

3 BOYUTLU GÖRÜNTÜ DOSYALARI

Kocaeli Üniversitesi Bilgisayar Mühendisliği Programlama Laboratuvarı 1 – 1. Proje

Emre Yelbey
Kocaeli Üniversitesi
Bilgisayar Mühendisliği
180202043
emre_yelbey@hotmail.com

Ege Özeren
Kocaeli Üniversitesi
Bilgisayar Mühendisliği
180202047
ege99@yahoo.com

Hazırlanan program kullanıcı tarafından proje kök dizini içerisinde verilen “.nkt” uzantılı dosyaları kullanarak birtakım hesaplamalar ve işlemler yapar.

I. GİRİŞ

Program çalışmadan önce kullanıcı tarafından proje kök dizinine “.nkt” uzantılı dosyalar dahil edilir. Bu dosyaların her biri iki temel bölümden oluşur. İlk programın gerekli işlemleri yerine getirmek için ihtiyaç duyduğu başlık bilgileri bölümü ve ardından belirli parametrelere sahip nokta verilerini içeren bölümdür.

Program çalıştırıldığı anda kullanıcıyı bir menü karşılar. Bu menüde kullanıcının seçip gerçekleştirebileceği işlemler bulunur. Bunlar:

1. Dosya Kontrolü
2. En Yakın & En Uzak Noktalar
3. En Küçük Küp
4. Küre İçerisindeki Noktalar
5. Noktalar Arası Uzaklık Ortalaması
6. Çıkış

şeklindedir. Bu işlemlerin detayı yazının ilerleyen kısımlarında verilecektir.

II. TEMEL BİLGİLER

Program C programlama dilinde geliştirilmiş olup geliştirme ortamı olarak **Code::Blocks** kullanılmıştır.

III. TASARIM

3 Boyutlu Görüntü Dosyaları programının geliştirilme aşamaları belirtilen başlıklar altında açıklanmıştır.

A. Algoritma

1. Dosya Kontrolü

Program çalıştırıldığı anda yapılması gereken ilk işlem dosya kontrolüdür. Hatalı dosyaların hataları ve dosyanın hangi modda okunacağı tespit edilir. Yazının başında belirttiğimiz gibi her dosyanın nokta verilerinden önce bir başlık bölümü bulunur. Bu başlık bölümü:

```
#Noktalar dosya format
VERSION 1
ALANLAR x y z veya ALANLAR x y z r g b
NOKTALAR (verilen nokta sayısı)
DATA ascii veya DATA binary (veri tipi)
```

şeklindedir. Bu bölümden sonra başlıkta belirtilen “ALANLAR” tipine uygun olarak nokta bilgileri verilmiştir. Örneğin dosya başlığında alanlar bilgisi “ALANLAR x y z” şeklinde verilmişse, 79.911334 51.105907 12.947192 şeklinde 3 adet parametreye sahip; “ALANLAR x y z r g b” şeklinde verilmişse, 79.911334 51.105907 12.947192 95 227 28 şeklinde 6 adet parametreye sahip nokta bilgileri verilmiştir.

Dosya kontrolü işleminde noktanın bu formata uyumlu olup olmadığı kontrol edilir. İlk olarak kontrol edilen kısım başlık bilgilerinin doğruluğudur. Örneğin data tipi bilgisi “DATA bnsry” şeklinde verilmişse bu bir hatadır. Bu şekilde tüm dosyaların ilk olarak başlık bilgileri kontrol edilir. Eğer dosya formata uygun değilse hatalı olarak nitelendirilir ve hesaplama işlemlerine dahil edilmez.

Daha sonraki adım dosyada bulunan nokta sayısı ve bilgilerinin doğru olup olmadığıdır. (Bu adımdaki hatalar nokta bilgileri okunurken tespit edilir. Bu işlem daha sonra detaylı olarak açıklanacaktır.) Örneğin noktalar açıklaması “NOKTALAR 391” şeklinde verilip gerçekte daha az veya daha fazla nokta bulunuyorsa bu bir hatadır. Ayrıca noktaların başlık kısmında verilen “ALANLAR” formatına uygunluğu da kontrol edilmektedir. Örneğin dosya alanlar tipi “ALANLAR x y z” şeklinde verilip, nokta 79.911334 51.105907 şeklinde verilirse bu bir parametrenin eksik olduğu anlamına gelir ve bu da bir hatadır ve bu dosyalar da aynı şekilde hesaplama işlemlerine dahil edilmez.

Bu işlemlerden sonra hatasız dosyaların nokta bilgileri bir “struct” yapısında saklanır ve gerçekleştirilecek işlemler için saklanmak üzere kaydedilir.

2. En Yakın & En Uzak Noktalar

Bu işlemde her dosya için kaydedilen noktaların birbirlerine olan uzaklıkları “iki nokta arası uzaklık formülü” (Kullanılan Bazı Matematiksel Formüller kısmında detaylı olarak açıklanacaktır.) ile hesaplanır. En küçük uzaklık birbirine en yakın olan noktalar, en büyük uzaklık ise birbirine en uzak olan noktalar tespit edilir.

Ek olarak en çok hesaplama yapılan ve zaman tüketen işlem tüm noktaların birbirlerine olan uzaklıklarının tespit edilmesidir. Program “n” adet nokta için $(n-1).n/2$ adet hesaplama işlemi gerçekleştirir. Daha anlaşılır ifadeyle 10000 nokta için program 4.999.500 adet hesaplama yapar. Nokta sayısı arttıkça hesaplama süresi giderek artmaktadır.

(Bu durum için 5. işlemde özel bir koşul tanımlanmıştır. Bkz. 5. Noktalar Arası Uzaklık Ort.)

3. En Küçük Küp

Bu seçimde yapılan işlem verilen dosyadaki tüm noktaları içine alan en küçük küpün köşe noktalarını hesaplamaktır. Noktaları hesaplamak için ilk olarak minimum x, minimum y ve minimum z değerlerine sahip olan noktayı referans kabul etmek amacıyla tespit etmektir. Minimum değerlere sahip nokta tespit edildikten sonra yapılan işlem tüm noktaları içine alabilecek en küçük küpün bir ayırıtının uzunluğunu tespit etmektir. Bu ayırıtın uzunluğunun hesaplanması için tüm eksenlerde noktalar arası maksimum uzaklık

hesaplanır. Daha sonra bu uzunluk ve referans köşe noktası kullanılarak küpün diğer 7 noktası hesaplanır.

4. Küre İçerisindeki Noktalar

Bu seçimde ilk olarak kullanıcıdan bir merkez noktası (x y ve z değerleri) ve yarıçap değeri girmesi istenir. Daha sonra her dosya için tanımlanan bu küre içerisinde kalan noktalar hesaplanır.

Bu hesaplamada yine 2. işlemde (en yakın & en uzak noktalar) de olduğu gibi iki nokta arası uzaklık formülünden yararlanılır. Her nokta ile verilen merkez arasındaki uzaklık hesaplanır. Eğer bu uzaklık yarıçaptan küçük veya yarıçapa eşitse, karşılaştırılan nokta küpün içerisinde veya küpün yüzeyinde bulunur.

5. Noktalar Arası Uzaklık Ortalaması

Bu işlemde yapılan hesaplama 2. işlemde olduğu gibi tüm noktaların birbirlerine olan uzaklıkların toplamı uzaklık sayısına bölünmesidir.

2. işlemin sonunda bahsedilen koşul 2. işlemin daha önce yapılıp yapılmadığıdır. Eğer ikinci işlem 5. işlemde daha önce yapılmış ise işlem süresini kısaltmak için tekrardan noktalar arası uzaklıklar hesaplanmaz. Bu durum daha önceden öngörülüp her dosya için noktalar arası uzaklıkların toplamı ve toplam uzaklık sayıları saklanmıştır. Bu sayede daha önceden 2. işlem yapılmış ise 5. işlemin sonucu anında verilebilmektedir.

6. Çıkış

Kullanıcının menü döngüsünü kırıp programı sonlandırmasını sağlar. Aksi takdirde program çalışmaya devam edecektir.

B. Yöntem

Programda hesaplama işlemlerinde kullanmak için gereken nokta verilerini tutmak amacıyla bir adet struct yapısı tanımlanmıştır. Program çalıştığında proje kök dizininde bulunan her dosya için bir adet struct yapısı üretir ve hatasız dosyaların nokta verilerini bu struct yapılarında saklar.

C. Kullanılan Fonksiyonlar

Programın anlaşılabilirliğini arttırmak ve yazımı kolaylaştırmak için çok sayıda fonksiyon kullanılmıştır.

- *void menuyu_goster();*

Terminal ekranında uygulama menüsünü gösterir.

- *void nkt_dosyalarini_al();*

Proje kök dizinindeki “.nkt” uzantılı dosyaları tespit edip saklar.

- *void her_dosya_icin_bellek_ayir();*

Dosya sayısı kadar struct yapısı oluşturur.

- *void dosyolari_oku_ve_kontrol_et();*

Dosyalar hata kontrolü fonksiyonlarına gönderilip bilgileri kaydedilir.

- *int baslik_hata_kontrolu(FILE *dosya);*

Dosyaların başlık kontrolü yapılır, hatalı dosyalar tespit edilir.

- *int ascii_deger_oku(FILE *dosya);*

Data tipi “ascii” olan dosyaların nokta değerleri okunur.

- *int binary_deger_oku(FILE *dosya);*

Data tipi “binary” olan dosyaların nokta değerleri okunur.

- *int nokta_bilgi_kontrolu(char satirTutucu[],int satir);*

Her noktanın parametre kontrolü yapılır.

- *void enyakin_ve_enuzak_noktalar();*

En yakın ve en uzak noktalar tespit edilir.

- *void ortalama_hesapla();*

Dosyadaki tüm noktaların birbirine olan uzaklıklarının ortalaması hesaplanır.

- *void küre_hesapla();*

Merkezi ve yarıçapı kullanıcı tarafından girilen kürenin içerisinde kalan noktaları hesaplar.

- *void kup_hesapla();*

Dosyadaki tüm noktaları içine alan en küçük küpün köşe noktalarını hesaplar.

- *void output_olustur();*

Kontroller ve hesaplamalar sonunda elde edilen sonuçları “output.nkt” dosyasına kaydeder.

D. Kullanılan Bazı Denklemler

İki Nokta Arası Uzaklık Formülü

$$|AB| = ((Ax-Bx)^2 + (Ay-By)^2 + (Cz-Bz)^2)^{1/2}$$

IV. SÖZDE KOD

1. Başla
2. Programın bulunduğu dizini aç
3. “.nkt” uzantılı dosyaları bul ve kaydet
4. Her dosya için bellek alanı ayır
5. Uygulama menüsünü göster
6. Kullanıcıdan bir seçim değeri al
7. Seçim değeri 1 ise dosyaları oku ve kontrol et
8. Her dosyanın başlık bilgilerini oku, başlık hatası bulunan ve bulunmayan dosyaları tespit et, işaretle
9. Sonuçların kaydedilmesi için gereken output dosyasını oluştur ve hatalı dosyalarda bulunan hataları bu dosyaya yazdır (“output.nkt”)
10. Dosya bilgilerini kaydet
11. Dosya tipi binary ise binary okuma fonksiyonuna, ascii ise ascii okuma fonksiyonuna gönder
12. Her dosyayı satır satır oku; nokta verileri belirtilen noktalar tipiyle eşleşmiyorsa tespit et ve işaretle, başlıkta belirtilen nokta sayısı verilen nokta sayısı ile eşleşmiyorsa tespit et ve işaretle
13. Bu işlemlerden sonra dosya kontrolü tamamlanmış olup tüm dosyaların tüm hataları tespit edilmiştir. Bu adımdan sonra tekrar menü ekranı açılır.

14. Menü ekranına dön
15. Kullanıcıdan bir seçim değeri al
16. Seçim değeri 2 ise hatalı olmayan her dosya için tüm noktaların birbirlerine olan uzaklıklarını hesapla ve kaydet. Bu adımda aynı zamanda bu uzaklıkların sayısı ve toplamı da kaydedilir. (Bkz 5. Noktalar Arası Uzaklık Ortalaması)
17. Hesaplanan en yakın ve en uzak noktaları “*output.nkt*” dosyasına yazdır
18. Menü ekranına dön
19. Kullanıcıdan bir seçim değeri al
20. Seçim değeri 3 ise hatalı olmayan her dosya için tüm noktaları içinde barındıran en küçük küpün köşe noktalarını hesapla ve kaydet
21. Hesaplanan köşe noktalarını “*output.nkt*” dosyasına yazdır
22. Menü ekranına dön
23. Kullanıcıdan bir seçim değeri al
24. Seçim değeri 4 ise kullanıcıdan bir küre tanımlaması iste. Önce merkez noktası olarak kabul edilecek “*x y ve z*” değerlerini, ardından yarıçap uzunluğunu iste
25. Hatalı olmayan her dosya için kullanıcı tarafından tanımlanan küre içerisinde kalan noktaları tespit et ve kaydet
26. Hesaplanan küre içerisindeki noktaları “*output.nkt*” dosyasına yazdır
27. Menü ekranına dön
28. Kullanıcıdan bir seçim değeri al
29. Seçim değeri 5 ise öncelikle daha önce 2. İşlemin yapıp yapılmadığını kontrol et. 2. işlem daha önce yapılmışsa her dosya için kaydedilen uzaklık toplamı ve uzaklık sayısını kullanarak ortalamaları hesapla. 2. işlem daha önce yapılmamışsa tüm noktaların birbirlerine olan uzaklıklarını ve uzaklık sayısını hesaplayarak ortalamaları bul
30. Hatalı olmayan her dosya için hesaplanan ortalama değerlerini “*output.nkt*” dosyasına yazdır

31. Menü ekranına dön

32. Kullanıcıdan bir seçim değeri al

33. Seçim değeri 6 ise programı sonlandır (Seçim değeri 6 girilmediği takdirde program çalışmaya devam edecektir.)

V. SONUÇLAR

Proje kök dizinde verilen tüm “*.nkt*” dosyalarını ayıklar, hataları tespit eder ve hesaplamalar sonucu elde edilen verileri başarıyla kaydeder.

VI. KAZANIMLAR

Projenin bize kattıkları:

- Bu projeden önce, dosya işlemlerinde dosya isimlerini manuel olarak ve bir kez girmektedirdik. Bu proje ile beraber çalıştığımız dizindeki dosya isimlerine ulaşmayı ve bu dosyaları sıralı şekilde açarak kullanabilmeyi öğrendik.
- Daha başarılı bellek yönetimi nasıl yapılabilir fark ettik. Sık sık “*malloc*” ve “*realloc*” fonksiyonları kullandık.
- Karşılaştığımız sorunları analiz etme ve çözümleme yeteneğimizin geliştiğine inanıyorum.

VII. KARŞILAŞILAN SORUNLAR VE ÇÖZÜMLERİ

1. İlk büyük problem ile binary formatında değer okuma kısmında karşılaştık. Ascii formatta okumada “*fgets*” fonksiyonuyla kolay bir şekilde değerleri alıp struct yapısına kaydedebilirken binary kısmında bu işlem biraz zorlanmamıza sebep oldu. Sorunu “*fread*” fonksiyonunun matığını anladığımızda çözebildik.
2. İkinci karşılaştığımız sorun output oluşturma kısmındaydı. Output işlemleri en sona kaldığında bazı gerekli bilgileri kaydetmediğimizi fark ettik. Bu sorunu eski fonksiyonlarımızda düzenlemeler yaparak aştık.

VIII. KAYNAKÇA

- [1] <https://stackoverflow.com/questions/3554120/open-directory-using-c>
- [2] Deitel, P. and Deitel, H., 2013. *C ile Programlama*. 7th ed. USA: Pearson Education Inc..
- [3] https://drive.google.com/open?id=1o0eQ8ufppL1jhByWKZrp-YXzV_mVPWGZ
- [4] https://drive.google.com/open?id=1hbrAuglv2jfTf1Zf8yTw3Fv07bO_A7P0
- [5] <https://drive.google.com/open?id=1GKTD6-zhsAbAC1Lg2-V3Nbl-GDbWKsdw>
- [6] <https://orion.math.iastate.edu/dept/links/formulas/form2.pdf>