



**T.C.
Gebze Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü**

**BİL396
Bilgisayar Mühendisliği Proje**

CanTouchThis

Final Raporu

Grup 5 Üyeleri

Yavuz Selim ERDÖL
Barış ŞAHİN
Yusuf AKDAŞ
Ferhat SENCER
Ubeydullah ÇOBAN
Mustafa PASLI
Emire KORKMAZ
Ahmet Yasir NACAK

Danışman: Prof. Dr. Erkan ZERGEROĞLU

Mayıs – 2019
Gebze – Kocaeli

İçindekiler

1. CanTouchThis Tanımı

1.1 Kontrol

1.2 Tarama

1.3 İşleme ve Dönüştürme

1.4 Görüntüleme

2. Sistem Gereksinimleri

2.1 Yazılımsal Gereksinimler

2.2 Donanımsal Gereksinimler

3. Projenin Modülleri

3.1 Donanım Modülü

3.2 Kontrol Arayüzü Modülü

3.3 Modüller Arası Haberleşme (İletişim) Modülü

3.4 Üç Boyutlu Obje Görüntüleme Modülü

3.5 Veri İşleme ve Dönüştürme Modülü

4. Örnek Sonuçlar

5. Maliyet

6. Grup Üyeleri ve Modüllerin Paylaşımı

1. CanTouchThis Tanımı

Bu projenin amacı, gerçek hayatta elimizde olan bir objeyi, dijital ortama fiziksel temas aracılığıyla aktarmak ve üç boyutlu olarak görüntüsünü elde etmektir.

Görüntü elde etme işlemi motorlu yapıyla sağlanmaktadır. 3 boyutlu cismin taranması için 3 eksenli bir yapı bulunmakta cisme temas eden donanımda ise hassasiyeti oldukça yüksek olan bir yöntem kullanılmaktadır.

1.1 Kontrol

Donanımın son kullanıcı tarafından kontrol edilmesi için minimal ve fonksiyonel bir arayüz bulunmakta. Arayüz motorlu yapıda eksenlerdeki motorların eksen bazlı bağımsız yönetilmesine imkan vermektedir.

1.2 Tarama

Tarama alanına konulan cisim başla komutu ile yapının eksenlerde konum almasıyla başlamaktadır. Cisim belli bir rota izlenerek taranmaktadır.

1.3 İşleme ve Dönüştürme

Tarama işlemi sırasında yükseklik ve tarama sırasında takip ettiği rota bilgisi ele alınarak cismin büyük oranda 3 boyutlu hali oluşturulmakta ve tarama işlemi dokunarak gerçekleştiğinden elde edilen sonuç Interpolation algoritması ile yumuşatılarak .obj formatına dönüştürülür.

1.4 Görüntüleme

Oluşan obje dosyasının görüntülenmesi için .obj görüntüleyici geliştirilmiştir. Tarama devam ederken obje gerçek zamanlı olarak güncellenmektedir.

2. Sistem Gereksinimleri

CanTouchThis'in geliştirilmesi ve çalıştırılması için iki tip gereksinim bulunmaktadır.

2.1 Yazılımsal Gereksinimler

Modüller arası iletişim için Boost::Asio kütüphanesinin bulunması gerekmektedir. .obj dosyalarını görüntülemek için geliştirilmiş olan Görüntüleme modülü için OpenGL ve GLUT kütüphanelerinin kurulu olması gereklidir. Arayüz modülü için ise Qt 5.9 veya üzeri gereklidir.

2.2 Donanımsal Gereksinimler

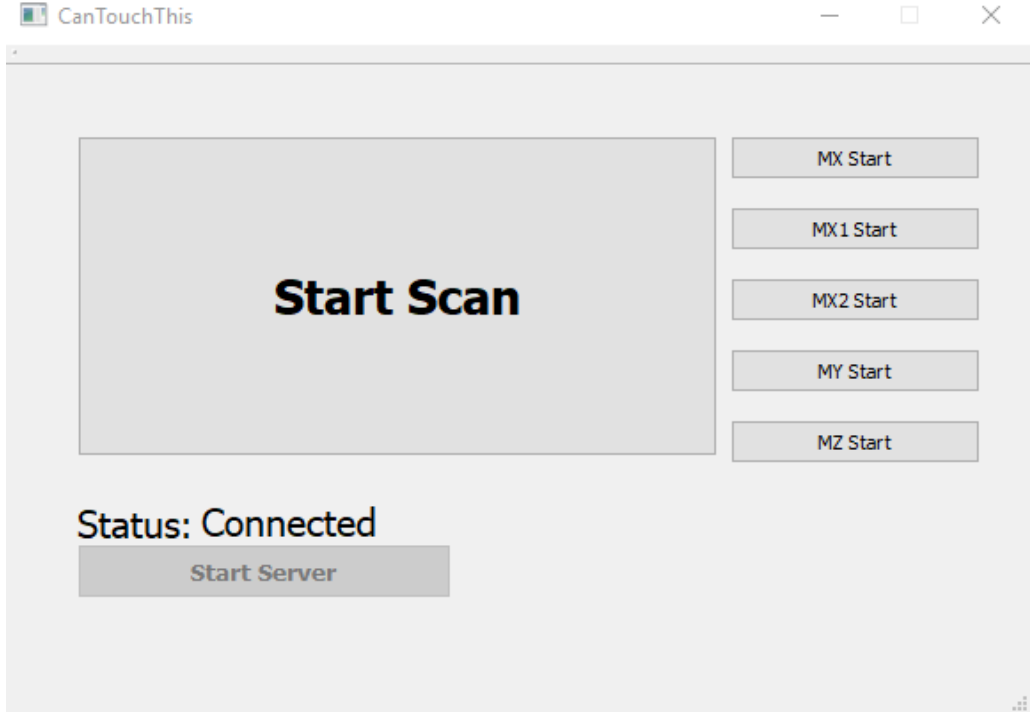
- Tarama alanını sınırlayan bir iskelet
- X ekseninde hareket edebilmesi için 2 adet Step Motor
- Y ekseninde hareket edebilmesi için 1 adet Step Motor
- Hareketin sağlanması için mil şaft ve dişli kayış ve şaft üzerinde kayabilecek bir parça
- Motorların sürülebilmesi için motor sürücü kartları
- Motorların yönetilebilmesi için Raspberry Pi mikroişlemcisi
- Başlangıç noktasına dönebilmesi için 2 adet mikro düğme
- Yükseklik için teması algılayabilecek ince uçlu hassas probe

3. Proje Modülleri

CanTouchThis'i bütün haline getiren beş kısım bulunmaktadır.

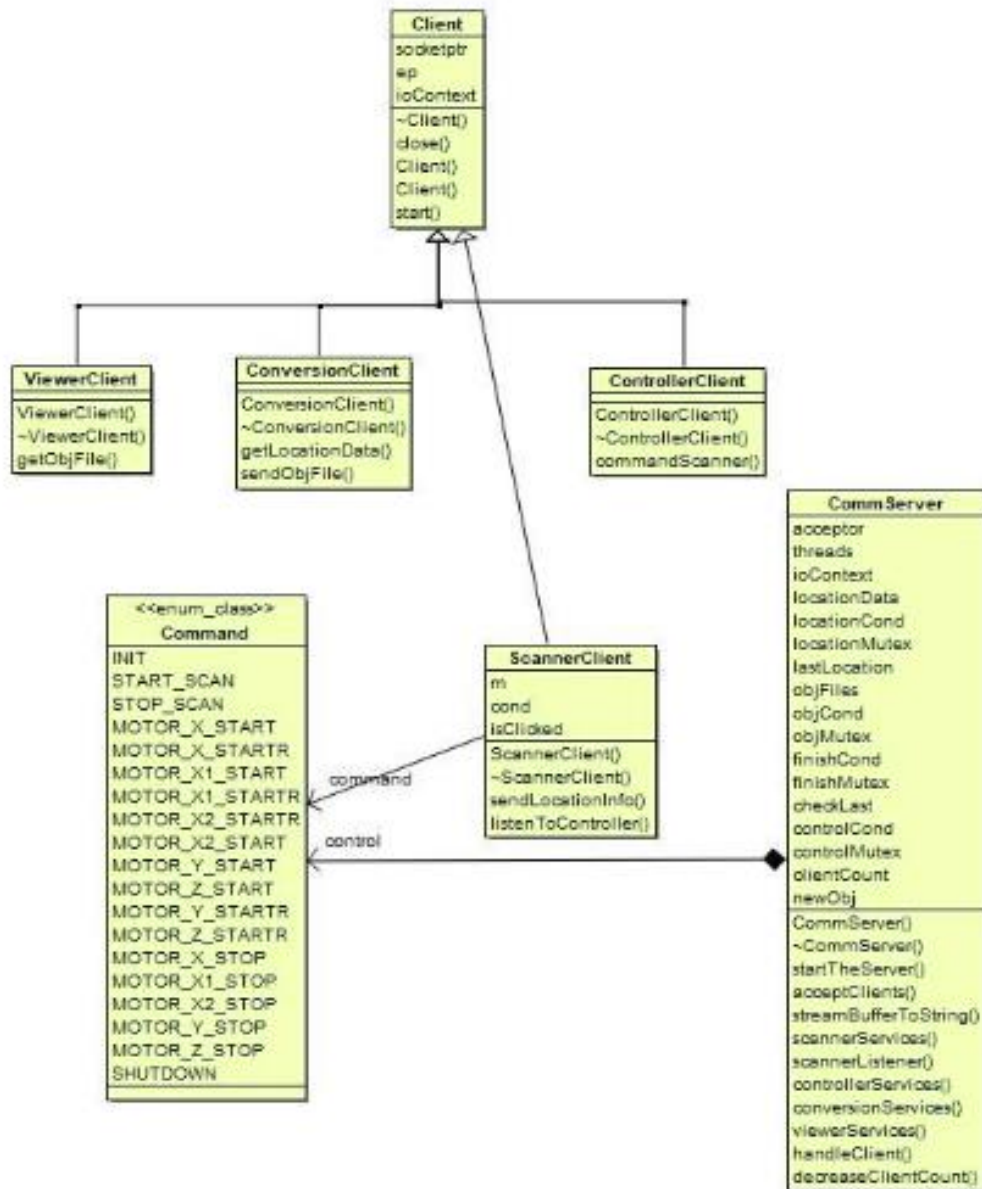
3.1 Kontrol Arayüzü Modülü

Kontrol modülü C++ için en iyi seçenek olan QT kullanıldı, arayüzde komponentlerin dinamikliği için QT'nin sinyal slot mekanizması kullanıldı. Böylece arayüz oldukça esnek ve kolay müdahale edilebilir oldu. İletişim modülünün entegrasyonu kolaylaştı. Kullanıcının etkileşime geçeceği bu modül olabildiğince sade ve işini yapan bir arayüz haline getirildi. Kontrol modülü senaryosu şöyledir; kullanıcı, üç boyutlu görüntüsünü almak istediği objeyi probe'un tarayacağı alana yerleştirdikten sonra "START SCAN" butonuna tıklayarak cihazın verilen objeyi taramaya başlamasını sağlar. Tarama devam ederken herhangi bir anda kullanıcı taramayı durdurup yeniden başlatabilir. Ayrıca, arayüz modülünde cihazımızın motorlarının çalışma durumlarını test etmek için eklenen "DEBUG MODE" bulunmaktadır. Motorları bir arada çalıştırmak yerine, motorların çalışmasının daha detaylı olarak incelenmesine olanak sağlayan bu butonlarla her bir motor ayrı ayrı çalıştırılabilmektedir.



3.2 Modüller Arası Haberleşme Modülü

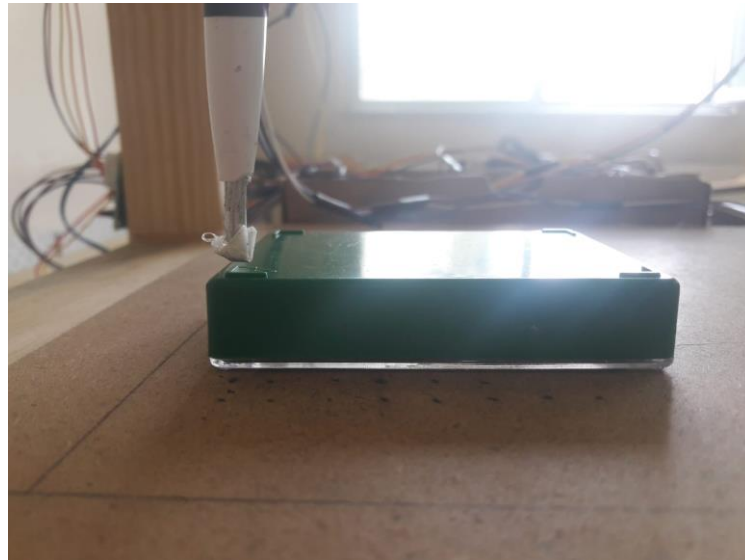
Modüller arası haberleşme için soket yapısı kullanıldı. Soket yapısının implementasyonunda C++ Boost kütüphanelerinin alt kütüphanesi olan Asio kullanıldı. Server-Client modeli kullanılarak diğer modüllerin veri akışları sağlandı. İletişim için JSON veri tipi kullanıldı. Arayüz - tarama modülü ve tarama - görüntüleme modülleri arasındaki iletişim hızlı ve güvenli olarak sağlandı. Modül, deployment işleminin hatasız yapılabilmesi için güncel kütüphanelere sadık kalındı bunun yanında donanım modülünde de çalışması için early-stable versiyonlara uyumlu olarak geliştirildi.



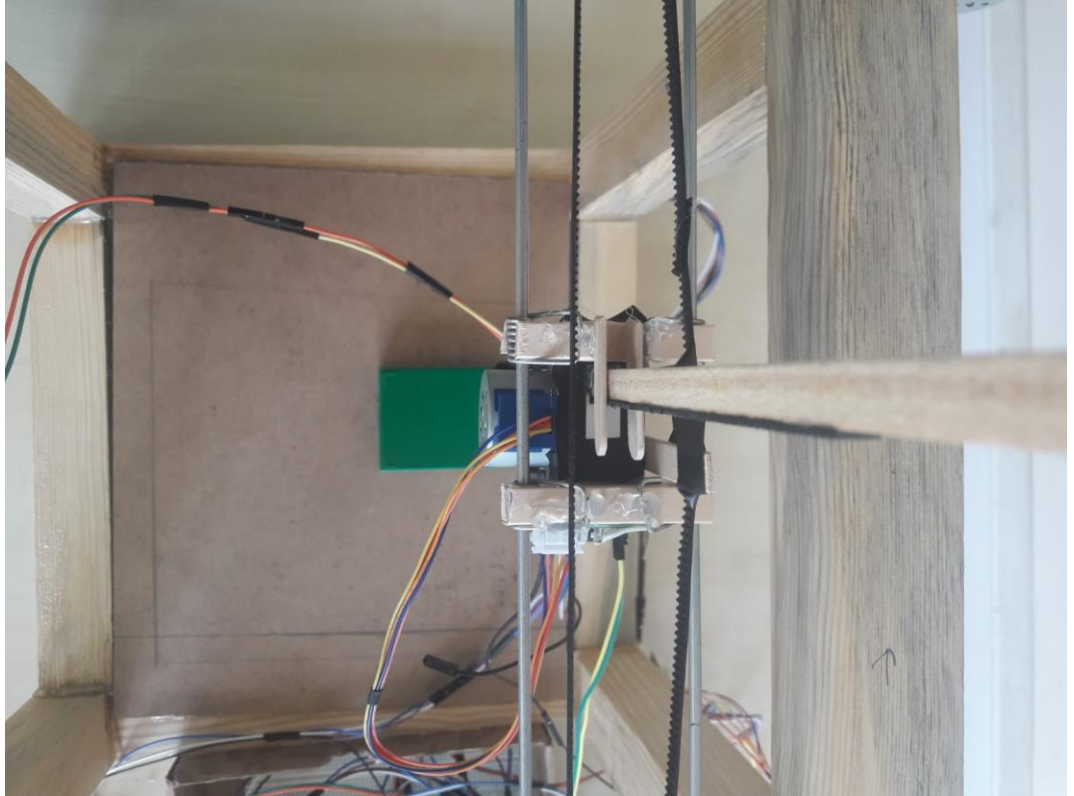
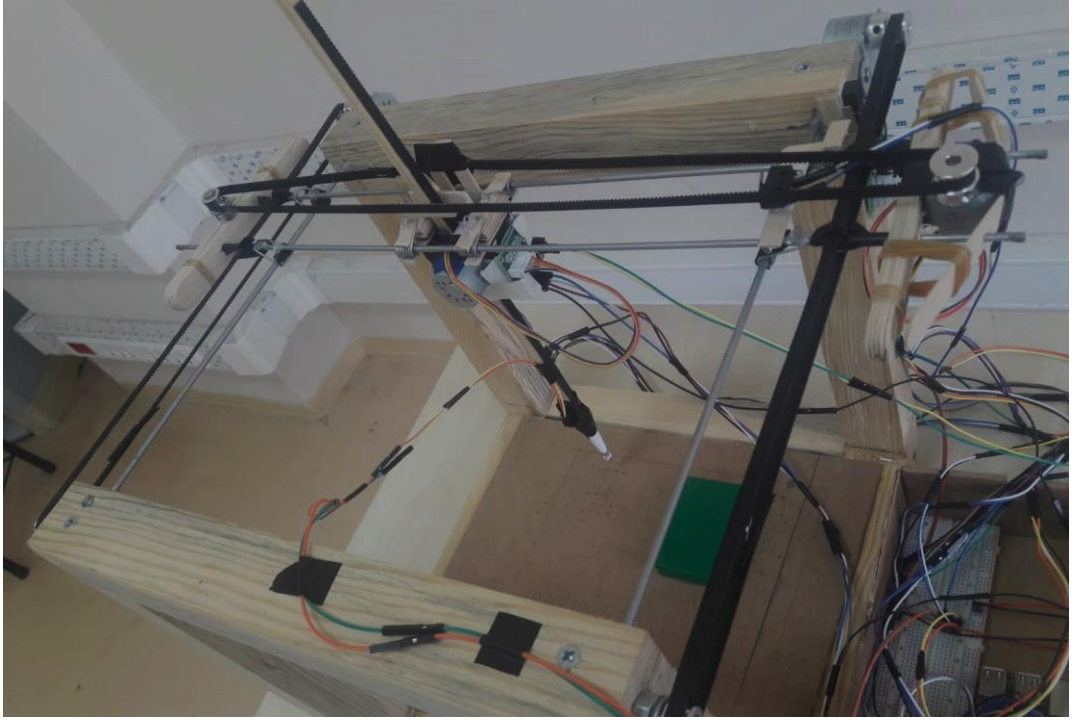
3.3 Donanım Modülü

Benzer cihazlar incelendi ve X, Y ve Z düzlemi için birer motorun kullanıldığı ve dokunma işlemi için bir düğmenin kullanıldığı tekerli tasarım yapıldı. Motorları sürmek ve yükseklik verisinin wifi üzerinden aktarılabilmesi için Raspberry Pi 3B modeli uygun görüldü. CanTouchThis tasarımı için tahtadan bir iskelet yapıldı. Daha sonra yeni tasarım fikirleri ile tekerler eski tasarımdan çıkarıldı. İskelet bu yeni tasarım için güncellendi. Motorlar düzeneğe yerleştirilmeden sürülmeye başlandı. Motorların kontrolü ve Raspberry Pi'a sinyal gönderme işlemleri için wiringPi kütüphanesi kullanıldı. Tarama işlemi için bütün motorların senkronize çalıştığı bir algoritma tasarlandı. Mikro düğme elle tetikleyerek sahte test verileri alındı. İskelete şaftlar ve y ekseninde hareket için gerekli olan birinci kızak monte edildi. Kızakların hareketi motorlar olmadan test edildi. Ara demodan sonra X ve Y eksenindeki motorlar monte edildi. Hareket testi yapıldıktan sonra X ekseninde bir motorun yeterli olmadığına karar verildi. X eksenine yardımcı olması için ikinci bir motor monte edildi ve düzeneğe bağlandı. Z ekseninde hareket için prob tasarlandı ve monte edildi. Yapılan testlerde mikro düğmenin yeterince hassas olmadığı sonuca varıldı. Cisme dokunulduğunda algılayabilmesi için ve yüksekliğin hesaplanabilmesi için dokunulduğunda sinyal gönderecek mikro düğmeden daha hassas bir touch probe geliştirildi. Dokunulduğunda gelen sinyallerin alınabilmesi için wiringPiISR metodu kullanıldı. Son tasarımda tarama için 2 motor X eksen, 1 motor Y eksen ve 1 motor Z eksenini kontrol için kullanıldı. Pozisyonun sıfırlanmasında sorun yaşandığı için X ve Y ekseninin 0 noktalarına 2 adet mikro düğme konuldu. Düğmelere dokununca probun 0 noktasında olduğu anlaşıldı. Testlere devam edildi ve yükseklik değerlerinde hata tespit edildi. Sebebinin kullandığımız step motorların ileri gidiş gücü ile geri dönüş gücü aynı olmadığından farklı değerler elde edildi. Hata yazılımsal olarak probun her Y eksenin başında yüksekliği boş bir yüzeye dokunarak hesaplamasıyla giderildi. Birbirine çok yakın yükseklik değerleri eşitlenerek pürüz giderildi.

Geliştirdiğimiz touch probe:



Donanımın Başka Açılardan Görselleri:

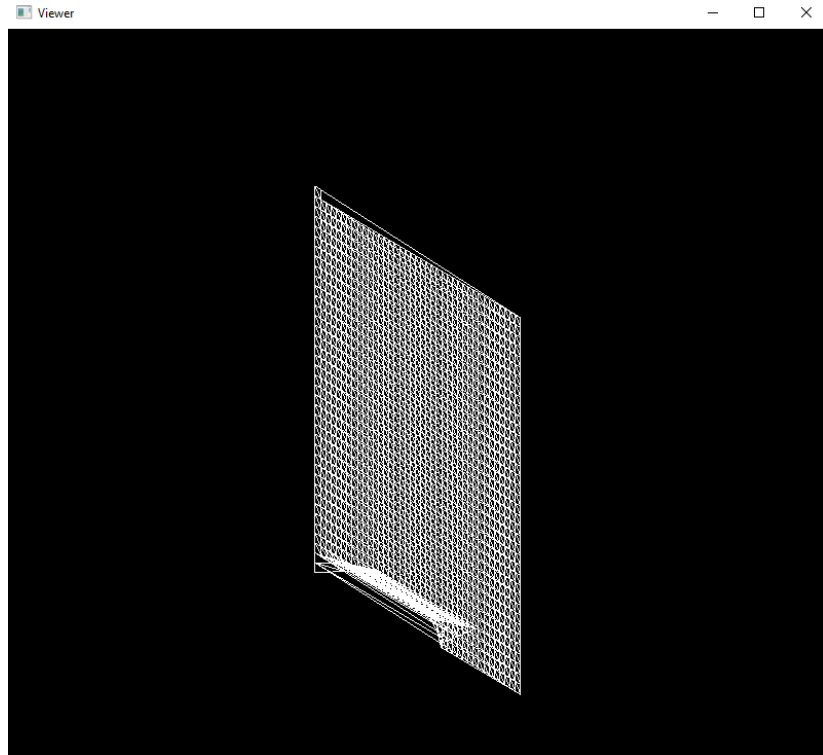


3.4 Üç Boyutlu Obje Görüntüleme Modülü

Bir üç boyutlu Wavefront Object (.obj) dosyasının bir pencerede görüntülenebilmesi için OpenGL Graphics API'dan yararlanıldı. Bu API sayesinde program, bilgisayarın GPU'su ile iletişime geçerek saniyede en az 60 kere kendisine verilen (.obj) dosyasını ekranda gösterdi. Bu modül ekranda görüntüleme yapmadan önce kendisine Veri İşleme ve Dönüştürme Modülü'nden gelen yükseklik haritasının (.obj)'e dönüştürülmüş halini alıp satır satır okuyarak kenar (vertex) ve yüzleri (face) bularak ekrana çizdirmesi gereken objenin bilgilerini OpenGL'in anlayabileceği bir şekilde tutuyor. Daha sonra bu yüklenen obje bir sonsuz döngü içerisinde saniyede en az 60 kare gösterecek şekilde render ediliyor. Kullanıcı isterse objeyi sağa-sola ve yukarı-aşağı döndürme seçeneklerine sahiptir.

Bu modülün bir pencere oluşturma ve sürekli kullanıcıdan input alabilmesi GLUT(OpenGL Utility Toolkit) tarafından sağlanmaktadır. Bu program içerisinde ayrıca kullanıcı istediği zaman taramanın en güncel sonucundaki objeyi görüntüleme seçeneğine sahiptir. Eğer kullanıcı en güncel durumu görmek istemiyor ve önceki durumda kalmak istiyorsa programı yenilemeyerek bunu başarabilir.

Obje Görüntüleme Modülü, başka hiçbir modüle bilgi aktarmaz ve ihtiyaç duyduğu tüm bilgiyi Veri İşleme ve Dönüştürme Modülünden sağlar. Bu modüle gelen bilgi yükseklik haritasından dönüştürülen (.obj) dosyasıdır.



3.5 Veri İşleme ve Dönüştürme Modülü

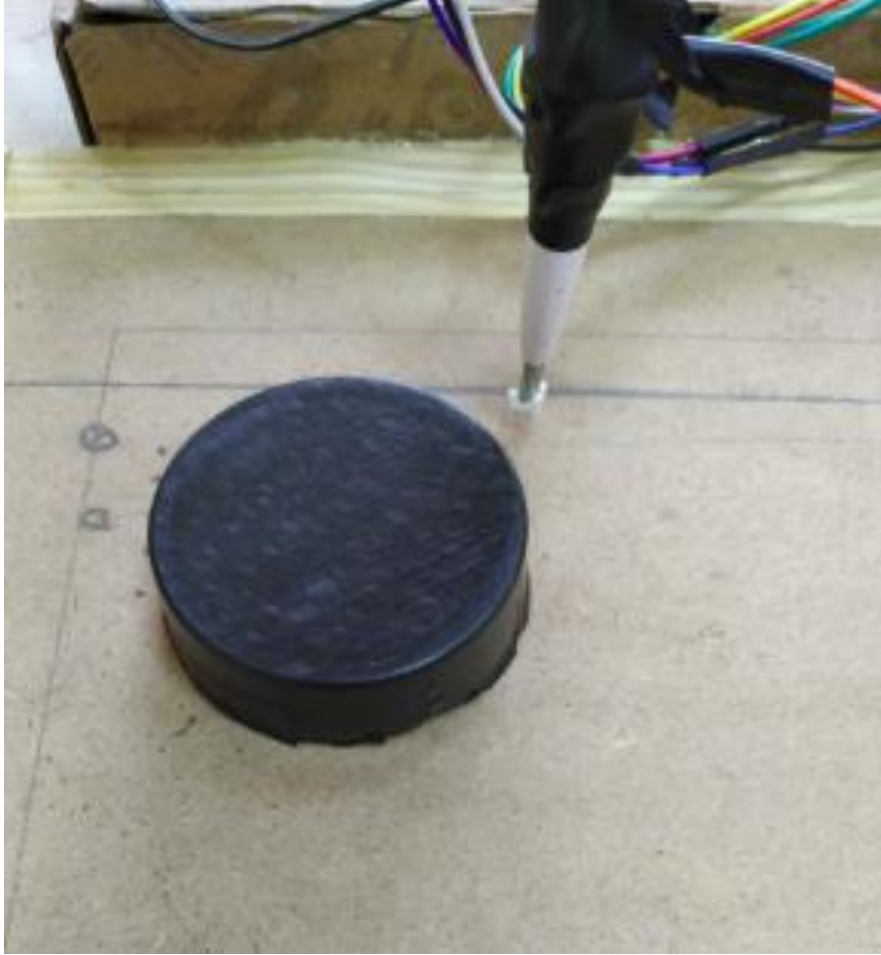
Dönüştürülen üç boyutlu obje dosyası formatı Wavefront Object (.obj) olarak belirlendi çünkü oluşturma ve düzenlenmesi en sade formatın bu olduğuna karar verildi.

Tarama sonuçları 20 x 20 olduğu ve bu sonuçların isabet oranının düşük olması sebebiyle, çıkan yükseklik haritası 40 x 40 olarak yeniden boyutlandırıldı. Bu yeniden boyutlandırma işleminde ara değerleri yakalamak ve yumuşatılmış bir obje elde edebilmek için Bilinear Interpolation algoritması kullanıldı. Bu algoritma, her nokta için yeni noktalar oluşturarak yükseklik değerlerinin ortalamasını bu yeni noktalarının yüksekliği olarak belirledi.

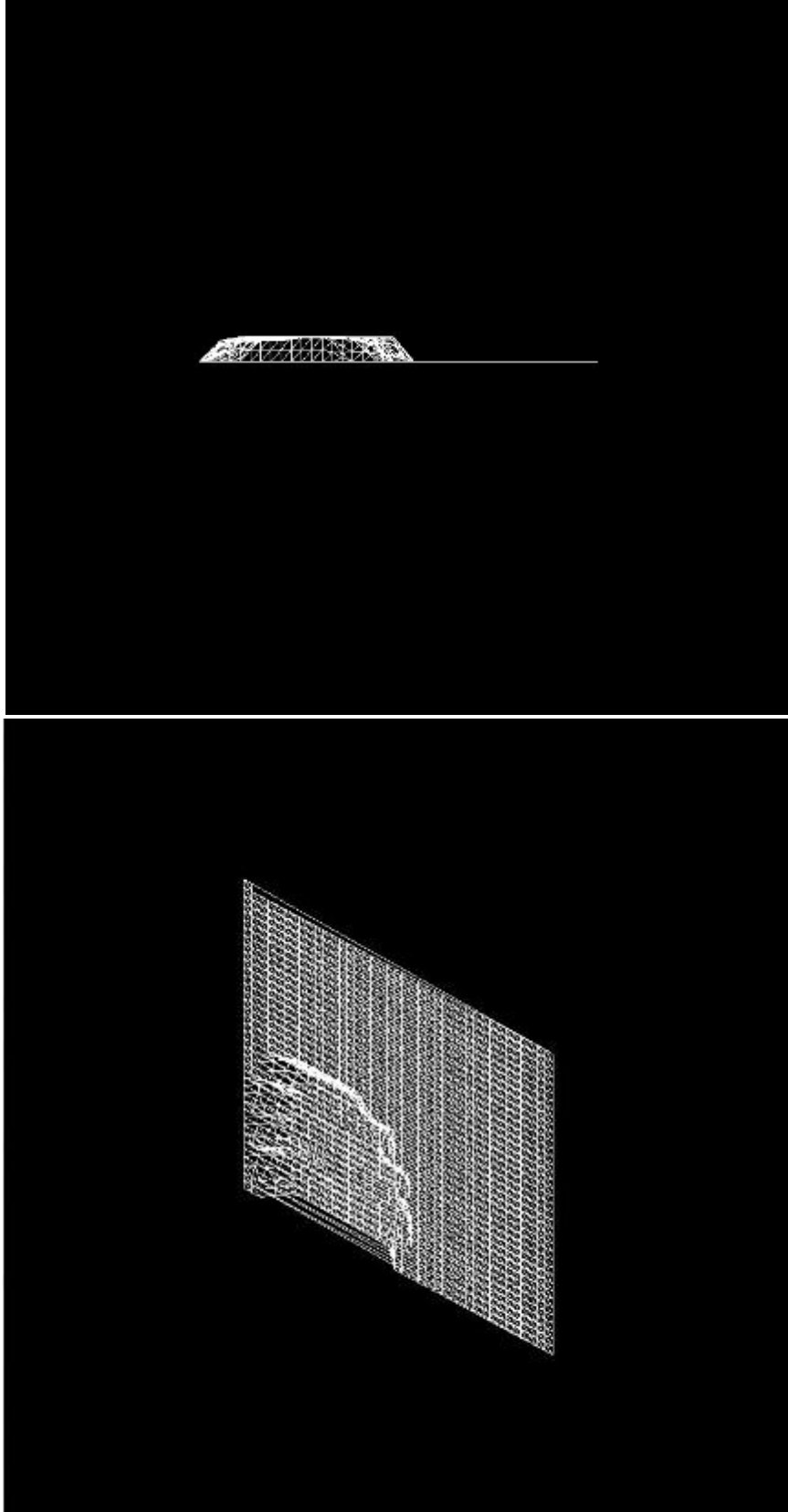
Bu modül, yükseklik değerlerini X, Y koordinatları ve tarama ucunun maksimum - minimum aralığında ne kadar yol katettiğinin 0 ile 1 arasında normalleştirilmiş halini Tarama (Donanım) Modülü'nden alıp, oluşturduğu yükseklik haritasını bir .(obj) dosyası olarak Üç Boyutlu Obje Görüntüleme Modülüne yollamaktadır.

4. Örnek Sonuçlar

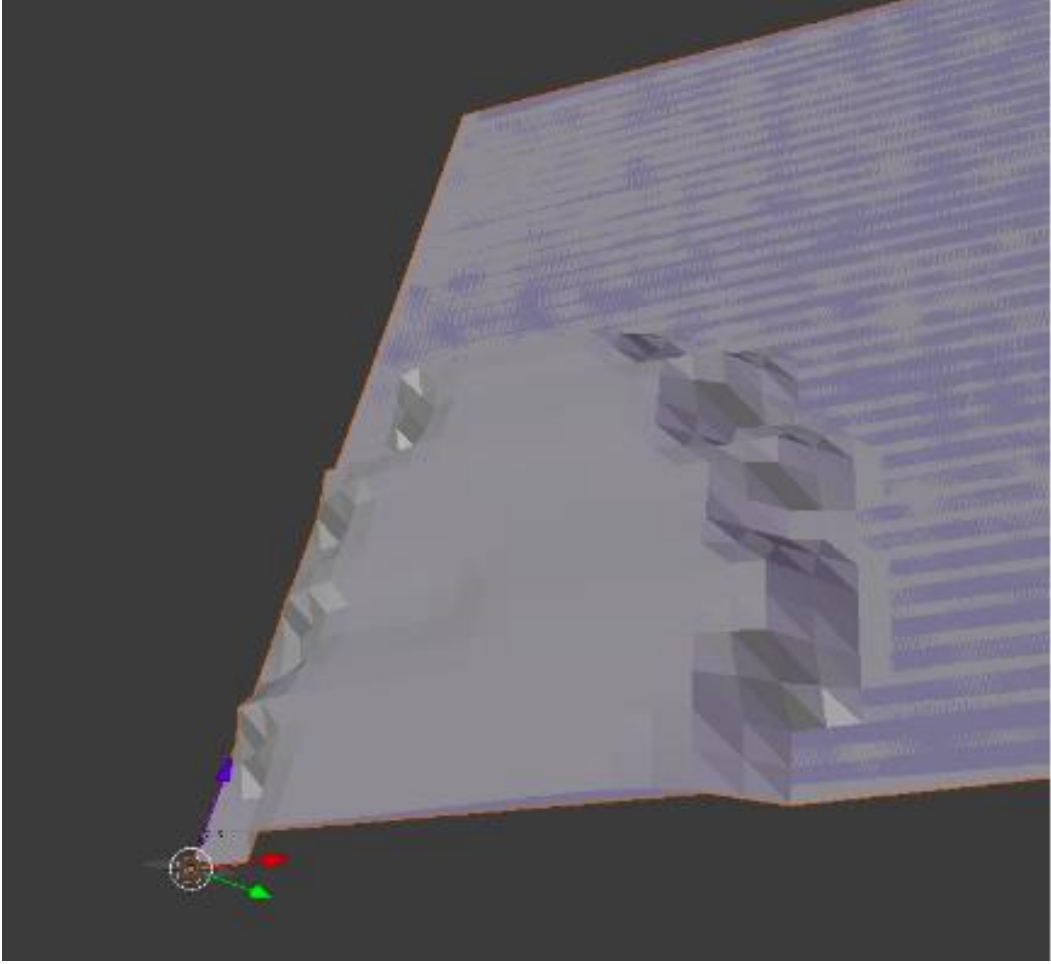
CanTouchThis'i test etmek için şekildeki gibi bir kapak yerleřtirdik.



Taranan cismin yükseklik haritası ařağıdaki görsellerdeki gibi çıkarılmıştır:



Sonuç olarak üretilen üç boyutlu obje şu şekildedir:



CanTouchThis'i test etmek için Őekildeki gibi bir kutu yerleŐtirdik.



Sonuĉ olarak űretilen űĉ boyutlu obje Őu Őekildedir:



5. Maliyet

CanTouchThis'in donanım maliyetleri tablosu aşağıdaki gibidir:

| | |
|---------------------------------|--------|
| Raspberry Pi 3 B | 290 TL |
| 4 Adet Stepper Motor (28byj-48) | 42 TL |
| 4 Adet Yazıcı Kasnak | 24 TL |
| 2 Adet Yazıcı Rulman | 22 TL |
| 4 Metre GT2 Kayış | 40 TL |

6. Grup Üyeleri ve Modüllerin Paylaşımı

CanTouchThis'in geliştirilmesi için ekibimiz aşağıdaki gibi gruplara ayrılarak, her modül birer alt ekip tarafından geliştirilmiştir.

| İsim | Görev Aldığı Modüller |
|-------------------|---|
| Yavuz Selim ERDÖL | Tarama Modüller Arası Haberleşme Veri İşleme ve Dönüştürme |
| Barış ŞAHİN | Tarama Veri İşleme ve Dönüştürme Üç Boyutlu Obje Görüntüleme |
| Yusuf AKDAŞ | Modüller Arası Haberleşme Veri İşleme ve Dönüştürme Üç Boyutlu Obje Görüntüleme |
| Ferhat SENCER | Tarama Modüller Arası Haberleşme Kontrol Arayüzü |
| Ubeydullah ÇOBAN | Tarama Modüller Arası Haberleşme Kontrol Arayüzü |
| Mustafa PASLI | Tarama Modüller Arası Haberleşme Kontrol Arayüzü |
| Emire KORKMAZ | Üç Boyutlu Obje Görüntüleme Kontrol Arayüzü |
| Ahmet Yasir NACAK | Modüller Arası Haberleşme Veri İşleme ve Dönüştürme Üç Boyutlu Obje Görüntüleme |