozisyon_y \\ Sag_z & Yukarı_z & Bakiş_z & Pozisyon_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Görüntü Matrisi} = \begin{bmatrix} Sag_x & Sag_y & Sag_z & -(Pozisyon \cdot Sag) \\ Yukarı_x & Yukarı_y & Yukarı_z & -(Pozisyon \cdot Yukarı) \\ -Bakiş_x & -Bakiş_z & -Bakiş_z & -(Pozisyon \cdot (-Bakiş)) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.49. Kamera dünya matrisi ve görüntü matrisi

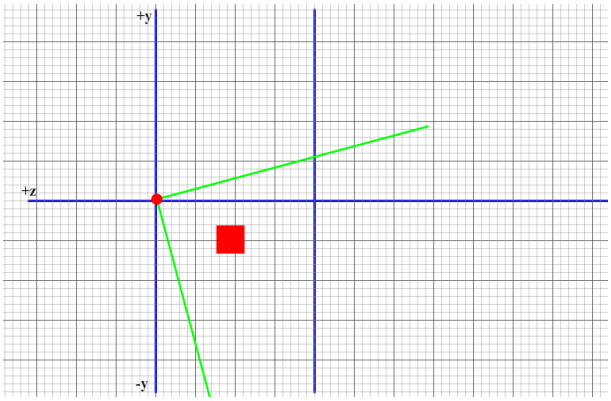
Kameranın merkeze çekilip oryantasyonunun değişmesi alınacak görüntünün bozulmasına sebep olacaktır. Buna engel olmak için kameranın uğradığı dönüşümün aynısını görüş alanındaki bütün şekillere de uygulamak gerekmektedir. Bu işlem Şekil 4.59.'de gösterilmektedir.

Kameranın Dünya üzerindeki yeri



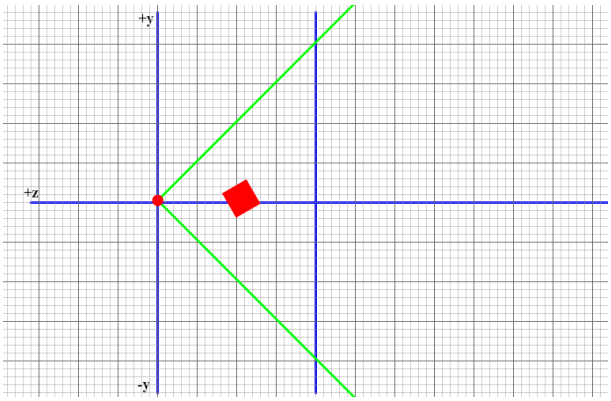
Kamera dünya üzerinde bir noktada ve bakış yönü –z den farklıdır.

Kameranın Orijine Çekiliyor



Kamera merkeze çekildiğinde görüntü alanında bozulma olmaması için bütün objelerinde aynı dönüşüme uğraması gerekir. Yandaki şekile dikkat edersek kamera açısından hiçbir şey değişmeiş gibi durmaktadır.

Kameranın Eksenleri Döndürülüyor.

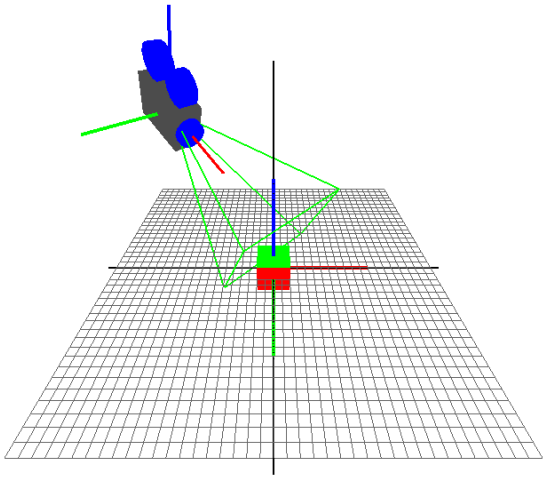


Kameranın –z yönünde bakması gerekir ve diğer eksenlerinin dünya eksenleri ile aynı yönde olması gerekir. Bunu sağlamak için kamera çeşitli döndürme işlemlerine tabi tutulur. Kameranın görüntüsünde bir değişim olmaması için bu döndürme işlemleri aynı şekilde oyun dünyasındaki bütün noktalara uygulanır.

Şekil 4.50. Kamera dönüşümünün şekiller üzerine etkisi

Grafik motoru içerisinde kamera dönüşüm ve iz düşüm işlemlerinin yönetilebilmesi için kamera kontrol modülü geliştirilmiştir. Bu modül içerisinde farklı iz düşüm tekniklerini kullanabilen kamera sınıfları bulunmaktadır. Kamera sınıflarının görevi görüntü ve iz düşüm matrislerinin elde edilmesi ve kullanıcının isteğine göre kamerayı

hareket ettirmektir. Sahne içerisinde birden fazla kamera bulunabilmektedir. Kameraların da kendilerine ait 3B modelleri bulunmaktadır. Bu sayede sahne içerisindeki kameraların konumları ve bakış yönleri kullanıcı tarafından da tesbit edilebilmektedir. Şekil 4.51.'da bir kamera modeli ve görüş alanı gösterilmektedir. Kameralar sahne kontrol ağacı içerisine de yerleştirilmektedir. Böylece görüş alanı içerisinde kalan modellerin tesbiti daha kolay hale gelmektedir.



Şekil 4.51. Kamera modeli ve görüş alanı

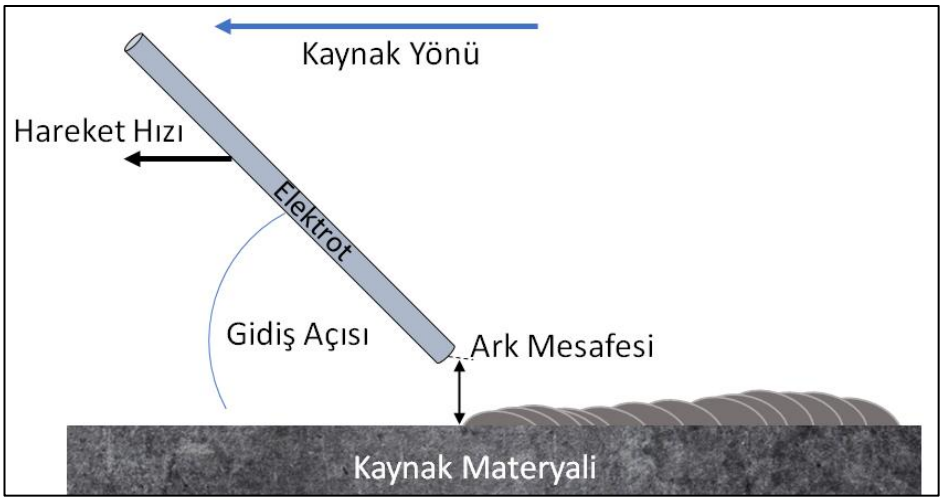
4.4.3. Sunum modülü

Grafik motorunda çizim işlemleri sunum modülü tarafından yaptırılmaktadır. Sunum modülü içerisinde de bu işlemi sunucu nesnelerine devretmektedir. Her sunucu nesnesi farklı çizim işlemleri gerçekleştirebilmektedir. Sunucu nesneleri çizdirecekleri bir sahne yönetim ağacına sahiptirler. Çizim işlemi gerçekleştirirken grafik donanımının en az sayıda durum değişikliği yapabilmesi için sunum nesneleri materyal yöneticisindeki program nesnesi, kaplama ve nokta dizisi bilgilerini kullanmaktadır. Çizim işlemi program süresi boyunca devamlı olarak gerçekleşmektedir.

BÖLÜM 5. ÖNERİLEN KAYNAK DOLGU MODELİ

Kaynak işlemlerinde kullanıcıların başarıları, materyalleri birleştirmek için oluşturulan dolguların şekli üzerinden ölçülmektedir. Sanal kaynak simülatöründeki kullanıcılar için de aynı değerlendirme sistemi geçerli olmaktadır. Bu tez çalışmasında kullanılan kaynak dolgu modelini elde etmek için literatürdeki gerçek ve sanal kaynak uygulamaları incelenmiştir. Modelin seçilmesinin ardından modeli üretmek için gerekli olan parametreler için de yine literatürdeki yer alan araştırmalar incelenmiştir.

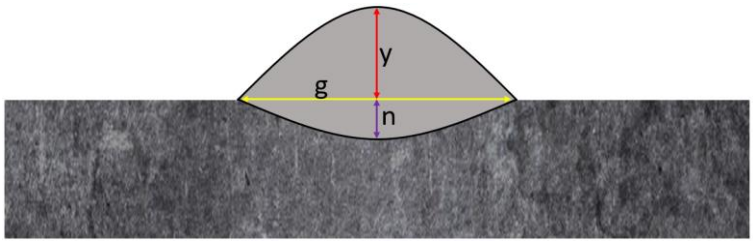
Kaynak dolgusu şeklini en çok etkileyen faktör kullanıcının el hareketleridir. Kaynakçının torcu hareket ettirme hızı, torç ile materyal arasındaki açı ve elektrodun materyale mesafesi, oluşacak kaynak dolgularının şeklini etkileyen faktörlerden bazılarıdır [39]. Şekil 5.1.'de bu parametreler kaynak işlemi üzerinde gösterilmiştir.



Şekil 5.1. Kaynak dolgu şeklinin oluşmasında etkili olan kullanıcıya bağlı parametreler

5.1. Kaynak Dolgu Şeklinin Seçilmesi

Kaynak dolgusunun 3B modelini oluşturabilmek için öncelikle temel kaynak dolgu şeklinin ne olduğuna karar vermek gerekmektedir. Bu sebeple konu üzerine literatürde bulunan çalışmalar incelenmiştir. Chambers ve arkadaşlarının [39] incelemelerinde kaynak metalinin kesiti alındığında kaynak dolgusunun parabol şekline benzediğini gözlemlemişlerdir. Wu [40] yaptığı çalışmada aynı şekilde kaynak kesitini ters çevrilmiş bir parabole benzetmiştir. Şekil 5.2.'de dolgu kesitinin iki parabol ile temsil edilmesi gösterilmektedir. Üstteki parabol kaynakçı tarafından görünürken alttaki bölüm kaynak materyaline nüfus ettiği için görünmemektedir.



Şekil 5.2. Kaynak dolgusu kesitinin parabol ile temsil edilmesi

Mavrikios [41] yaptığı çalışmada kaynak dikişini elipsoit olarak düşünmüştür. Chambers geliştirdiği simülatörde kaynak dolgu şekli olarak yine parabolü temel almıştır. Parabolü üç boyutlu olarak oluşturmak için ilgili parabolü y ekseni etrafında döndürerek bir paraboloid elde etmiştir. Paraboloid genişliğini materyal üzerindeki ısı yayılımı ile sonlu diferansiyel farklar yöntemi kullanarak hesaplamıştır. Geliştirdiği simülatör saniye 12 görüntü oluşturmaktadır. Saniyede 12 görüntü gerçek zamanlı bir uygulama için oldukça düşük bir değerdir. Ayrıca hesaplamalar merkezi işlemci üzerine yoğunlaşmış ve grafik donanımı kullanılmamıştır. Elde edilen modelin detayı hakkında da ve dolgu olarak bırakılan paraboloid şekilleri arasında geçişlerin nasıl gerçekleştirildiği konusunda da bilgi verilmemiştir. Günümüzde yüksek çözünürlüklü monitörlerin ve HMD cihazlarının sıklıkla kullanıldığı düşünülürse kaynak dolgu parametlerinin hesabını diferansiyel farklar ile yapmak performansın daha fazla düşmesine yol açacaktır.

Yapılan literatür çalışmasının ışığında bu çalışmada kaynak dolgu modeli olarak paraboloid şekli seçilmiştir. Gerçek zamanlı kaynak dolgusu elde eden diğer araştırmalardan farklı olarak paraboloidin parametreleri yapay sinir ağı kullanılarak hesaplanmaktadır. Böylece parametrelerin hesaplanması çok daha hızlı gerçekleşmekte ve daha akıcı bir simülasyon tecrübesi elde edilmiş olmaktadır. Parabolid dolgu parçalarının birleştirilmesinde yine diğer çalışmalardan farklı olarak vokselizasyon tekniği kullanılmaktadır. Bu sayede kaynak dolgularının geçişi çok daha pürüzsüz ve gerçekçi olmaktadır.

5.1.1. Kaynak parametrelerinin seçilmesi

Kaynak dolgusunun oluşmasında hangi parametrelerin etkili olacağına yönelik yapılan literatür çalışmasında kaynak hızı, akım, gerilim, ark mesafesi, gidiş açısı, tel besleme hızı parametlerinin sıklıkla kullanıldığı gözlemlenmiştir. Yapılan araştırma sonucunda MIG (Metal Inert Gas) kaynak yönetiminde kaynak dolgusunu etkileyen parametler tablo 5.1.'de gösterilmektedir. Tabloya göre MIG kaynak yönteminde kaynak dolgusuna en çok etki eden parametrelerin kaynak hızı, akım ve gerilim olduğu tespit edilmektedir. Buna ek olarak 109M087 kodlu Tübitak projesinde görev alan kaynak işlemi konusunda uzman araştırmacı akademisyenler kaynak dolgu şekline en çok etki eden parametlerin ilk olarak kaynak hızı, ardından ark mesafesi ve kaynak gidiş açısı olduğuna karar vermişlerdir [42,43]. Yapılan araştırmalar sonucunda, kaynak dolgusu şeklini etkileyecek parametreler akım, kaynak ilerleyiş hızı, ark mesafesi ve kaynak açısı olarak seçilmiştir.

Tablo 5.1. MIG kaynak yönteminde kaynak dolgusunu etkileyen parametreler

Yayın	Kaynak Hızı	Akım	Gerilim	Gidiş Açısı	Ark Mesafesi
Son[44]	+	+	+	-	-
Sreeraj[45]	+	+	-	+	+
Karadeniz[46]	+	+	+	-	-
Kim[47]	+	+	+	-	-
Das[48]	+	+	+	-	-
Gazvinlo[49]	+	+	+	-	-
Tewari[50]	+	+	+	-	-
Shoeb[51]	+	-	+	-	-

5.1.2. Yapay sinir ağının yapısı

Bu çalışmada, kaynak dolgusunu temsil eden paraboloitlere ait parametreleri üretmek için bir yapay sinir ağı modeli oluşturulmuştur. Ağın giriş parametrelerini daha önceden de belirtildiği gibi kaynak dolgu şeklini en çok etkileyen akım, ark mesafesi, kaynak ilerleyiş hızı ve gidiş açısı oluşturmaktadır. Çıkış parametrelerini ise genişlik, yükseklik ve nufiziyet miktarları oluşturmaktadır. Tablo 5.2.'de de görüldüğü gibi yapılan çalışmaların çoğunda çıkış parametresi olarak genişlik, nüfuziyet ve yükseklik elde edilmiştir.

Tablo 5.2. MIG kaynak yönteminde elde edilen çıkış parametreleri

Yayın	Genişlik	Yükseklik	Nüfuziyet
Son[44]	+	-	-
Sreeraj[45]	+	+	+
Karadeniz[46]	+	+	+
Kim[47]	+	+	+
Das[48]	-	-	+
Gazvinlo[49]	-	-	+
Tewari[50]	-	+	+
Shoeb[51]	+	+	+

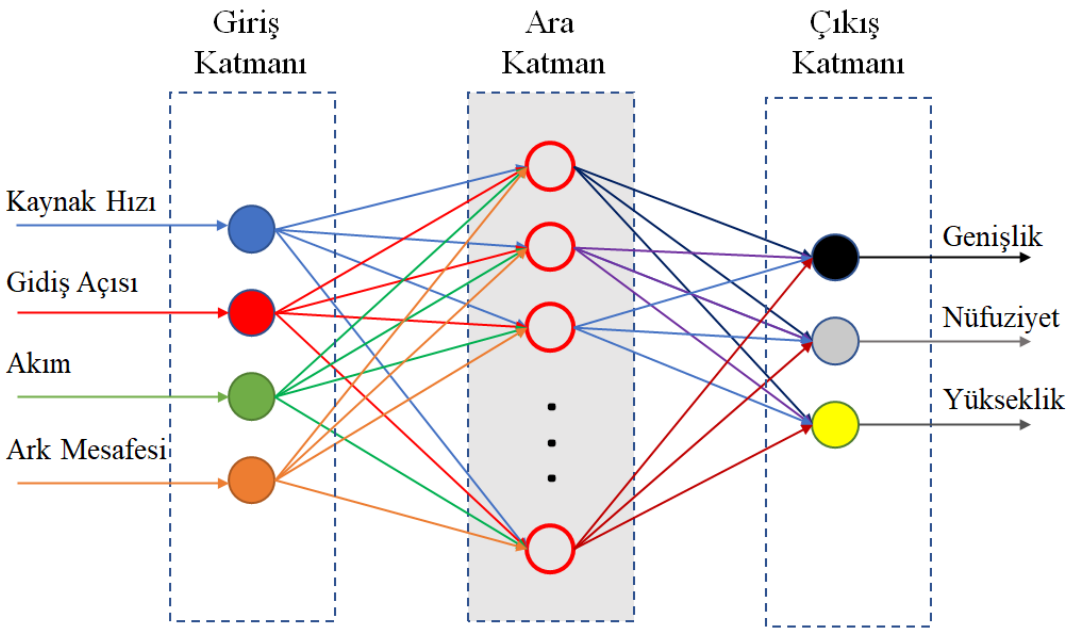
Geliştirilen yapay sinir ağının modeli Şekil 5.3.'de gösterilmektedir. Çok katmanlı bir yapıya sahip olan ağda danışmanlı öğrenme stratejisi kullanmaktadır. Buna göre danışman gerçek testlerden alınmış veri setlerini (giriş ve çıkış) yapay sinir ağına vermektedir. Böylece geliştirilen ağın ürettiği veriler ile beklenen veriler arasındaki farkın yani hatanın azaltılması amaçlanmaktadır. Geri yayılımcı çalışan ağda hata miktarını azaltmak için düğümlerin ağırlıkları devamlı olarak değiştirilmektedir.

Yapay sinir ağının eğitimi için Sreeraj [45] ve arkadaşlarının yaptığı 32 tane gerçek kaynak uygulamasından alınan veriler kullanılmıştır. Verilerin %15'i test, %15'i doğrulama ve %70 eğitim için kullanılmıştır. Ara katmandaki nöron sayısı deneme yöntemi ile tespit edilmiştir. Yapılan denemeler ve sonuçlarını tablo 5.3.'de gösterilmektedir. Buna göre ara katmandaki nöron sayısının 8 olmasına karar verilmiştir. Yapay sinir ağı tasarlanırken eğitim fonksiyonu olarak TrainLM

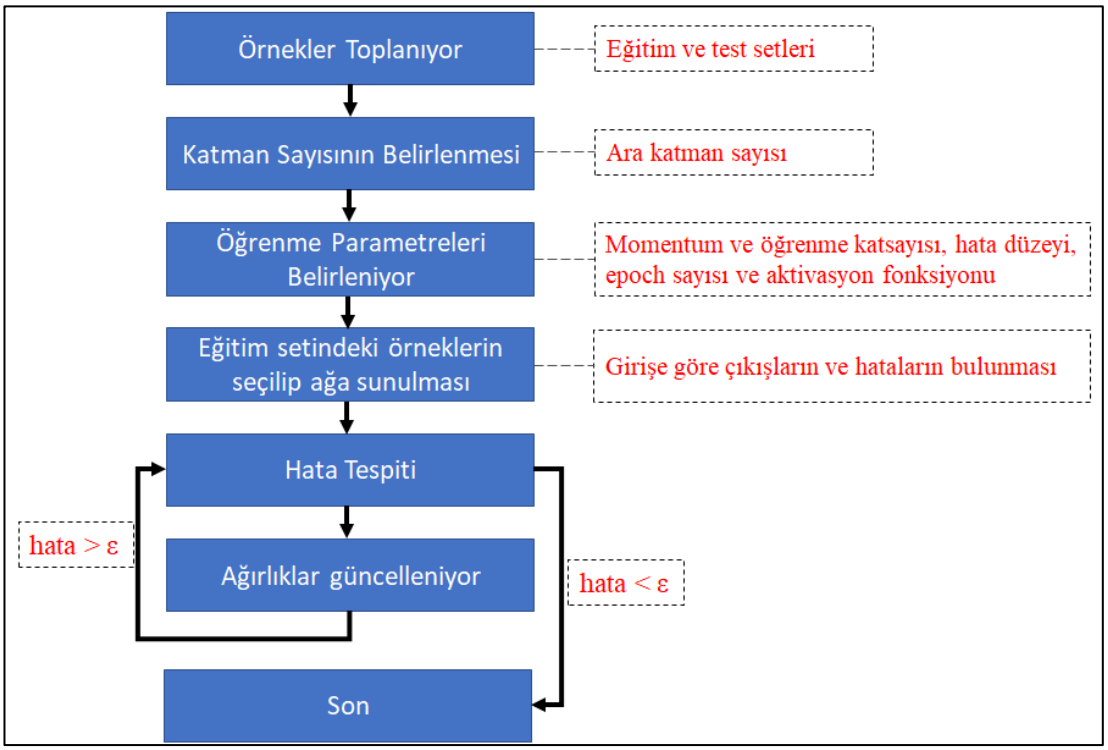
(Levenberg- Marquart Geri yayılım fonksiyonu) seçilmiştir. Matlab programının resmî sitesinde TrainLM fonksiyonunun en hızlı geri yayılım fonksiyonlarından birisi olarak belirtmiştir. TrainLM fonksiyonunun etkisini inceleyen başka çalışmalar da mevcuttur [52-55]. Transfer fonksiyonu olarak denemelerde en iyi sonuçları veren LogSig fonksiyonu kullanılmıştır. Geliştirilen yapay sinir ağının çalışma prensibi Şekil 5.4.'de gösterilmektedir.

Tablo 5.3. MIG Kaynağı için ara katman nöron sayısı belirleme denemeleri

Deney No	Ağ Yapısı	Transfer Fonksiyonu	Epok Sayısı	Eğitim	Doğrulama	Test	Ortalama
1	4/4/3	LOGSIG	1000	0,97372	0,99121	0,97790	0,97603
2	4/6/3	LOGSIG	1000	0,97302	0,95009	0,97399	0,96962
3	4/8/3	LOGSIG	1000	0,98602	0,97848	0,96159	0,98253
4	4/10/3	LOGSIG	652	0,98808	0,9242	0,82398	0,94719
5	4/12/3	LOGSIG	596	0,90988	0,98008	0,97169	0,93978
6	4/14/3	LOGSIG	583	0,98198	0,99189	0,99194	0,98179
7	4/16/3	LOGSIG	521	0,98437	0,93519	0,99146	0,97962
8	4/18/3	LOGSIG	213	0,98001	0,96049	0,98913	0,97986
9	4/20/3	LOGSIG	221	0,93283	0,95398	0,95816	0,95978
10	4/4/3	PURELIN	14	0,90005	0,93386	0,96111	0,91008
11	4/6/3	PURELIN	10	0,92263	0,92398	0,77969	0,89798
12	4/8/3	PURELIN	11	0,90601	0,98225	0,92786	0,92789
13	4/10/3	PURELIN	12	0,93702	0,95501	0,81577	0,91997
14	4/12/3	PURELIN	12	0,94488	0,89674	0,83713	0,91927
15	4/14/3	PURELIN	13	0,92855	0,91194	0,92584	0,91991
16	4/16/3	PURELIN	15	0,91759	0,95412	0,91741	0,91197
17	4/18/3	PURELIN	11	0,92509	0,94168	0,94034	0,92028
18	4/20/3	PURELIN	14	0,90909	0,92968	0,93934	0,91922
19	4/4/3	TANSIG	1000	0,97169	0,99689	0,97837	0,97379
20	4/6/3	TANSIG	1000	0,97668	0,97589	0,93568	0,96689
21	4/8/3	TANSIG	1000	0,97545	0,97671	0,96139	0,97005
22	4/10/3	TANSIG	1000	0,97086	0,97336	0,97883	0,97241
23	4/12/3	TANSIG	1000	0,99616	0,96799	0,87487	0,97003
24	4/14/3	TANSIG	956	0,98958	0,94068	0,98235	0,97809
25	4/16/3	TANSIG	914	0,98457	0,93803	0,92212	0,96864
26	4/18/3	TANSIG	907	0,98357	0,93986	0,97786	0,97002
27	4/20/3	TANSIG	850	0,96467	0,92803	0,92202	0,95986



Şekil 5.3. Kaynak dolgusu şeklinin parametrelerini belirleyecek YSA yapısı

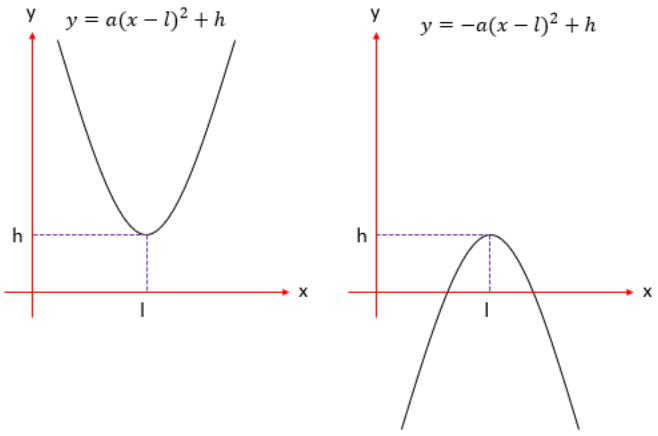


Şekil 5.4. Seçilen yapay sinir ağı modelinin çalışma prensibi

5.1.3. Paraboloid şekli

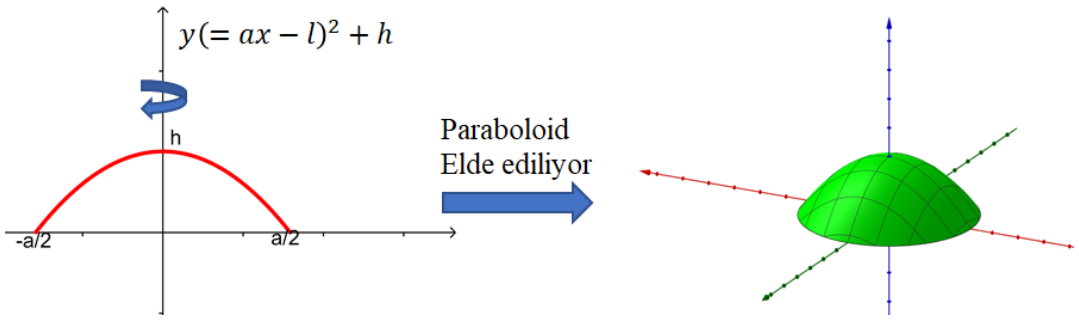
Parabol ikinci dereceden bir fonksiyonun grafiğidir. Denklem 5.1.'de parabol denklemi gösterilmektedir. Burada l ve h parabolün başladığı koordinatı temsil etmektedir. Denklemin grafiği Şekil 5.5.'de gösterilmektedir. Grafiğin yönünü a sabiti belirtmektedir. Bu değer negatif olduğunda parabol -y yönünde yani ters olacaktır.

$$y = a(x - l)^2 + h \tag{5.1}$$



Şekil 5.5. Örnek parabol grafikleri

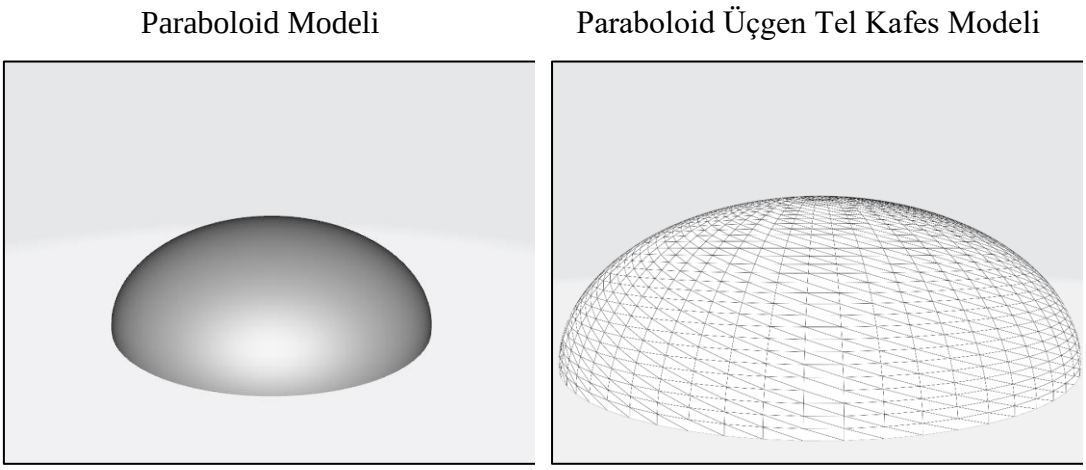
Kaynak dolgusunu temsil edecek grafik -y yönünde olacağından a değeri bu çalışmada daima negatif değere sahip olmaktadır. Buna ek olarak parabolün x koordinatında daima 0 noktasından başladığı varsayılmaktadır. Şekil 5.6.'da kaynak dolgusu için kullanılacak olan parabol grafiği gösterilmektedir. Parabol grafiğinden üç boyutlu bir şekil elde edebilmek için parabol y ekseninde 360 derece döndürülüp bir paraboloid elde edilmektedir.



Şekil 5.6. Kaynak dolgusunda kullanılacak olan parabol grafiği

İki boyutlu parabol denkleminden paraboloid denkleminin elde edilmesi denklem 5.2.'de gösterilmiştir. Bu denklem kullanılarak paraboloid modelini oluşturan noktalar hesaplanabilmektedir. Denklem içerisinde nokta koordinatlarını bulmak için x ve z koordinatları, belirli aralıklarla değerler verilerek bu koordinatlara karşılık gelen y koordinatları da hesaplanmaktadır. Elde edilen noktalar kullanılarak da modeli oluşturan üçgenler üretilebilmektedir. Şekil 5.7.'de paraboloid modeli ve modeli oluşturan üçgenler gösterilmektedir.

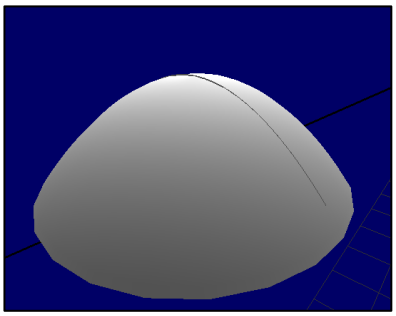
$$y = \frac{x^2}{a^2} + \frac{z^2}{a^2} + h \tag{5.2}$$



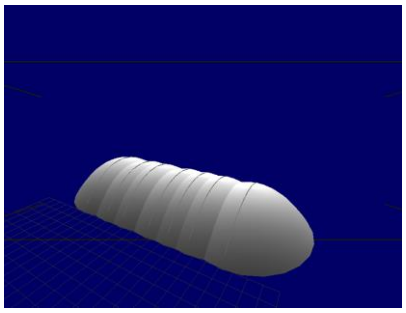
Şekil 5.7. Paraboloid denklemi kullanılarak elde edilen 3B modeller

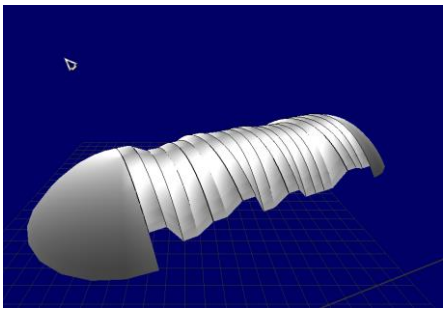
Geliştirilen parabolid modeli kullanılarak simülatör üzerinde yapılan denemelerde birden fazla modelin birleştirilmesinde problemler yaşandığı gözlemlenmiştir. Oluşturulan kaynak dolguları arasındaki geçişin keskin ve pürüzlü olması elde edilen kaynağın gerçekçi görünmesine engel olmaktadır. Şekil 5.8.'de elde edilen görüntüler gösterilmektedir.

Tek Kaynak Dolgu Modeli



Sıralı Kaynak Dolgu Modeli





Şekil 5.8. Paraboloid Dolgu Modellerin Birleştirilmesi

5.2. Geliştirilen Kaynak Dolgu Modeli

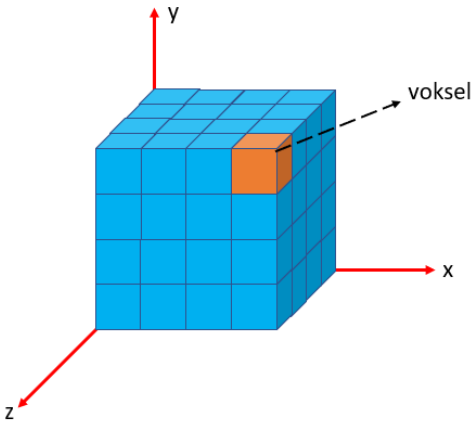
Bu tez çalışmasında kaynak dolgu modelleri arasındaki geçişleri pürüzsüzleştirmek ve daha gerçekçi bir kaynak dolgu dizisi elde etmek için dolgu şekline vokselleştirme tekniği uygulanarak vokselleştirilmiş paraboloid şekli üretilmiştir. Üretilen vokseller yürüyen küpler algoritması kullanılarak üçgen eşyüzeyler oluşturulmuştur.

5.2.1. Vokselleştirme

Bilgisayar grafiği teknolojisindeki gelişmeler ile birlikte ortaya çıkan üç-boyutlu bilgisayar grafiğinde kullanılan 3B geometrik nesneleri tanımlamak için genellikle vektör grafiklerinin 3B veri gösterimine adaptasyonundan doğmuş B-rep gibi poligon ızgaralar, bezier eğrileri ve NURBS gibi parametrik yüzeyler ve yapısal katı geometri kullanılır. Ancak, vektör grafiklerine alternatif olarak kullanılan piksel-temelli model gösterimine karşılık üç boyutlu bilgisayar grafiklerinde geometrik nesneleri modellemek için iki-boyutlu pikselin üç-boyutlu karşılığı olan voksel kavramı geliştirilmiştir [56].

Voksel, en basit haliyle üç-boyutlu uzayda merkezi (x,y,z) olan bir birim küp olarak tanımlanabilir. Üç-boyutlu bilgisayar grafiğinde modelin dış yüzeyine ait konum bilgisi tutan yüzey gösterimi yaklaşımlarının aksine, bir voksel karşılık geldiği birim hacme ait renk, sıcaklık, yoğunluk, opaklık, sertlik vb. gibi gerçek geometrik nesnelere ait olan fiziksel özellikleri de barındırabilir. Şekil 5.9.'da bir voksel örneği gösterilmektedir. Şekilde uzay x, y ve z eksenlerinde 4 eşit parçaya bölünmüştür. Buna göre toplamda 4x4x4 yani 64 adet voksel bulunmaktadır. Uzayın daha fazla sayıda parçalara ayrılması için voksellerin boyutlarının da küçülmesi gerekmektedir.

Bu çalışmada kaynak dolgusuna ait yüzey bilgisini içeren üçgen ızgara modelinin tüm hacmine ait bilgiyi elde etmek amacıyla vokselleme işlemi uygulanır. Buradaki amaç, görüntülemede kullanılan üçgen kafes modeli üzerinde kaynak işlemlerini yapabilmek için arka planda bu modele ait bir voksel haritası elde etmektir. Kullanıcının kaynak dolgusu yapmak amacıyla uyguladığı hareket verilerine uygun olarak oluşturulan hacim voksel yapısının her bir vokseline karşılık gelen eş-hücrelerinin hesaplanması, ağaç veri yapısının oluşturulması ve son olarak da kaynak dolgu yüzeyinin yeniden oluşturulması amacıyla yürüyen küpler algoritmasının uygulanması gerekmektedir.

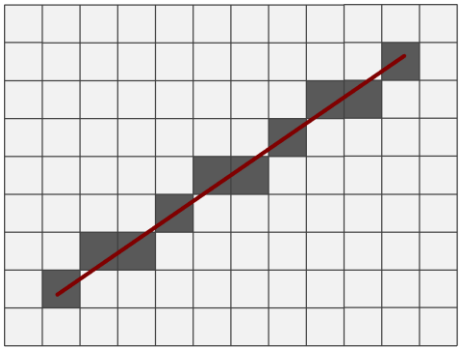


Şekil 5.9. Voksel haritası

Üç boyutlu geometrik yüzey gösterimlerine alternatif olarak ortaya çıkan ve hacim grafiklerinin temelini oluşturan voksel verisini elde edebilmek için "vokselleştirme" olarak adlandırılan ve nesnenin 3B ayrık bir ızgara içerisinde ayrık vokseller kümesi olarak gösterilmesi olarak tanımlanan bir dönüşüm işlemi yapılması gerekir.

Kaufman ve arkadaşları "Hacim Grafikleri" isimli makalede [57] vokselleştirme için şu cümleleri söylemiştir: "...Vokselleme olarak adlandırılan bu adım geometrik nesnelerin sürekli gösterimlerinden sürekli nesneye en yakın vokseller kümesine dönüştürülmesi olarak ele alınır. 2B nesneleri pikselleştiren tarama-dönüşümü (scan-conversion) işlemini taklit ettiğinden dolayı, 3B tarama dönüşümü olarak da anılır…".

Kaufman ve Shimony tarafından geliştirilen literatürdeki ilk vokselleştirme çalışmasında, algoritma doğrular, daireler, poligonlar, çok yüzlü ve quadratik nesneler gibi çeşitli yüzeyler ve nesneleri vokselleştirir [56]. Bu çalışmada, iki boyutlu vektör grafiklerinden raster grafiğe geçiş için yapılan 2B tarama-dönüşümü işleminin [58,59] üç boyutlu uzantısı gerçeklenmiştir. Şekil 5.10.'da 2B tarama-dönüşümü algoritmalarından birisi olan ve Kaufman ve Shimony'nin önerdikleri vokselleştirme çalışmasına ışık tutan Bresenham'ın doğru çizme algoritması gösterilmektedir [60].



Şekil 5.10. 2B tarama dönüşümü örneği

Voksel veri yapısı ve bu yapıyı oluşturmak amacıyla kullanılan vokselleme işlemi mikro ölçekteki görüntüleme problemlerinin çözümünde gün geçtikçe artan bir kullanım alanı bulmaktadır. Literatürde tıp/dişçilik [61-71], sanal heykeltıraşlık [72-78] ve mühendislik [79-83] alanlarında yapılmış vokselleme uygulamaları görülmektedir.

Agus ve arkadaşlarının haptic bir cihaz kullanılarak gerçek-zamanlı kemik kesme/delme işlemi yaptığı çalışmada voksellenmiş CT/MRI hasta verisi kullanılmıştır [62]. Daha sonra, Lai ve arkadaşları tarafından kemiğe ait mikro yapıyı inceleyen çalışmasında kemik yapısının detaylarının arttırılması gerektiği vurgulanmış [63], Eriksson ve arkadaşları tarafından da cerrahi frezeleme işlemini simüle ederek cerrahların tümörlü yapıların çıkartılması gibi karmaşık kafatası kemiği işlemlerini daha kolay gerçekleştirebilmeleri amacıyla bir haptic ve sanal gerçeklik simülatörü geliştirilmiştir [64,65].

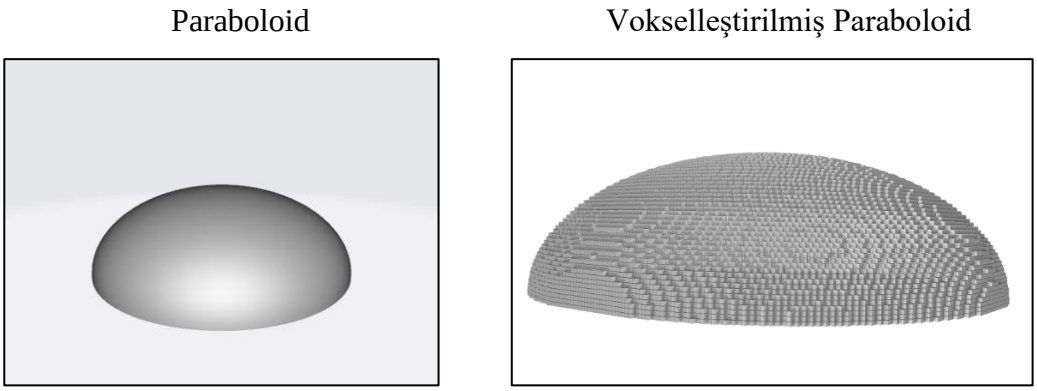
Tıp alanında yapılan ve yukarıda sıralanan tüm bu çalışmaların yanı sıra, literatürde modelleme ve tasarım alanında eğitim amacıyla yüksek çözünürlüklü voksel verisi kullanılarak önerilen sanal heykeltıraşlık çalışmaları da yer almaktadır [72-76]. Örneğin Ferley, detay seviyesi yüksek voksel temelli modeller oluşturmak amacıyla bir vokselin 26 parçaya bölündüğü bir yapı önermiştir [72-73].

Williams ve diğerleri tarafından önerilen çalışmada üç-boyutlu sanal nesneyi yontabilmek amacıyla voksel içeren veri yapısı üzerinde hacim eksiltme işlemi uygulanmıştır [77]. Çalışmada, tomografi görüntülerinden kullanılarak elde edilen

voksel temelli veriler, modele ait yüzey bilgisini görüntülemek amacıyla üçgen ızgaralara dönüştürülmüştür. Bu çalışmaya, O'Neill ve diğerleri tarafından, yontma işlemini gerçekleştirmek amacıyla nesne yüzeyine dokunmak için heptik bir cihaz kullanımını eklemişlerdir [78].

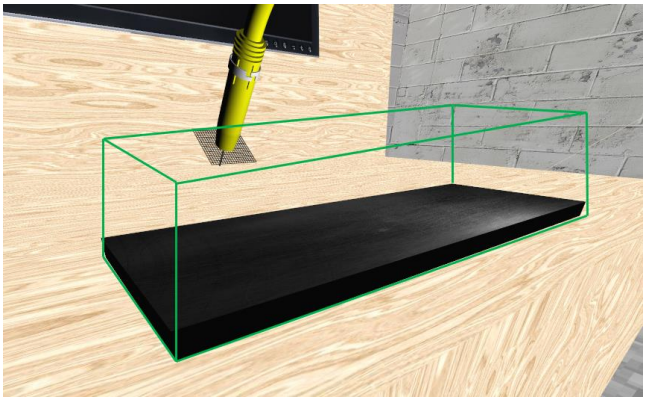
Çit, önerdiği sanal heykeltıraşlık sistemi uygulamasında, üzerinde çalışacağı 3B modelleri vokselizasyona tabi tutarak gerçek zamanlı yontma işlemi gerçekleştirmiştir [84]. Daha sonra vokselizasyon işlemini hızlandırmak için OpenCL kütüphanesi kullanılarak hesaplama süreleri kısaltılmıştır. [82].

Bu tez çalışmasında kaynak sırasında üretilen dolgu modelleri vokselleştirmeye tabi tutulmaktadır. Şekil 5.11.'da bir paraboloid kaynak dolgusunun vokselleştirilmiş hali gösterilmektedir.



Şekil 5.11. 2B paraboloid kaynak dolgusunun vokselleştirilmiş gösterimi

Paraboloid modelinin voksellere çevrilebilmesi için öncelikle vokselizasyona tabi tutulacak alanın belirlenmesi gerekmektedir. Bu alan bir küp şeklinde seçilebileceği gibi bir kutu yani dikdörtgenler prizması şeklinde de olabilir. Vokselleştirme işlemi sadece kaynak dolgusu ve nüfuz edeceği kaynak materyalini etkileyeceği sadece bu alanı içine alacak bir kutu vokselleştirme alanı olacak seçilecektir. Şekil 5.12.'de vokselleştirme alanı gösterilmektedir. Vokselleştirme alanının iki uzunluğu materyalden alınırken yüksekliği kullanıcı tarafından alınmaktadır.



Şekil 5.12. Vokselleştirme alanı

5.2.2. Voksel haritası ve sekizli ağaç veri yapısı

3B modellerin vokseller ile temsil edilmesi, özellikle yüksek çözünürlüklü haritalarda büyük hafıza ve işlem gücü gerektirebilmektedir. Yüzey oluşturma algoritmasının voksel haritasında uygulandığı alanları sınırlandırmak ve erişimi hızlandırmak simülatörün görüntü elde etmesini de kolaylaştıracaktır. Bu yüzden voksel haritasını temsil etmek için, literatürde sıklıkla kullanılan ve uzayı belirli sayıda eşit parçalara bölüp kontrol eden ağaç veri yapıları kullanılmaktadır.

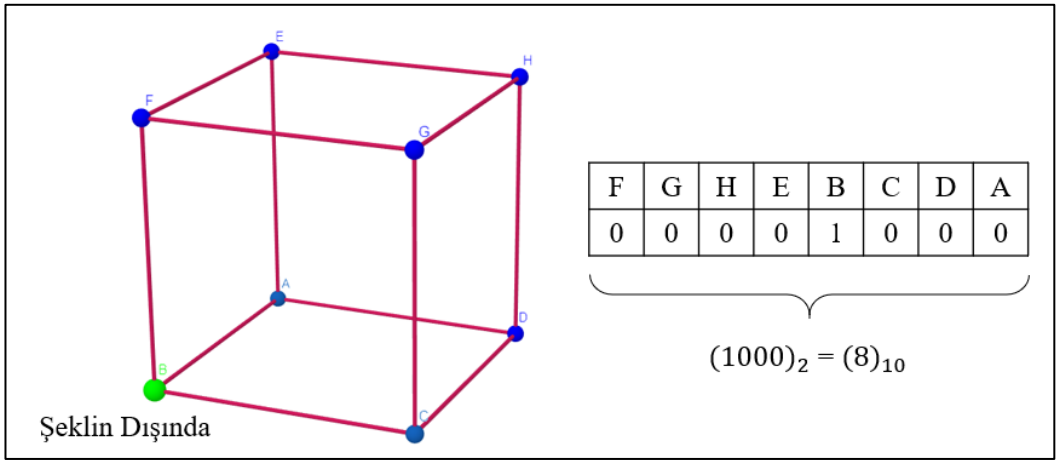
Barentzen [85] ve Ho [75] çalışmalarında yüksek çözünürlüklü voksel haritalarında daha az hafıza kullanılması için sekizli ağaç veri yapısının kullanılmasını önermişlerdir. Çit [84], yaptığı sanal heykeltıraşlık çalışmasında geliştirdiği hash tabanlı sekizli ağaç veri yapısının yüksek çözünürlüklü voksel haritalarında daha hızlı erişim sağlayarak yüzey oluşturma algoritmalarının işlem süresini kısaltmıştır. Bu çalışmalar göz önünde bulundurularak kaynak dolgusunu temsil eden voksel haritasının Çit'in hash tabanlı sekizli ağaç veri yapısında saklanmasına karar verilmiştir.

5.2.3. Voksel haritasından üçgen yüzeylerin oluşturulması

3B modellerin yüzeyleri üçgenlerden oluşmaktadır. Vokselleştirilmiş kaynak dolgusunun da üçgenlerden oluşan bir yüzeyinin elde edilmesi gerekmektedir. Vokselleştirilmiş modellerin üçgenleştirilmesi için genelde delaunay üçgenleme ve

yürüyen küpler algoritmaları kullanılmaktadır [86]. Yürüyen küpler algoritması delaunay algoritmasına göre daha basit bir yapıdadır [87]. Yüzey elde etmekte kullanılan algoritmalar üzerinde yapılan diğer araştırmalar da incelendiğinde yürüyen küpler algoritmasının en sık kullanılan algoritma olduğu görülmektedir [87-91].

Yürüyen küpler algoritmasında her bir voksel yerini bir veya daha fazla üçgene bırakacaktır. Vokselleri temsil edecek üçgenler voksele ait noktaların temsil ettiği modelin içinde olup olmadığına göre değişmektedir. Nokta modelin içerisinde kalıyor ise 0, dışında kalıyor ise 1 değeri kullanılmaktadır. Yürüyen küpler algoritmasının çalışması sırasında iki adet tabloya ihtiyaç duyulmaktadır. Bunlardan ilki üçgenin oluşmasını sağlayacak köşe kombinasyonlarını içeren tablodur. Köşe tablosundaki indis değeri ise voksele ait noktaların modelin içerisinde olup olmamasına göre hesaplanmaktadır. Noktaların değerleri Şekil 5.13.'deki gibi 8 bitlik bir sayının bitleri şeklinde sıralanacak ve elde edilecek onluk tam sayı değeri köşe tablosundaki indisi temsil edecektir. Buna göre köşe tablosunun 8. elemanındaki değer üçgen kombinasyonlarını oluşturacak köşelerin tutulduğu köşe tablosuna indis olarak kullanılacaktır.

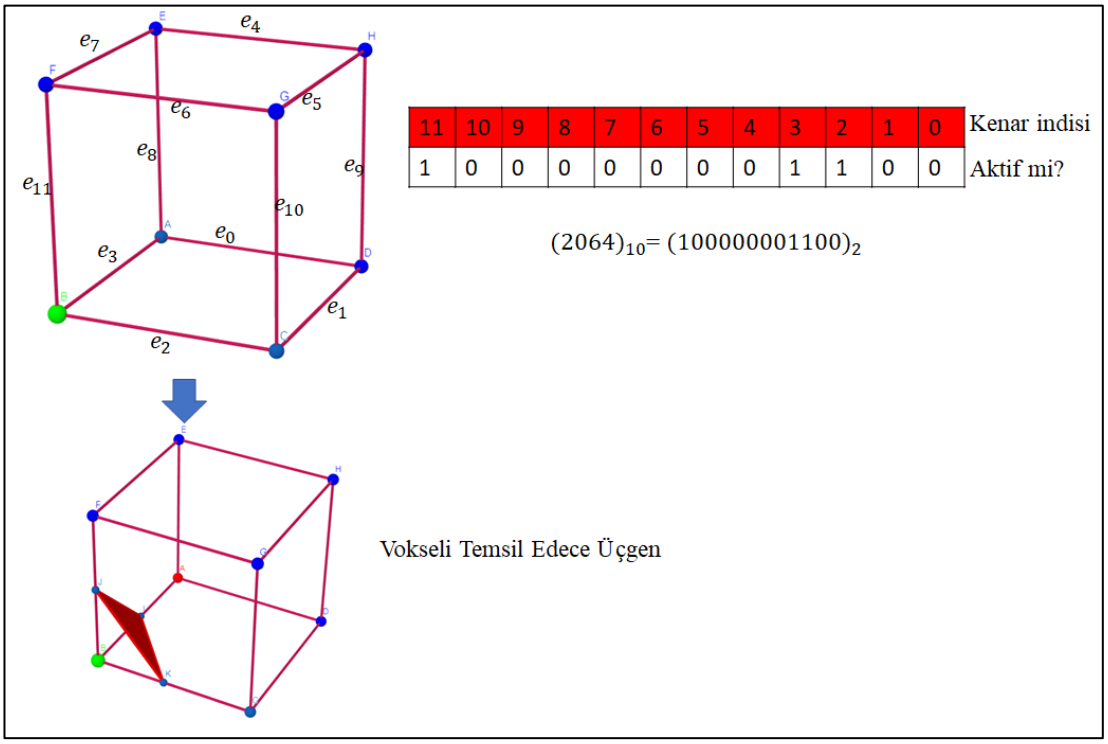


Şekil 5.13. Voksele ait köşe tablosundaki indisin hesaplanması

Algoritmada kullanılan diğer tablo üçgenleri oluşturan kenarların sayısal değerlerini tutmaktadır. Şekil 5.12.'den yola çıkarak kenar tablosunun 8. indisine ulaşılmaktadır. 8. elemanın değeri 2064 tam sayı değeridir. Bu değerin bitleri vokselin hangi köşelerinin üçgen oluşturmak için kullanılacağını belirtmektedir. Şekil 5.14.'de

vokseli oluşturan kenarların değerleri gösterilmektedir. Şekle göre 2, 3 ve 11. köşelerin orta noktaları kullanılarak bir üçgen oluşturulacaktır.

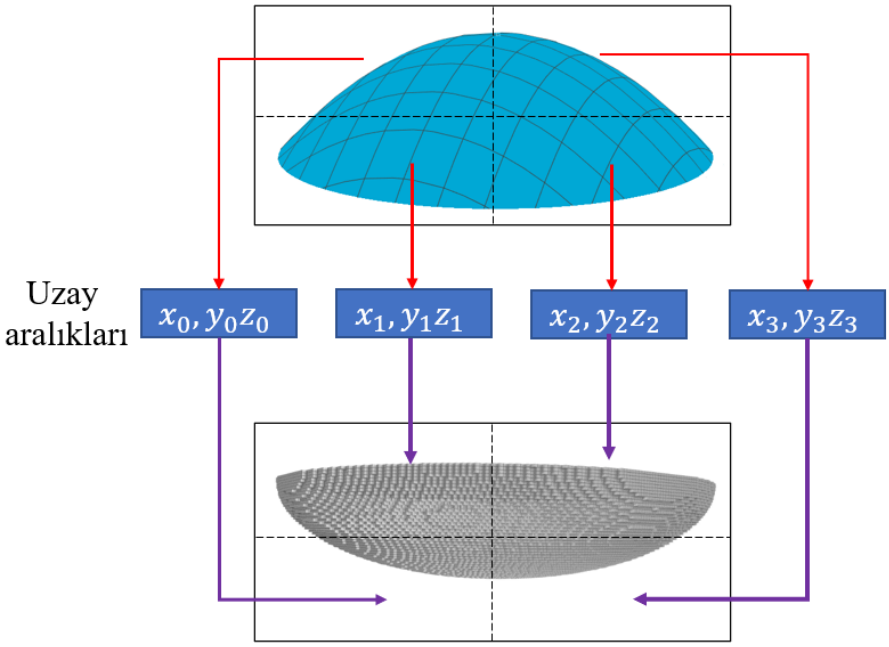
Kaynak süresi boyunca yeni eklenen dolgu modelleri öncelikle vokselleştirilmekte hemen ardından yürüyen küpler algoritması kullanılarak üçgen yüzeyleri elde edilmektedir. Simülatör sadece üçgen yüzeyleri çizeceği için kaynakçı arka planda gerçekleşen işlemlerden habersiz olacaktır.



Şekil 5.14. Voksele ait kenarları bulma ve vokseli temsil eden üçgen

5.2.4. Algoritmaların paralelleştirilmesi

Vokselleştirme ve yürüyen küpler algoritması paralel çalıştırılmaya uygundur. Kaynak dolgu modelini temsil eden paraboloid x, y ve z eksenleri üzerinden eşit parçalara bölünürse her bir parçanın vokselleştirilmesi ayrı bir işlemci çekirdeği tarafından gerçekleştirilebilir. Şekil 5.15.'de kaynak dolgusunu temsil eden paraboloid şeklinin eşit iş parçalarına bölünmesi gösterilmiştir. Grafik motoru içerisinde bulunan paralel işlem birimi, kendisine verilen görevleri farklı çekirdeklerin gerçekleştirmesini sağlamaktadır.



Şekil 5.15. Vokselleştirme işleminin 4 eşit iş parçasına ayrılması

Vokselleştirme işleminin farklı iplik sayılarına göre süreleri Tablo 5.4.'de verilmektedir. Voksel haritasının oluşturması iplik sayısı arttıkça azalmaktadır. İplik sayısı simülatörün çalıştığı bilgisayarın çekirdek sayısını geçtiğinde durum tersine dönmektedir. Aynı şekilde tablo 5.5.'de vokselleştirilmiş kaynak dolgusu üzerine yürüyen küpler algoritmaların uygulanması için gereken süreler verilmiştir. Simülatörün gerçekçiliğini arttırmak için kullanıcı hareketleri ile sanal ortamın güncellenmesi akıcı olması gerekmektedir. Bunun için simülatörün saniyede en az 60 görüntü elde etmesi gerekmektedir. Her bir görüntünün yaklaşık 16 milisaniyeden az sürede hazırlanması gerekmektedir. Elde edilen vokselleştirme süreleri ile sanal kaynak simülatörü gerçek zamanlı olarak çalışabilmektedir. Sonuçlar ve öneriler bölümünde simülatörün ortalama görüntü sayısı verilmektedir.

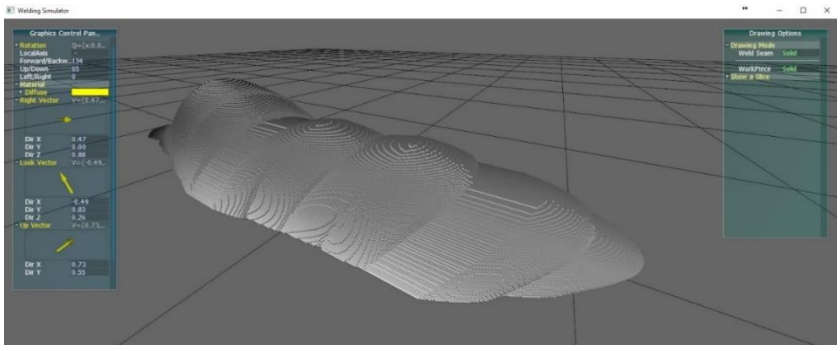
Tablo 5.4. Bir kaynak dolgusunu vokselleştirmek için geçen süre

Çözünürlük	Voksel Sayısı	Bir Dolguyu Vokseketirmek İçin Geçen Süre (ms)			
		İplik Sayısı			
		1	2	4	8
128x64x32	214	0,112	0,021	0,013	0,015
256x128x64	538	0,412	0,234	0,162	0,173
512x256x128	1394	1,212	0,782	0,421	0,512
1024x512x256	3782	3,242	2,123	1,782	1,923

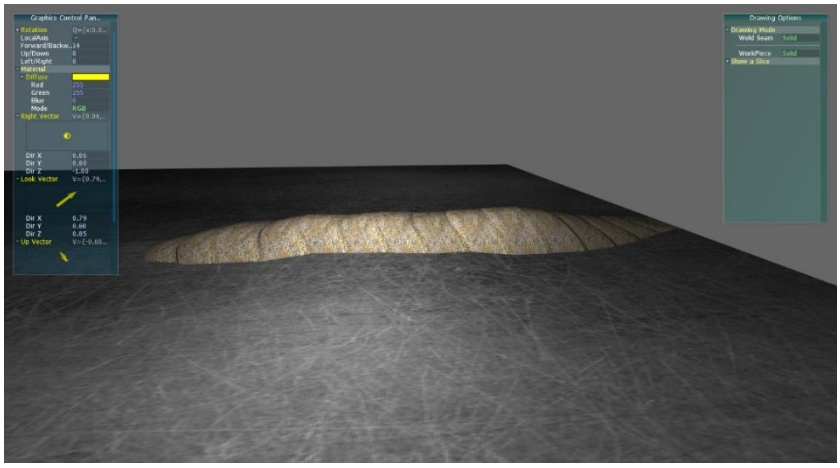
Tablo 5.5. Vokselleştirilmiş bir kaynak dolgusundan üçgen yüzey elde etmek için geçen süreler.

Çözünürlük	Vokselleştirilmiş Dolgudan Üçgen Yüzey Oluşturmak İçin Geçen Süre (ms)			
	İplik Sayısı			
	1	2	4	8
128x64x32	0,181	0,123	0,073	0,075
256x128x64	0,612	0,325	0,171	0,184
512x256x128	2,243	1,482	0,822	0,892
1024x512x256	5,132	3,313	2,037	2,165

Şekil 5.16.'de bir sanal kaynak uygulamasından alınan vokselleştirilmiş kaynak dolgularının görüntüsü verilmiştir. Vokselleştirilmiş görüntü sadece tasarımcılara yardımcı olmak amacı ile geliştirilmiştir. Kullanıcılar yürüyen küpler algoritması uygulandıktan sonraki üçgenlerden oluşan yüzeyi görmektedir. Şekil 5.17.'de yürüyen küpler algoritması sonucunda oluşan görüntü verilmiştir.



Şekil 5.16. Vokselleştirilmiş kaynak dolguları



Şekil 5.17. Yürüyen küpler uygulanmış kaynak dolguları

5.3. Kaynak Sorgu Modülü

Kaynak sorgu modülü kaynak sırasında oluşturulacak dolgulara ait parametleri üretmek ile görevli olan yazılım birimidir. Modül içerisinde yapısı Şekil 5.3.'de gösterilmiş olan bir yapay sinir ağı bulunmaktadır. Bu ağ simülatör çalıştırılmadan önce eğitimini tamamlamış ve kullanıma hazır bir şekilde sisteme entegre edilmiştir. Böylece simülatör her çalıştığında, zaman kaybı olmaksızın direk kullanılabilmektedir. Yapay sinir ağının girişleri kullanıcının el hareketleri üzerinden hesaplanmaktadır.

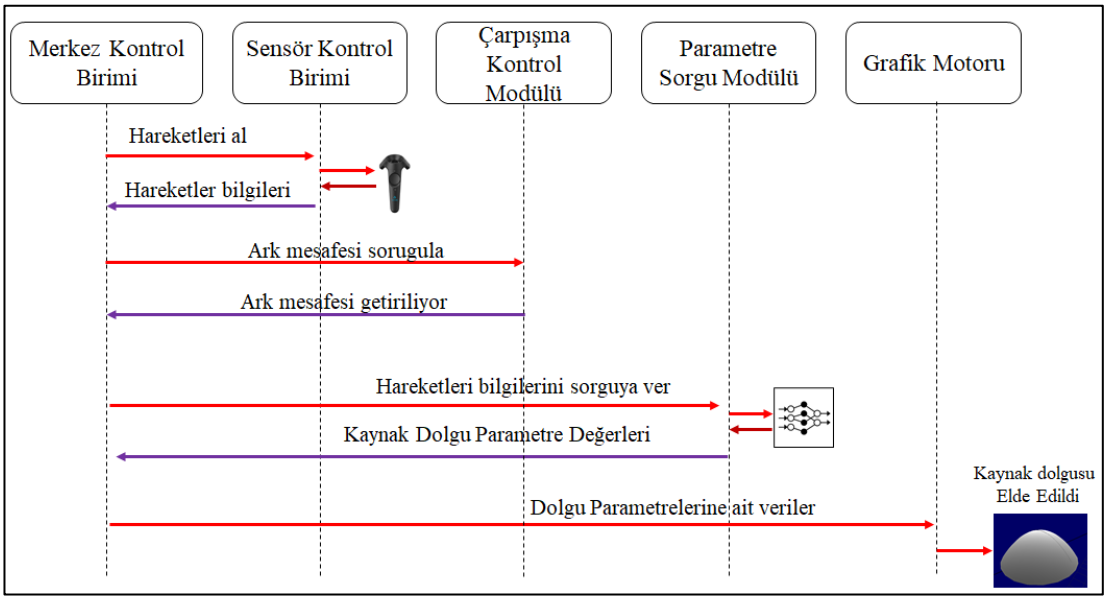
Sanal kaynak simülatöründe el hareketleri HTC Vive yöneticisi aracılığıyla sürekli olarak takip edilmektedir. Bu veriler simülatörün sensör kontrol birimi tarafından takip edilmektedir. Merkez kontrol birimi, sensör kontrol biriminden aldığı verileri Dolgu kaynak sorgu modülüne iletmektedir. Bu modülün görevi kullanıcının el hareketlerine göre elde edilecek kaynak dolgusunun parametrelerini üretmektir.

5.4. Kaynak Dolgu Şeklinin Çizilmesi

Kaynak işleminin başlayabilmesi için öncelikle torca ait elektrotun, birleştirilecek materyale ark oluşturacak kadar yaklaşması gerekmektedir. Merkezi kontrol birimi, simülatörün çalışma süresi boyunca sensör kontrol birimininden gelen bilgileri kontrol etmektedir. Bu bilgiler ile çarpışma kontrol modülünü kullanarak elektrotun materyale

olan uzaklığını elde eder. Eğer ark oluşacaksa parametre sorgu modülüne oluşacak kaynak dolgusu için el hareket bilgilerini gönderir. Ardından kaynak dolgusuna ait parametreleri grafik motoruna çizim yapması için devreder.

Merkezi kontrol birimi dolgu parametrelerini model üretmesi için grafik motoruna devreder. Grafik motoru kaynak parametrelerini kullanarak dolgu modeli nesnesini inşa eder ve 3B sahne içisine yerleştirir. Bir sonraki ekran görüntüsünde yeni kaynak dolgusu sahne üzerinde görünmektedir. Bu işlemlerin sıra diyagramı Şekil 5.18.'de gösterilmektedir.



Şekil 5.18. Simülatör modüllerinin kaynak dolgusunun elde edilmesindeki işlem sıra diyagramı

BÖLÜM 6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, MIG kaynak eğitimi için üretilen sanal kaynak simülatöründe kullanılacak olan yeni bir sanal kaynak dolgu modeli geliştirilmiştir. Sanal kaynak dolgu modeli, 3B ortamda kullanıcıların kaynak yaptığı materyale yerleştirilmekte ve dolgular arası bağlantı da kurulabilmektedir. Böylece kullanıclar yaptıları kaynak işleminin sonucunu gerçeğe yakın bir şekilde gözlemleyebilmektedir.

Kaynak dolgusunun üretiminde kullanılan sorgu modülü ve grafik motorunun herhangi bir ticari yazılım kütüphanesi kullanılmadan tez çalışmasında geliştirilmesi sanal kaynak simülatörünün maliyetinin düşürülmesine katkı yapmıştır. Tez çalışmasındaki sonuçlar, intel i5 6600K 3.3 Ghz. işlemciye, 32 gb ram ve nvidia gtx 970 ekran kartına sahip bir bilgisayar üzerinde alınmıştır.

Sanal kaynak simülatörü geliştirilirken temel amaç, kaynakçı adaylarına gerçeğe yakın bir kaynak ortamı sunarken eğitim maliyetlerini de azaltmak olmuştur. Bu sebeple, literatürde kaynak dolgusunun oluşumunu inceleyen araştırmalarda göz önünde bulundurulan erime ve katılaşma sürecindeki iş materyallerinin türü, elektrot türü gibi parametre hesaplamaları, karmaşık ve yüksek işlem gücü gerektirdiğinden sanal kaynak dolgusunun üretiminde göz ardı edilmişlerdir. Tez çalışmasında önerilen sanal kaynak dolgusu, simülatörün gerçek zamanlı olarak çalışabilmesi göz önünde bulundurularak geliştirilmiştir. Gerçek zamanlı çizim uygulamalarında saniyede ideal olarak 60 resim elde edilmesi gerektiği düşünülürse her bir resim için simülatörün 16,7 milisaniye sürede üretilmesi gerekmektedir. Bu sürenin bir kısmı da ekran kartındaki hesaplamalar için geçeceği düşünülürse kaynak dolgusunun üretilmesi çok daha düşük bir süre içerisinde gerçekleşmelidir. Tablo 6.1.'de bir sanal kaynak dolgusunun farklı voksel çözünürlüklerinde üretilme süreleri verilmiştir. Çözünürlük miktarının artması hesaplama zamanını da arttırmakta, fakat görüntü kalitesi de artmaktadır. Bu nedenle saniyede ortalama 69 resim üreten voksel çözünürlüğü olarak 1024x512x256 ideal

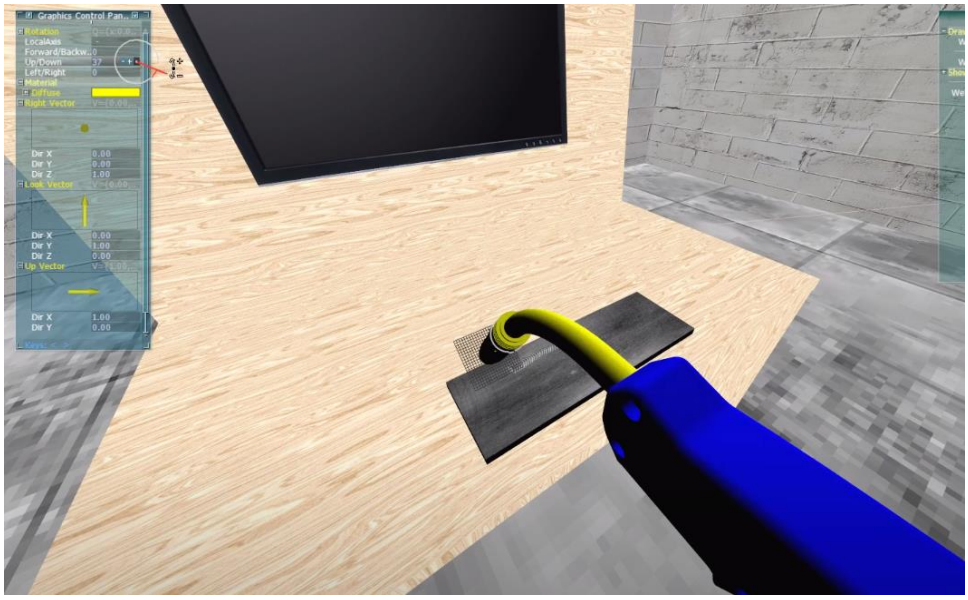
çözünürlük olarak belirlenmiştir. Paralel programlama sayesinde dört çekirdek kullanılarak kaynak dolgusu oluşturma işlemi çok daha hızlı hale getirilmiştir. İşlemcinin çekirdek sayısının iş parçacığı yani iplik sayısı ile eşit olması durumunda optimum verim sağladığı tabloda da görülmektedir. İplik sayısı çekirdek sayısının üstüne çıkartıldığında verim azalmaktadır.

Tablo 6.1. Bir kaynak dolgusunun çizim süreleri

Çözünürlük	Bir Saniyede Elde Edilen Ortalama Resim Sayısı	1 Kaynak Dolgusu Üretmek İçin Geçen Süre (ms)			
		İplik Sayısı			
		1	2	4	8
128x64x32	328	0,42	0,23	0,14	0,18
256x128x64	137	2,15	1,54	1,38	1,41
512x256x128	94	6,18	4,22	2,66	2,84
1024x512x256	69	9,23	6,84	3,73	3,82

Şekil 6.1.'de sanal kaynak simülatöründe bir kaynak uygulaması sonucunda elde edilen dolgunun görüntüsü verilmektedir. Şekilde 6.2.'de sanal kaynak simülatörüne ait 3B sanal ortam görünmektedir.

Şekil 6.1. Kaynak dolgusu örneği



Şekil 6.2. 3B sanal ortamın görüntüsü

İlerleyen dönemlerde bu çalışmada kullanılan vokselleme tekniği yerine dörtyüzlüler tekniği kullanarak sonuçların yeniden değerlendirilmesi de amaçlanmaktadır. Çizim için OpenGL kütüphanesi yerine daha düşük seviyede çalışan Vulkan kütüphanesinin kullanılması da hedeflenmektedir. Ayrıca grafik donanımlarını da parallel iş modülüne katmak da simülatörün çalışma hızını geliştirebilir.

KAYNAKLAR

[1] Öz, C., Serttaş, S., Ayar, K., Fehim, F., Effect Of Virtual Weldıng Sımulator On Tig Welding Training. Journal of Materials Education., 37(5-6), 197-217, 2015.

[2] Wu, C.S., Zhang, M.X., Li, K.H., Zhang, Y.M., Numerical analysis of double-electrode gas metal arc welding process. Comput Mater Sci., 39, 416-423, 2007.

[3] Kumar, A., Chauhan, V., Bist, A.S., Role of Artificial Neural Network in Welding Technology: A Survey, International Journal of Computer Applications,
67(1), 32-37, 2013.

[4] Shah, J., Pate, G., Makwana, J., A Review on Optimization and Prediction of MIG Welding Process Parameters Using ANN, International Journal of Engineering Development and Research, 5(1), 289-291, 2017.

[5] Dutta, P., Pratihar, D.K., Modelling of TIF Welding Process Using Conventional Regression Analysis and Neural Network-Based Approaches, Journal of Materials Processing Technology, 184, 56–68, 2007.

[6] Pal, K., Pal, S.K., Study of Weld Joint Strength Using Sensor Signals for Various Torch Angles in Pulsed MIG Welding, Journal of Manufacturing Science and Technology, 3, 55-65, 2010.

[7] Planckaert, J.P., Djermoune, E.H., Brie, D., Briand, F., Richard, F., Modeling of MIG/MAG Welding With Experimental Validation Using An Active Contour Algorithm Applied On High Speed Movie, Applied Mathematical Modelling, 34, 1004-1020, 2010.

[8] Ates H., Prediction of Gas Metal Arc Welding Parameters Based On Artificial Neural Networks, Materials and Design, 28, 2015–2023, 2007.

[9] Pal, S., Pal, S.K., Samantaray, A. K., Artificial Neural Network Modeling of Weld Joint Strength Prediction of A Pulsed Metal İnert Gaswelding Process Using Arc Signals, Journal of Materials Processing Technology, 202, 464-474, 2008.

[10] Sathiya, P, Panneerselvam K., Soundarajan, R., Optimal Design for Laser Beam Butt Welding Process Parameter Using Artificial Neural Networks And Genetic Algorithm For Super Austenitic Stainless Steel, Optics & Technology, 44, 1995-1914, 2012.

[11] Sathiya P., Panneerselvam K., Abdul Jaleel M.Y., Optimization of Laser Welding Process Parameters For Super Austenitic Stainless Steel Using Artificial Neural Networks And Genetic Algorithm" Materials and Design, 36, 490-498, 2012.

[12] Achebo, J.I., Optimization of GMAW Protocols and Parameters for Improving Weld Strength Quality Applying the Taguchi Method, Proceedings of the World Congress on Engineering, 2011.

[13] Nagesh, D.S., Datta, G.L., Prediction of Weld Bead Geometry and Penetration in Shielded Metal-Arc Welding Using Artificial Neural Networks, Journal of Materials Processing Technology, 123, 303-312, 2002.

[14] Correia, D.S., GMAW Welding Optimization Using Genetic Algorithms, Journal of the Brazilian Society of Mechanical Sciences and Engineering, 1, 28-33, 2004.

[15] Khuder, A.W.H., Ebraheam, E.J., Study the Factors Effecting on Welding Joint of Dissimilar Metals, Al-Khwarizmi Engineering Journal, 7(1), 76-81, 2011.

[16] Kumar, A., Jadoun, R.S., Bist, A.S., Optimization of MIG welding parameters using Artificial Neural Network (ANN) and Genetic Algorithm (GA), International Journal Of Engineering Sciences & Research Technology, 3(7), 614-620, 2014.

[17] Hooda, A., Dhingra, A., Sharma, S., Optimization of MIG Welding Process Parameter To Predict Maximum Yield Strength in AISI 1040, International Journal of Mechanical Engineering And Robotics Research, 1(3), 203-213, 2012.

[18] Balasubramanian V., Ravisankar, V., Reddy, G.M., Effect of Pulsed Current Welding On Mechanical Properties of High Strength Aluminum Alloy, International Journal Advance Manufacturing Technology, 36, 254-262, 2008.

[19] Patel, C.N., Chaudhary, S., Parametric Optimization of Weld Strength of Metal Inert Gas Welding and Tungsten Inert Gas Welding by using Analysis of Variance and Grey Relational Analysis, International Journal of Research in Modern Engineering and Emerging Technology, 1(3), 48-56, 2013.

[20] Ghazvinloo H.R., Honarbakhsh-Raouf A., Shadfar N., Effect of Arc Voltage, Welding Current and Welding Speed on Fatigue Life, Impact Energy and Bead penetration of AA6061 Joints Produced by Robotic MIG Welding. Indian Journal of Science and Technology, 3(2), 2010.

[21] Aktepe, A., Öncel, Ç., Ersöz, S., An Artificial Neural Network Model on Welding Process Control of 155 mm. Artillery Ammunition, International Advanced Technologies Symposium, 153-158, 2011.

[22] Ibrahim, I.A., Mohamat, A.A., Amir, A., Ghalib, A., The Effect of Gas Metal Arc Welding (GMAW) processes on different welding parameters, International Symposium on Robotics and Intelligent Sensors, 1502-1506, 2012.

[23] Shoeb, M., Parvez, M., Kumari, P., Effect of MIG Weldıng Input Process Parameters on Weld Bead Geometry on HSLA Steel, International Journal of Engineering Science and Technology, 5(1), 200-212, 2013.

[24] Mishra, B., Panda, R.R., ve Mohanta, D.K., Metal Inert Gas (Mig) Welding Parameters Optimization, International Journal of Multidisciplinary and Current Research, 2, 637-639, 2014.

[25] Chaki, S. ve S. Ghosa, S., A GA-ANN Hybrid Model for Prediction and Optimization of CO2 Laser-MIG Hybrid Welding Process, International Journal of Automotive and Mechanical Engineering, 11, 2458-2470, 2015.

[26] Prakash, A., Bag, R.K., Ohdar, P., Raju, S.S., Parametric Optimization of Metal Inert Gas Welding by Using Taguchi Approach, International Journal of Research in Engineering and Technology, 5(2), 176-182, 2016.

[27] Schneider, C.F., Lisboa, C.P., Silva, R.D.A., ve Lermen, R.T., Optimizing the Parameters of TIG-MIG/MAG Hybrid Welding on the Geometry of Bead Welding Using the Taguchi Method, Journal of Manufacturing and Materials Processing, 1(2), 14, 2017.

[28] Sudarshan, R. Devaiah, M., Effect of Process Parameters in MIG Welding on Mild Steel IS 2062, International Journal of Applied Engineering Research, 13(4), 2046-2054, 2018.

[29] Kanti, K.M., Rao, P.S., Prediction of bead geometry in pulsed GMA welding using back propagation neural network. Journal of Materials Processing Technology., 200, 300-305, 2008.

[30] Sudhakaran, R., Murugan, V., Sivasakthivel, P.S., Balaji, M., Prediction and Optimization of Depth of Penetration For Stainless Steel Gas Tungsten Arc Welded Plates Using Artificial Neural Networks and Simulated Annealing Algorithm. Neural Computing & Applications., 22, 637-649, 2013.

[31] Sreeraj, P., Kanna, T., Modelling and Prediction of Stainless Steel Clad Bead Geometry Deposited by GMAW Using Regression and Artificial Neural Network Models. Advances in Mechanical Engineering, 2012.

[32] Iqbal, A., Khan, S.M., Sahir, M.H., ANN Assisted Prediction of Weld Bead Geometry in Gas Tungsten Arc Welding of HSLA Steels. Proceedings of the World Congress on Engineering, 2011.

[33] Möller At., Haines, E., Real-Time Rendering. AK Peters Ltd., 2002.,

[34] An Object-Oriented Graphics Engine.International Conference on Computer Science and Software Engineering, CSSE, Wuhan,12-14,2008.

[35] Shreiner, D., Sellers, G. OpenGL Programming Guide: The Official Guide to Learning OpenGL. Version 4.3, Pearson Education, Inc, 1-2, 2013.

[36] Wright, R., Lipchak, B. Haemel,N. OpenGL Super Bible Forth Edition, Pearson Education, Inc, 94-135, 2007.

[37] Wolf, D. OpenGL 4.0 Shading Language Cookbook, Packt Publishing Ltd, Inc, 11-85, 2011.

[38] Armeni,I., He,Zhi. 3D Scene Graph: A Structure for Unified Semantics,3D Space, and Camera., IEEE/CVF International Conference on Computer Vision,2019.

[39] Chambers, T.L., Aglawe, A., Reiners, D., White, S., Borst, C.W., Prachyabrued, M., Bajpayee, A., Real-time simulation for a virtual reality-based MIG welding training system. Virtual Reality., 16(1), 45-55, 2012.

[40] Wu, C.S., Zhang, M.X., Li, K.H., Zhang, Y.M., Numerical analysis of double-electrode gas metal arc welding process. Comput Mater Sci., 39, 416-423, 2007.

[41] Mavrikios, D., Karabatsou, V., Fragos, D., Chryssolouris, G., A Prototype Virtual Reality-Based Demonstrator for Immersive and Interactive Simulation of Welding Process. International Journal of Computer Integrated Manufacturing., 19, 294-300, 2006.

[42] Öz, C., Fındık, F., İyibilgin, O., Soy, U., Sanal Kaynak Simülatörü Tasarımı ve İmalatı. Tubitak-109M087, 2011.

[43] Öz, C., Serttaş, S., Ayar, K., Fehim, F., Effect Of Virtual Weldıng Sımulator On Tig Welding Training. Journal of Materials Education., 37(5-6), 197-217, 2015.

[44] Son, J.S., Kim, I.S., Kim, H.H., Kim, I.J., Kang, B.Y. ve Kim, H.J., A Study on The Prediction of Bead Geometry in The Robotic Welding System. Journal of Mechanical Science and Technology., 21, 1726-1731, 2007.

[45] Sreeraj, P., Kanna, T., Modelling and Prediction of Stainless Steel Clad Bead Geometry Deposited by GMAW Using Regression and Artificial Neural Network Models. Advances in Mechanical Engineering, 2012.

[46] Karadeniz, E., Ozsarac, U., Yildiz, C., The Effect of Process Parameters on Penetration in Gas Metal Arc Welding Processes, Materials and Design., 28, 649-656, 2007.

[47] Kim, I.S., Son, K.J., Yang, Y.S., Yaragada, P.K.D.V., Sensitivity analysis for process parameters in GMA welding processes using a factorial design method. International Journal of Machine Tools & Manufacture., 43, 763–769, 2003.

[48] Das, B., Debbarma, B., Rai, R.N., Saha, S.C., Influence of Process Parameters on Depth of Penetration of Welded Joint In MIG Welding Process. International Journal of Research in Engineering and Technology., 02(10), 220-224, 2013.

[49] Ghazvinloo, H.R., Honarbakhsh-Raouf A., ve Shadfar, N., Effect of arc voltage, welding current and welding speed on fatigue life, impact energy and bead penetration of AA6061 joints produced by robotic MIG welding. Indian Journal of Science and Technology., 3(2), 2010.

[50] Tewari, S.P., Gupta, A., Prakash, J., Effect of Welding Parameters on The Weldability of Material. International Journal of Engineering Science and Technology., 2(4), 512-516, 2010.

[51] Shoeb, M., Parvez, M., Kumari, P., Effect of MIG Welding Input Process Parameters On Weld Bead Geometry On HSLA Steel. International Journal of Engineering Science and Technology., 5(1), 200-212, 2013.

[52] Çavuşlu, M.A., Becerikli, Y., Karakuzu, C., Levenberg-Marquardt Algoritması ile YSA Eğitiminin Donanımsal Gerçeklenmesi. TBV Bilgisayar Bilimleri ve Mühendisliği Dergisi., 5, 31-38, 2012.

[53] Wilamowski, B.M., Chen, Y., Efficient algorithm for training neural Networks with one hidden layer. Proc. of the International Joint Conference on Neural Networks., 3, 1725-1728, 1999.

[54] Dohnal, J., Using of Levenberg Marquardt method in identification by neural networks. Student EEICT, 361-365, 2004.

[55] Arif, J., Chaudhuri, N.R., Ray, S., Chaudhuri, B., Online Levenberg-Marquardt Algorithm For Neural Network Based Estimation And Control Of Power Systems. International Joint Conference on Neural Networks, IEEE, 199-206, 2009.

[56] Kaufman, A., Shimony, E., 3D Scan Conversion Algorithms for Voxel-Based GRaphics. Proc. ACM 1986 Workshop on Interactive 3D Graphics, pp.45-76, 1986.

[57] Kaufman, A., Cohen, D., Yagel, R., Volume Graphics. IEEE Computer; 26:7:51-64, 1993.

[58] Foley, J.D., Van Dam, A., Feiner, S.K., Hughes, J.F., Computer Graphics: Principles and Practice. 2nd Edition, Addison-Wesley, pp.92-99, 1990.

[59] Hearn, D., Baker, M.P., Computer GRaphics, C Version. 2nd Edition, Donald Hearn, 1996.

[60] Bresenham, J.E., Algorithm for Computer Control of a Digital Plotter. IBM Systems Journal; 4:1:25-30, 1965.

[61] Nuber, C., Bruckshen, R.W., Hamann, B., Joy, K.I., Interactive Visualization of Very Large Medical Datasets Using Point-Based Rendering. In Proc. SPIE 5029, Medical Imaging Visualization, Image-Guided Procedures, and Display, 27, 2003.

[62] Agus, M., Giachetti, A., Gobbetti, E., Zenetti, G., Adaptive techniques for real–time haptic and visual simulation of bone dissection. In IEEE Virtual Reality Conference, IEEE Computer Society Press, 102-109, 2003.

[63] Lai, Y.M., Quinb, H.Y., Lee, K.K.H., Chan, K.M., Regional differences in trabecular BMD and micro-architecture of weight-bearing bone under habitual gait loading—A pQCT and microCT study in human cadavers. Bone., 37, 274-282, 2005.

[64] Eriksson, M.G., Flemmer, H., Wikander, J., A Haptic and Virtual Reality Skull Bone Surgery Simulator. World Haptics Conference, Italy, 2005.

[65] Eriksson, M.G., Dixon, M., Wikander, J., A Haptic VR Milling Surgery Simulator Using High-Resolution CT Data. The 14th MMVR Conference in Los Angeles, USA, 2006.

[66] Chevalier, Y., Pahr, D., Allmer, H., Charlebois, M., Zysset, P., Validation of a voxel-based FE Method for prediction of the uniaxial apparent modulus of human trabecular bone using macroscopic mechanical tests and nanoindentation. J. Biomech., 40, 3333-3340, 2007.

[67] Altintas, G., Node-id based non-recursive flood fill algorithm for non-uniform discrete solid domains. 2nd World Conference On Information Technology, Antalya, Turkey, 2011.

[68] Altintas, G., Erdem, R.T., Effect of micro-ct slice intensity on natural vibration behavior of cancellous bone models based on reverse engineering techniques. Procedia Technology., 1, 318-322, 2012.

[69] Niu, Q., Chi, X., Leu, M.C., Ochoa, J., Image Processing, Geometric Modeling and Data Management for Development of a Virtual Bone Surgery System. Computer Aided Surgery., 13(1), 30-40, 2008.

[70] Duriez, C., Syllebranque, C., Siz Degree-of Freedom Haptic Rendering for Dental Implanatory Simulation, 139-149, 2010.

[71] Eriksson, M.G., Wikander, J., A Face Validated Six Degrees-of-Freedom Haptic Bone Milling Algorithm, IEEE Transactions on Haptics, 2012.

[72] Ferley, E., Cani, M.P., Gascuel, J.D., Practical Volumetric Sculpting. The Visual Computer., 168, 469-480, 2000.

[73] Ferley, E., Cani, M.P., Gascuel, J.D., Resolution Adaptive Volume Sculpting. Journal of Graphical Models., 63(6), 459-478, 2001.

[74] Raffin, R., Gesquiere, G., Remy, E., Thon, S., VirSculpt: A Virtual Sculpting Environment. International Conference Graphicon, 184-187, 2004.

[75] Ho, C.C., Tu, C.H., Ouhyoung, M., Detail Sculpting using Cubical Marching Squares. ICAT'0, 10-15, 2005.

[76] Heurtebise, X., Thon, S., Discrete Tools for Virtual Sculpture. GRAPP'06, 415-422, 2006.

[77] Williams, J., O'neill, G.T., Lee, W.S., Interactive 3D Haptic Carving using Combined Voxels and Mesh. IEEE International Workshop on Haptic Audio Visual Environments and their Applications, 108-113, 2008.

[78] O'neill, G.T., Lee, W.S., Williams, J., Haptic-Based 3D Carving Simulator, Advances in Haptics, 299-314, 2010.

[79] Altıntaş, F., Gültekin, B., Sınır, Değer Seçiminin Voksel Tabanlı Sonlu Eleman Modellerinin Doğal Titreşim Davranışları Üzerindeki Etkilerinin İncelenmesi. C.B.Ü., Fen Bilimleri Dergisi., 9(1), 1-9, 2013.

[80] Prior, A., Real-Time Collaborative Volumetric Virtual Sculpting with Haptic Force-Feedback, Doktora Tezi, 2009.

[81] Echegaray, G., Borro, D., A methodology for optimal voxel size computation in collision detection algorithms for virtual reality. Virtual Reality., 163, 205-213, 2012.

[82] Ayar, K., Çit, G., Öz, C., Serttaş, S., Voxelization with OpenCL for Virtual Sculpting. International Symposium On Innovative Technologies. Engineering And Science, 2014.

[83] Nourian, P., Gonçalves, R., Zlatanovac, S., Ohori, K.A., Vo, A.V., Voxelization Algorithms for Geospatial Applications: Computational methods for voxelating spatial datasets of 3D city models containing 3D surface, curve and point data models, MethodsX., 3, 69-86, 2016.

[84] Çit, G., Ayar, K., Öz, C., A Real-Time Virtual Sculpting Application By Using An Optimized Hash-Based Octree. Turkish Journal of Electrical Engineering & Computer Sciences., 24, 2274-2289, 2016.

[85] Barentzen, A., Octree-based Volume Sculpting. IEEE Visualization '98, Late Breaking Hot Topics Preoceedings, IEEE Computer Society Press, 9-12, 1998.

[86] Özkurt, A., Ameliyatlar İçin İnteraktif Tıbbi Görüntüleme, DEÜ, Fen Bilimleri Enstitüsü, Doktora Tezi, 2001.

[87] Guha S., An Optimal Mesh Computer Algorithm for Constrained Delaunay Triangulation, 8. International Parallel Processing Symp, 102-109, Meksika, 1994.

[88] Han, J., MRI and CT Image Based on 3D Reconstruction and Medical Rapid Prototyping. Puerto Rico Üniversitesi, Makine Mühendisliği, Yüksek Lisans Tezi, 2005.

[89] Chernikov, A.N., Xu, J., A computer-assisted proof of correctness of a marching cubes algorithm. Proceedings of the 22nd International Meshing Roundtable, 505-523, 2014.

[90] Parmar, B.N., Bhatt, T., Volume Visualization using Marching Cubes Algorithms: Survey & Analysis. Internatıonal Journal Of Innovatıve Research In Technology., 2, 21-25, 2016.

[91] Roy, S., Augustine, P., Comparative Study of Marching Cubes Algorithms for the Conversion of 2D image to 3D International Journal of Computational Intelligence Research., 13, 327-337, 2017.

ÖZGEÇMİŞ

Kayhan AYAR, 29 Mart 1982 tarihinde Muş'ta doğdu. İlk ve orta eğitimini Sakarya'da, lise eğitimini ise İstanbul'da tamamladı. 2007 yılında Sakarya Üniversitesi Bilgisayar Mühendisli bölümünden mezun oldu. 2008 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünde yüksek lisans eğitimine başladı ve 2009 yılında mezun oldu.