T.C.
AKDENİZ ÜNİVERSİTESİ

HAZIRLAYAN
Büşra OLGUN
20163405002

FEN FAKÜLTESİ

UZAY BİLİMLERİ VE TEKNOLOJİLERİ BÖLÜMÜ

LASER / LİDAR DERSİ

UYGULAMA RAPORU

Aralık 2021
ANTALYA

İçindekiler

GİRİŞ

LASER verilerinin görselleştirilmesi ve analizinin gerçekleştirilmesi üzerine python programlama dilini, QGIS, SAGA GIS, Cloud Compare ve Aldpat uygulamalarını kullanarak işlemler yaptık. Bu raporda bu uygulamaların yapılış şekilleri, sonuçları ve yorumları yer almaktadır.

1. Python Programlama Dilinde 8 Bitte Görselleştirme Algoritması

Programın işleyişi şöyledir;
Opentopography sitesinden indirilmiş txt formatında olan nokta bulutu verisinden z değerlerini yani yükseklik değerlerini çekerek, bu yükseklik değerlerinin max ve min değerini bulup yazar.

$$intensity = \frac{z_i - z_{min}}{z_{max} - z_{min}} \times 255$$

Ardından yukarıdaki formülü her yükseklik değerine uygulayarak bir intensity değer listesi oluşturur. Bir txt formatında dosya oluşturarak, x,y,z,intensity değerlerinin olduğu nokta veri tablosunu bu dosya içine yazar.

Yazdığım kodlar şu şekildedir;

```
import pandas as pd
from pandas import read_csv,DataFrame, concat

# Dosyayı açtı.
points = pd.read_table("points.txt" , sep = ',')

#Tablodaki değerleri sütun olarak çekmek için kullanılacak fonksiyon
def sutun_liste(a):
    a_ = []
    for line in range(len(a)):
        a_line = float(a[line])
        a_.append(a_line)
    return a_

# Tablodaki sütunları listeye çevirir.
x=points.iloc[:,0:1].values
x_ = sutun_liste(x)
y=points.iloc[:,1:2].values
y_ = sutun_liste(y)
z=points.iloc[:,2:3].values
z_ = sutun_liste(z)

# Yüksekliklerin oluştuğu listede minimum ve maximum değerleri bulma
zmin = z_[0]
for zi in z_:
    if zi < zmin:
        zmin = zi

zmax = zmin
for zi in z:
    if zi > zmax:
        zmax = zi
print ("Maksimum Yükseklik:" , zmax , "Minimum Yükseklik:", zmin)

# Her bir değer için piksel değerindeki karşılığını bulma
intensity = []
for zi in z_ :
    g = ((zi - zmin) / (zmax - zmin))*255
    intensity.append(g)
```

```
# x, y, z ve intensity değerlerini bir tablo şeklinde txt dosyasına yazma
with open('visualitons.txt','w') as p:
    p.write ("x, y, z, intensity")
    p.write('\n')
    for x,y,z,intensity in zip(x_,y_,z_,intensity):
        p.write("{}, , ,  ".format(x,y,z,intensity))
    p.close()

plt.scatter(x_,y_,marker = 'o',s = 2)
plt.show()
```

Yazılan program sonucunda 'z' yüksekliğinin maksimum değeri 2304.87 ve minimum değeri 2267.13 olarak bulundu. QGIS programında nokta verisi açılıp, özellikler tablosuna bakıldığında z yüksekliği için minimum ve maksimum değerler aynı çıkmıştır.

Görüntüler de ise python kodu ile oluşturulan veride x ekseninde daha fazla yayılmış gözükmekte, QGIS programında açılan veride y ekseninde daha fazla yayılmış gözükmektedir. Fakat bunun sebebi x ve y ekseninde kullanılan skalaların farklı olmasından dolayı verilerin yayılımlarındaki aralıkların farklılık göstermesindendir.

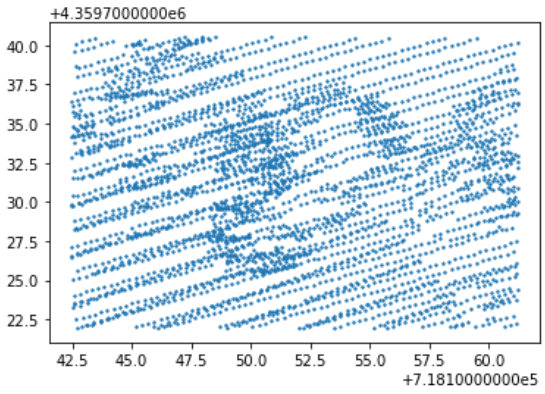


Figure 1: Python Kodları Sonucunda Oluşan Nokta Bulutu Verisi

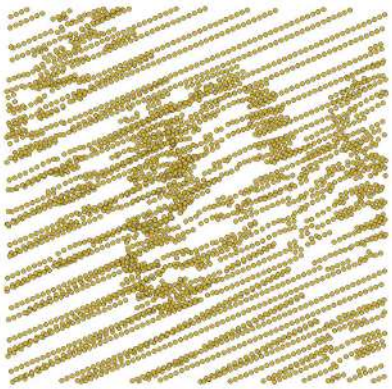


Figure 2: QGIS Uygulamasında Açılan Nokta Bulutu Verisi

2. Ham Veriden Grid Veriye Dönüşüm Kodları

Programın işleyişi şöyledir;
Opentopography sitesinden indirilmiş txt formatında olan nokta bulutu verisinden x, y ve z değerlerini liste halinde çekerek minimum ve maksimum değerlerini elde eder. Bu değerlerle minimum değerlerden düşük, maksimum değerlerden yüksek sınır değerleri oluşturur. Bu değerler arasında grid oluşturur ve bu gridlerin değerleri 5 birim aralıklarla artmaktadır. Ayrıca bu nokta verilerinde görselleştirilmesi bu grid üzerinde yapılmıştır.

```
import math
import pandas as pd
from pandas import read_csv,DataFrame, concat
import matplotlib.pyplot as plt

# Dosyayı açtı.
points = pd.read_table("points.txt" , sep = ',')
```

```
#Tablodaki değerleri sütun olarak çekmek için kullanılacak fonksiyon
def sutun_liste(a):
    a_ = []
    for line in range(len(a)):
        a_line = float(a[line])
        a_.append(a_line)
    return a_

# Tablodaki sütunları listeye çevirir.
x=points.iloc[:,0:1].values
x_ = sutun_liste(x)
y=points.iloc[:,1:2].values
y_ = sutun_liste(y)
z=points.iloc[:,2:3].values
z_ = sutun_liste(z)

# x,y ve z verilerinde minimum ve maksimum değerlerinde kullanılacak fonksiyon
xmin=x_[0]
for xi in x_:
    if xi<xmin:
        xmin=xi

xmax=xmin
for xi in x_:
    if xi>xmax:
        xmax=xi
print("x minimum:" , xmin , "x maksimum:" , xmax)

    ymin=y_[0]
for yi in y_:
    if yi<ymin:
        ymin=yi

ymax=ymin
for yi in y_:
    if yi>ymax:
        ymax=yi
print("y minimum:" , ymin , "y maksimum:" , ymax)

    zmin=z_[0]
for zi in z_:
    if zi<zmin:
        zmin=zi

zmax=zmin
for zi in z_:
    if zi>zmax:
        zmax=zi
print("z minimum:" , zmin , "z maksimum:" , zmax)
```

```
xmin = xmin - (xmin % 5)
xmax = xmax - (xmax % 5) + 5
ymin = ymin - (ymin % 5)
ymax = ymax - (ymax % 5) + 5
xminbes = xmin
yminbes = ymin
x=[]
y=[]
while xmax>=xminbes and ymax>=yminbes:
    x.append(xminbes)
    y.append(yminbes)
    yminbes += 5
    xminbes +=5
print("x ekseni:", x)
print("y ekseni:", y)

# Grid oluşturma kodu
plt.scatter(x_, y_ , marker ='o', s=2)
plt.xlim((x[0], x[-1]))
plt.ylim((y[0], y[-1]))
plt.grid(True)
plt.show()
```

Programı çalıştırdığımda aldığım çıktılar şunlardır:
x minimum: 718142.44 x maksimum: 718161.24
y minimum: 4359721.97 y maksimum: 4359740.52
z minimum: 2267.13 z maksimum: 2304.87
x ekseni: [718140.0 , 718145.0, 718150.0, 718155.0, 718160.0, 718165.0]
y ekseni: [4359720.0, 4359725.0, 4359730.0, 4359735.0, 4359740.0, 4359745.0]

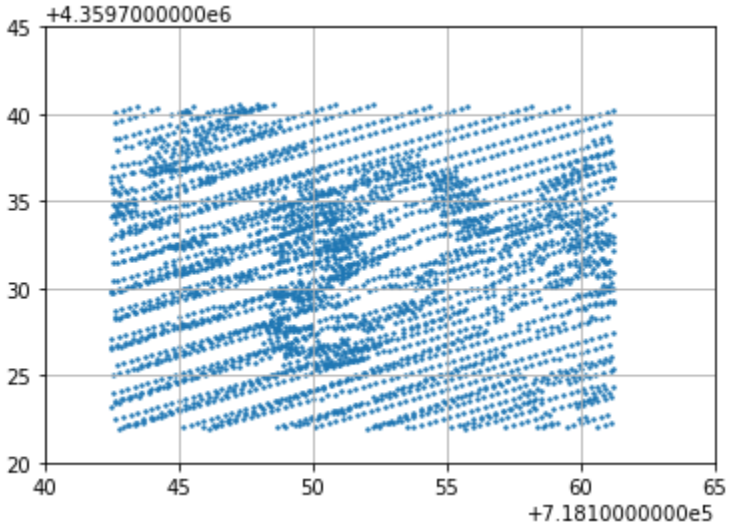
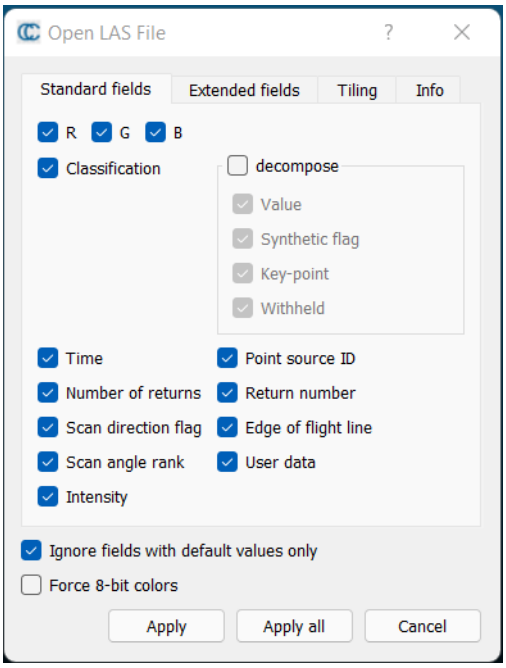
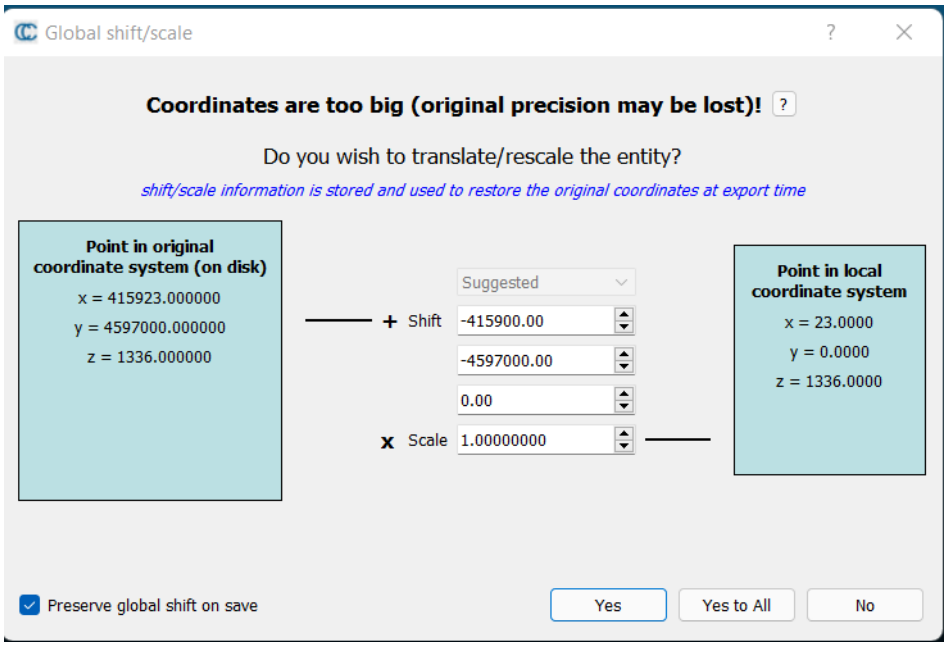


Figure 3: Nokta Veriye Grid Uygulama

3. Cloud Compare Uygulamasında Las Formatındaki Verinin Görselleştirilmesi
Opentopography sayfasından ABD'nin Utah eyaletindeki Brigham City bölgesinden Las formatında veri indirildi. Cloud Compare uygulamasında open diyerek las formatındaki veri açıldı. Uygulamada şu şekilde bir sekme açıldı.



Açılan pencerede işaretli olan her bir parametre nokta üzerine yazılmaktadır. Buradan apply all denildiğinde uygulama aşağıdaki sekmeyi açmaktadır.



Açılan sekmedeki nesnelerin koordinatları ne kadar öteleme yaptıkları ve oluşturulan koordinatlar yazmaktadır. Burada program yaptığı koordinat dönüşümünü yazmıştır. Yes to All diyince program görüntüyü açar.

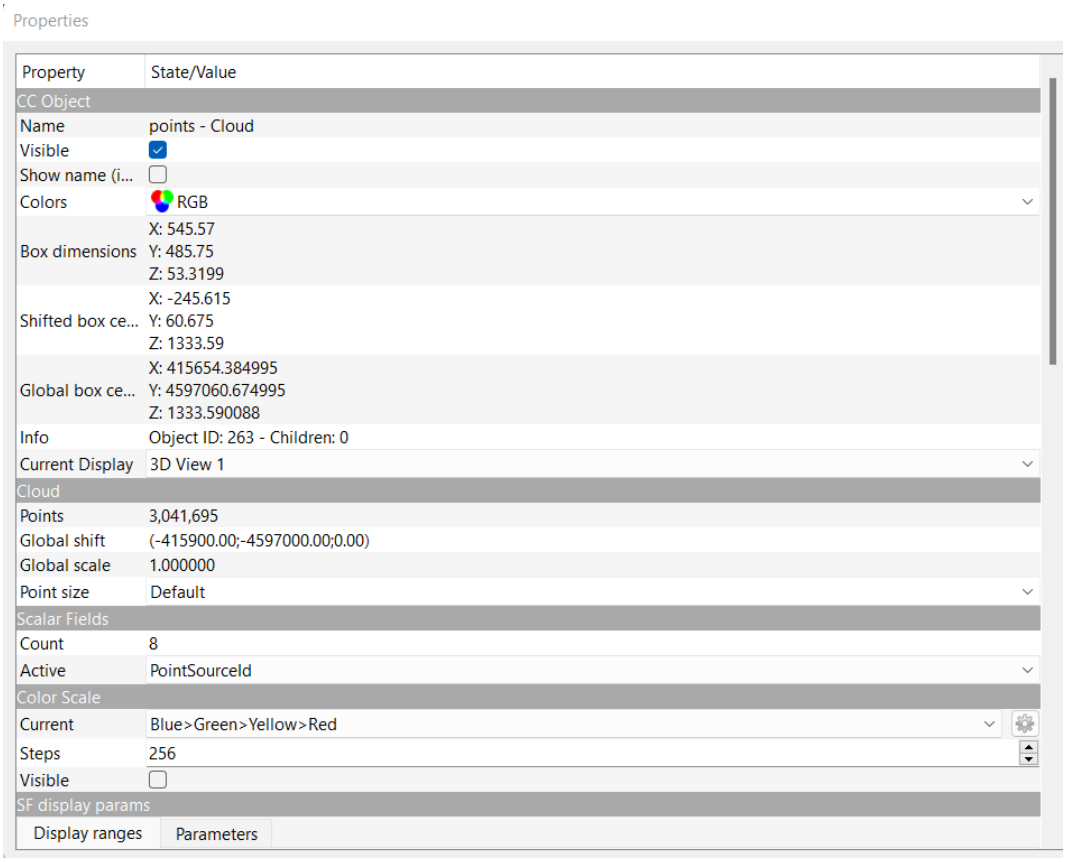


Figure 4: Properties

Burada Colors kısmından RGB ve Scalar Field olmak üzere iki seçenek vardır. RGB seçildiğinde veri uydu görüntüsü şeklinde gözükmektedir. Scalar Field seçildiğinde ise Active kısmından farklı şekillerde görüntülenmesini sağlarız. Ayrıca bu program sayesinde noktaları 3 boyutlu şekilde görebiliriz.

Figure 5: RGB Seçildiğinde Oluşan Görüntü

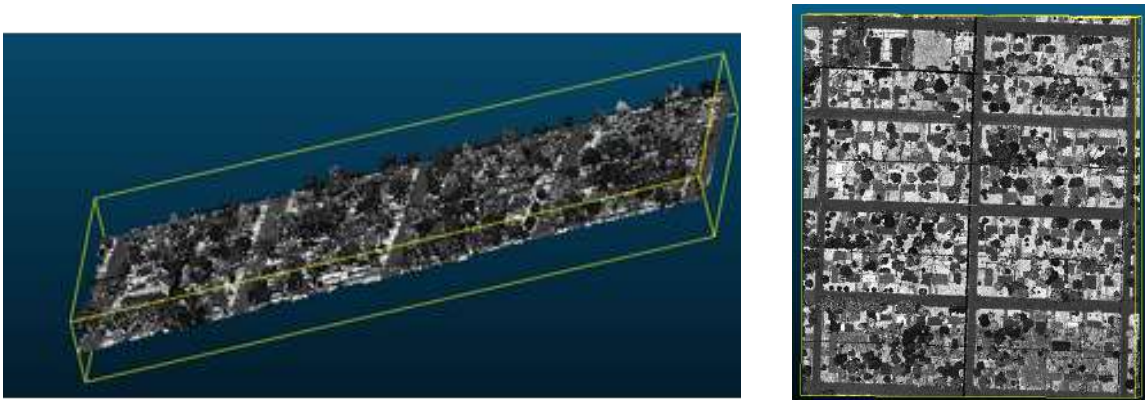


Figure 6: İntensity Seçildiğinde Oluşan Görüntü

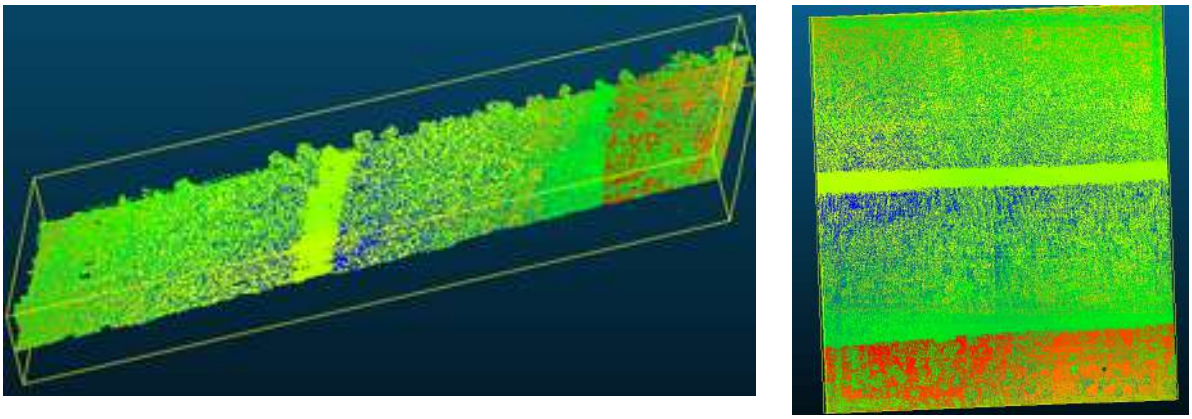


Figure 7: Scan Angle Rank Seçildiğinde Oluşan Görüntü

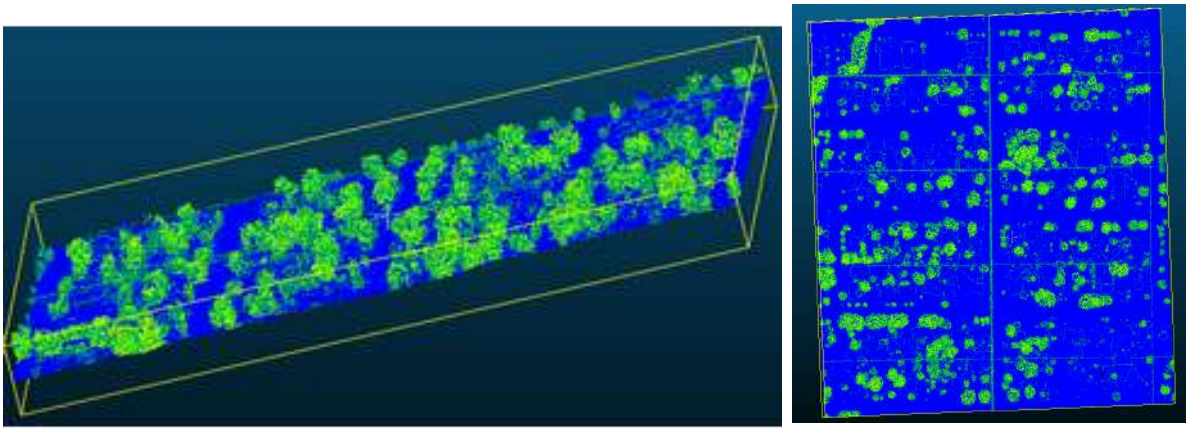


Figure 8: Number of Returns Seçildiğinde Oluşan Görüntü

Farklı görselleştirmede farklı yapılar öne çıkmaktadır. Mesela intensity verisinde z değerlerine göre sınıflandırırken ve yüksek yerleri siyah gösterirken, Number of Returns yapıldığında sadece ağaç kısımları yeşil gösterilmiştir ve bu sebeple bitki yapısı kolaylıkla tespit edebilebilir.

Ayrıca görüntülere bakıldığında bina alanlarının taban kısmında veri yoktur. Çünkü Laser verisi binalarda çatıdan geçip yerden veri alamaz. Ama bitkilere bakıldığında yoğunluğu fazla olan bir veri görebiliriz.

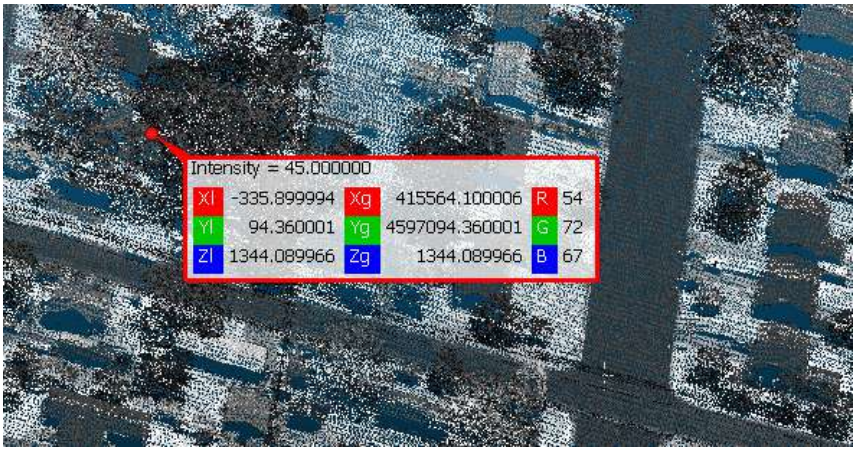


Figure 9: Caption

Uygulama üstünde point picking'e tıklanıldığında ve verideki bir noktanın üstüne tıklandığında noktanın yukarıdaki görüntüde olduğu gibi şu andaki nokta verileri, ötelenme uygulanmadan önceki nokta verileri ve RGB verilerinin değerleri görülür. Ayrıca intensity verisine bakarkende burada seçilen noktanın intensity değeri gözükmektedir.

Edit - Normals - Compute denildiğinde verinin yüzey normalleri hesaplanmaktadır. Hesaplandıktan sonra bir noktanın infosuna bakıldığında her eksen için bir noktanın yüzey normali verileri gözükmektedir.

4. Cloud Compare Uygulamasında Veri Üzerine Yapılan İşlemler

Cloud Compare uygulamasında las formatında açılan görüntü rasterize kısmına tıklanarak aşağıdaki şekilde rastera dönüştürüldü. Böylelikle grid veriye dönüştürüldüğü için işlem yaparken veri boyutu daha küçük olacaktır. Bu sebeple de işlem yaparken süreç daha kısa olacaktır. Fakat veri kaybına uğradığı için doğruluk azalacaktır.

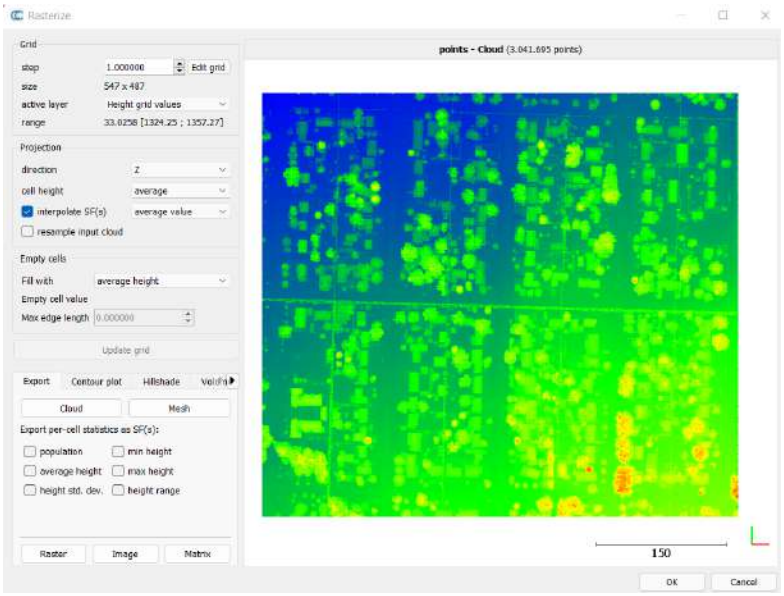


Figure 10: Görüntünün Rastera Dönüştürülmesi

Cloud Compare uygulamasında açılan görüntü üzerinde Tools - Clean - SOR Filter yapıldığında görüntüdeki gürültüler giderilmiş olur. Ayrıca uygulamada Noise Filter algoritmasıda mevcuttur. Fakat en doğru sonucu SOR Filtresi vermektedir. Böylelikle gürültüler olmadığı için yapılan analizlerden daha doğru sonuçlar elde edilir. Bu sebeple veri işlemeden önce yapılması gereken bir uygulamadır.

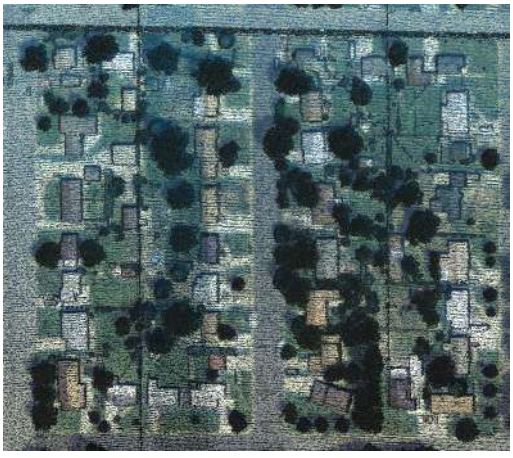


Figure 11: Görüntünün İlk Hali

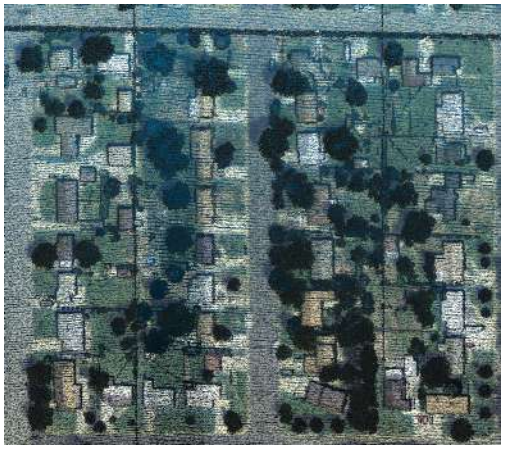


Figure 12: Görüntüye SOR Fitre Uygulanmış Hali

Cloud Compare uygulamasında Tools - Statistics - Compute 2.5D Volume yapıldığında çalışılan alanda hacim hesabı yapıldı. İşlem sonucunda görüntüdeki nesnelerin yükseklikleri göreli olarak elde edildi.

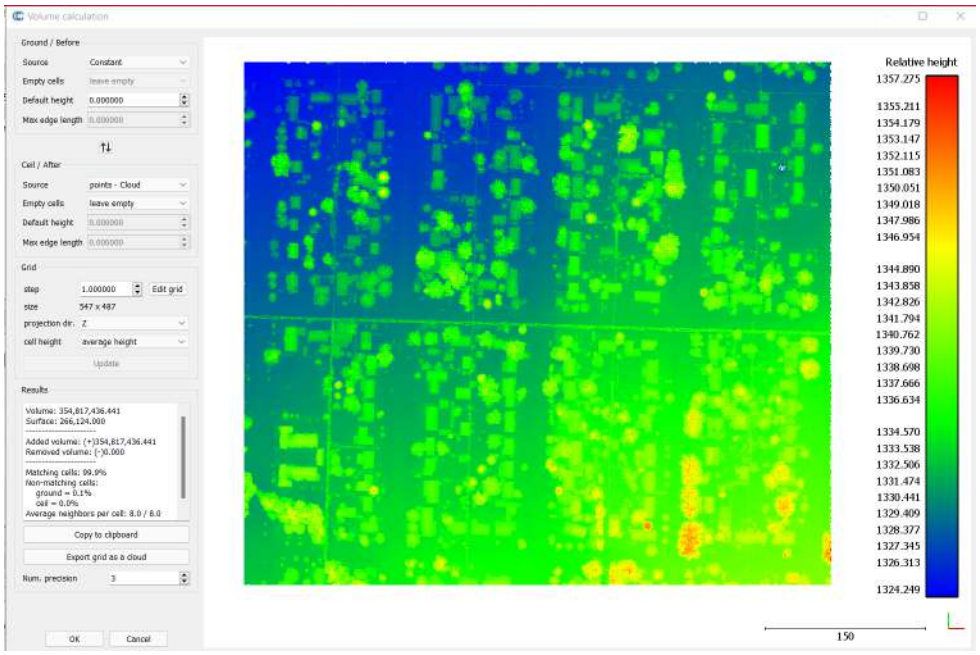


Figure 13: Hacim Hesabı

Cloud Compare uygulamasında açılan görüntü üzerinde Edit - Mesh - Delaunay 2.5D (x,y plane) yapıldığında noktalar arasında üçgenleme yapıldı. Böylelikle verideki noktalar üçgene dönüştüğü için kenar ve yükseklikler kolaylıkla gözükmektedir. Bu sayede hatalı sonuçlar doğruya daha çok yaklaştırıldı.

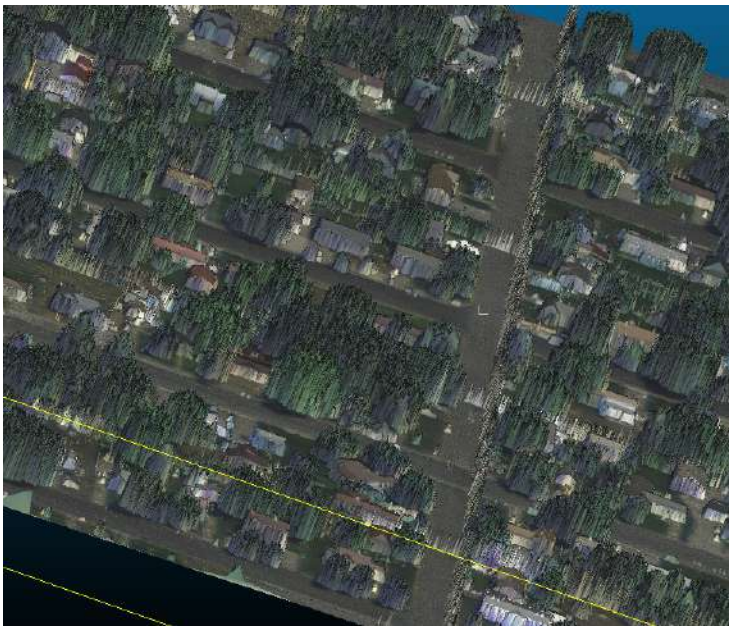


Figure 14: Delaunay 2.5D Algoritması Sonucu Oluşan Görüntü

Cloud Compare uygulaması SOR filtreleme uygulanmış görüntüyü intesity olarak görselleştirmeden sonra Tools - Statistics - Local Statistical Test yöntemiyle gauss uygulandığında aşağıdaki görüntü elde edilir. Bu görüntüde genellikle ağaçlar yeşil, evlerin çatıları ve yollar beyaz, zemin ise kırmızı gözükmektedir. Fakat burada bazı evlerin çatılarının malzemeleri farklı olduğu için beyaz dışında renklerde görülür. Ağaçlara yakın olan binalarda yeşil olarak gözükebilmektedir.

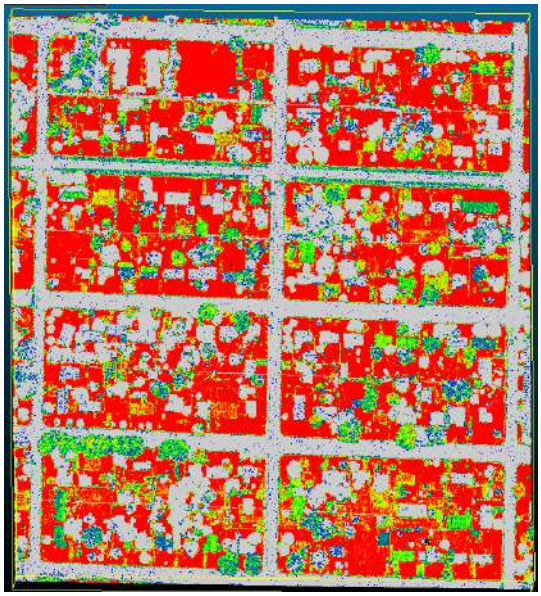


Figure 15: Local Statistical Test Uygulandıktan Sonra Oluşan Görüntü

Cross Section eklentisi ile veride incelenecek kısımda bir en-kesit oluşturularak kesitte kalan alanlar içerisinde bakılan eksen yönündeki cisimlerin birbirleri ile olan farkları incelenebilir. Burada önemli olan ise verinin doğruluğudur.

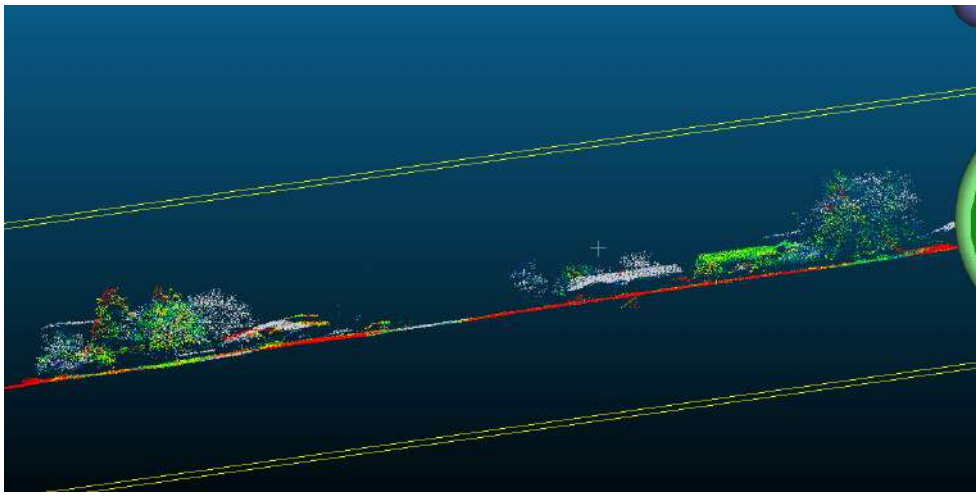


Figure 16: En Kesit Oluşturulmuş Görüntü

5. Eğimin Hesaplanması

Opentopography üzerinden indirilen nokta veri QGIS uygulamasında açıldıktan sonra export edildi. Ardından Raster - Analiysis - Grid (Nearest Neighbor) uygulanarak raster veriye çevrildi. Raster veriye ise Slope uygulandı. Burada eğim verisinin piksel değerleri 0.00321668 ile 497.751 arasında olmasına rağmen, histogramına bakıldığında eğimin ağırlıklı olduğu değer 2,75 çıktı.

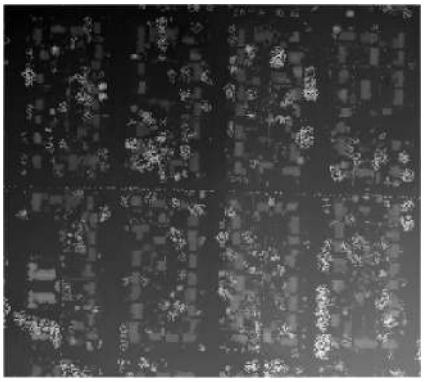


Figure 17: Nearest Neighbor İle Oluşan Görüntü

Figure 18: Slope Uygulandığında Oluşan Görüntü

Opentopography üzerinden indirilen las formatındaki veri Cloud Compare uygulamasında açıldı. Ardından SOR filtresi uygulanarak gürültülerden arındırıldı. Rasterize eklentisi kullanılarak veri raster formatına çevirildi ve tif dosyası olarak kaydedildi. Kaydedilen görüntü QGIS uygulamasında açılarak eğim hesabı yapıldı. Burada eğim görüntüsünün piksel değerleri 0 ile 1151.84 aralığında olmasında rağmen, histogramına bakıldığında eğimin ağırlıklı olduğu değer 2.80 çıktı.

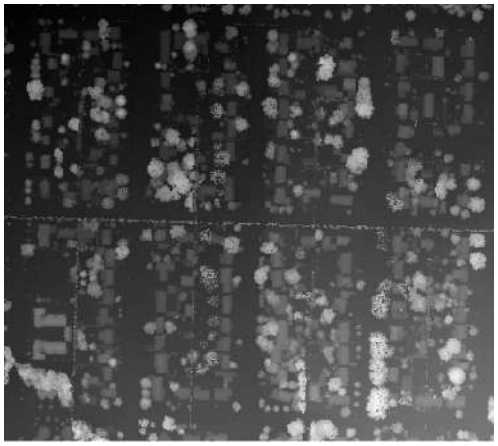


Figure 19: Cloud Compare Uygulaması İle Oluşan Görüntü

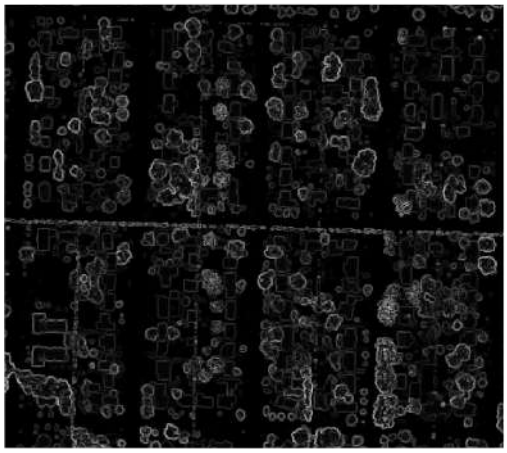


Figure 20: Slope Uygulandığında Oluşan Görüntü

QGIS uygulaması ile hesaplanan eğimde yakın komşuluk uygulanarak oluşturulmuş bir raster veri kullanıldı. Bu sebeple görüntüde kayıplar olabilir. Ayrıca gürültüleri gidermek için de bir uygulama yapılmadı. Cloud Compare uygulamasında ise gürültüler giderilerek raster veriye çevrilmiş görüntü oluşturuldu. Bu sebeplerden dolayı iki görüntü arasında eğim farklılıkları mevcuttur. Cloud Compare uygulaması ile yapılan rastera dönüştürme işlemi daha doğru olduğu için eğim hesaplarında 2.80 değeri kullanıldı.

6. Morfolojik Filtre Uygulama

Cloud Compare üzerinde yapılan raster veri SAGA GIS uygulamasında açıldı. Tools - Grid - Filter - Morphological Filter eklentisi açıldı. Pencere modu kare ve methodu opening seçildi. Pencere boyutuda ilk önce 5, ardından 10 olacak şekilde iki defa uygulandı. Böylelikle morfolojik filtre uygulayarak sayısal arazi modeli elde edildi. Birde ilk görüntüden son görüntü çıkarılarak normalize sayısal yüzey modeli elde edildi.

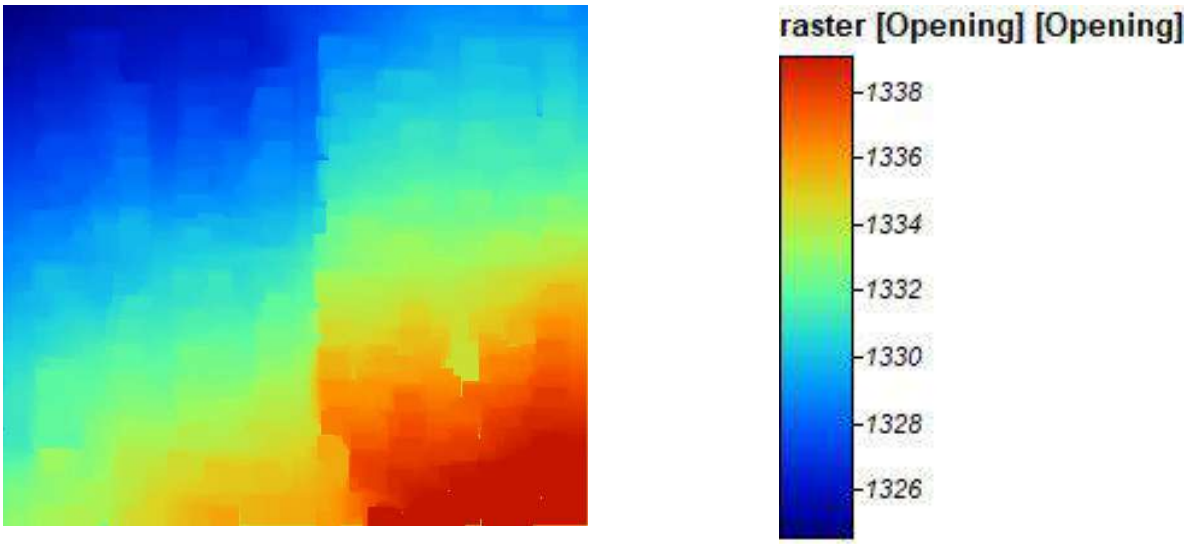


Figure 21: Morfolojik Filtre İle Oluşan Sayısal Arazi Modeli

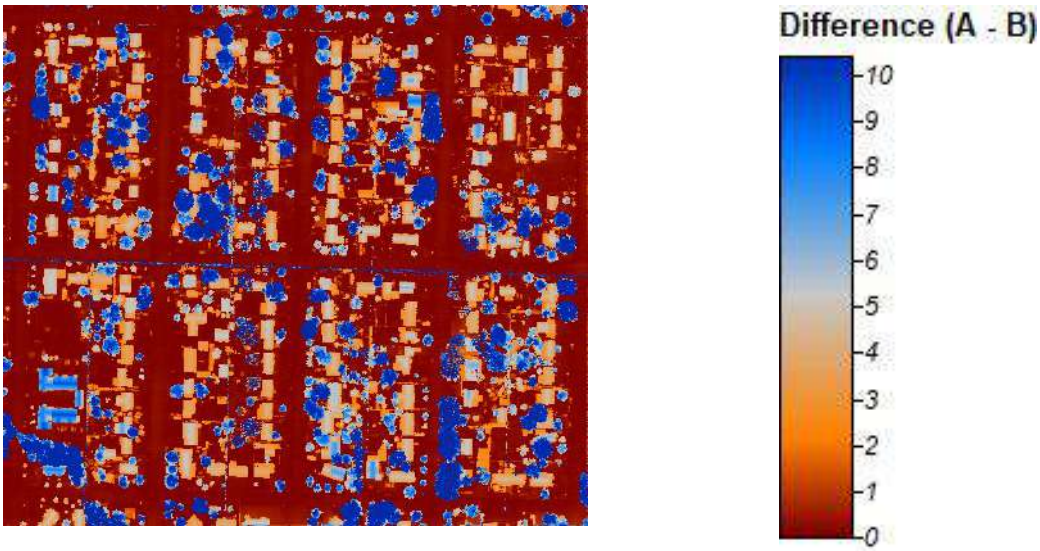


Figure 22: Normalize Sayısal Arazi Modeli

ALDPAT uygulaması açıldı. Tools - WorkSheet kısmından algoritmalar seçildi. Buradan nokta bulutu ve yapılacak filtreleme seçildi. Burada Morph Filter uygulandı. Add New Jobs'a tıklandığında filtrelemede kullanılacak veriler girildi. Filtreyi uygulamak için eğim değeri 2.8 olarak ve seri değeri 10 olarak değiştirildi. Ardından Run Jobs denildiğinde filtreleme yapıldı. Yaptığı filtreleme sonucunda nokta veri oluştu. Nokta veri QGIS programında açıldı. Algoritma sonucunda binaların olduğu alanlar veriden silindi.

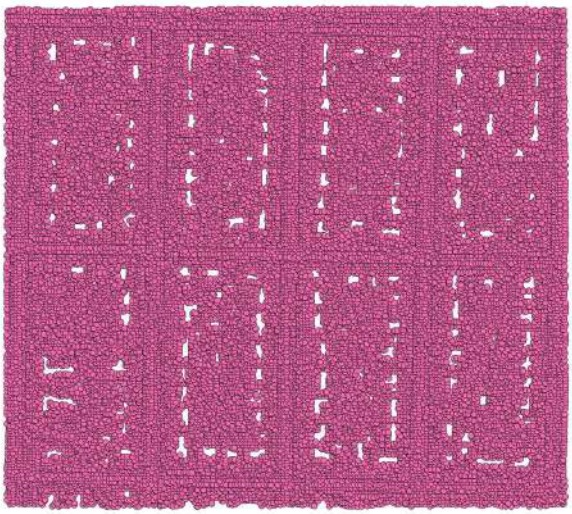


Figure 23: ALDPAT Uygulamasında Morph Filter Uygulanmış Veri

QGIS üzerinde açılan nokta veriyi, raster veriye dönüştürmek için Nearest Neighbor uygulandı. Ardından ALDPAT uygulamasında uygulanan filtre ile SAGA GIS uygulamasında uygulanan filtrenin farkı alındı. Burada farkın piksel değerleri -35.96 ile 4.34998 arasındadır.

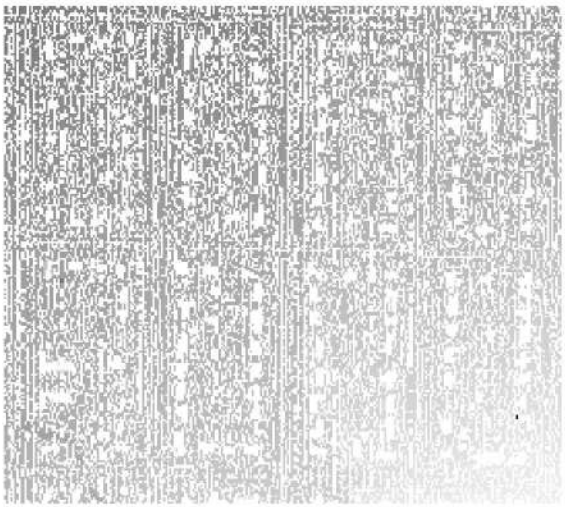


Figure 24: Nearest Neighbor Uygulanmış Hali

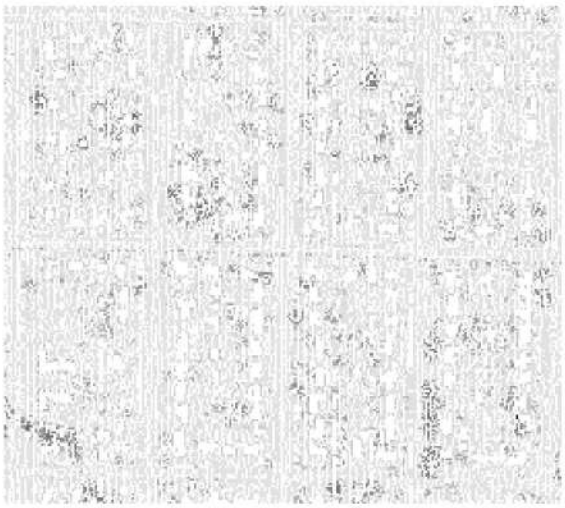


Figure 25: İki Farklı Şekilde Filtrelemenin Farkı

7. Adaptive TIN Filter

ALDPAT uygulamasında Tools - WorkSheet kısmı açıldı. Nokta bulutu ve Adaptive TIN Filter seçildi. Add New Jobs'a tıklandığında filtrelemede kullanılacak veriler girildi. Burada verilerde hiç bir değişiklik yapılmadı. Ardından Run Jobs denildiğinde filtreleme yapıldı. Yaptığı filtreleme sonucunda nokta veri oluştu. Nokta veri QGIS programında açıldı. Açılan nokta veri Nearest Neighbor uygulanarak interpole edildi. Böylelikle noktaların büyüklüğünden dolayı belli olmayan cisimlerde interpolasyon edildiğinde belirgin hale gelmektedir.

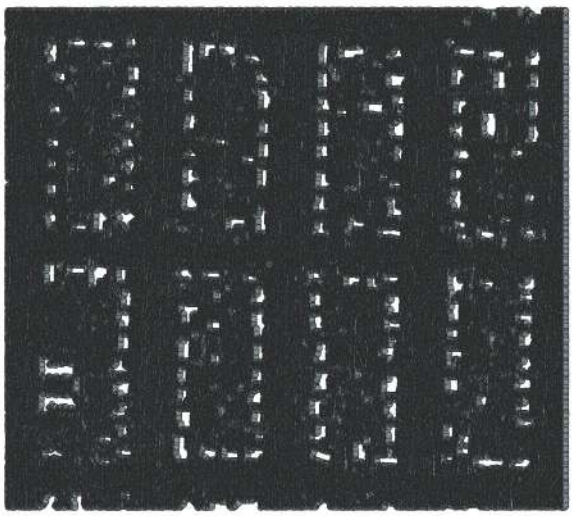


Figure 26: Adaptive TIN Filter Görünümü

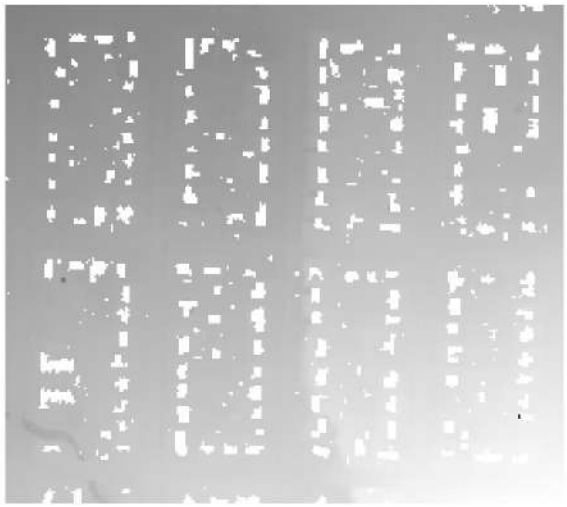


Figure 27: Adaptive TIN Filter'a Nearest Neighbor Uygulanması

8. Sel Analizi

Ham nokta verisine Nearest Neighbor uygulanmış hali SAGA GIS uygulamasına eklendi. Tools - Terrain Analysis - Hydrology - Flow Accumulation (Flow Tracing) kısmından kümülatif akış yönü bulundu. Bu görüntüyü tif formatında export ederek QGIS uygulamasında açıldı.

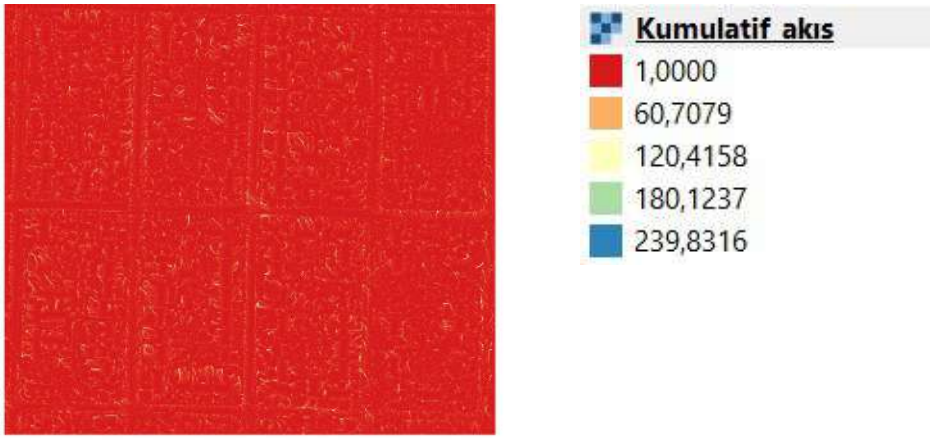


Figure 28: Sel Analizi

Adaptive TIN Filter uygulanan veri QGIS üzerinde açıldı. İki verininde koordinat sistemlerine dikkat edilerek üst üste getirildi. Böylelikle bina alanlarının olduğu bölgelerde sarı değerler de az olduğu için binaları su basma ihtimali çok düşüktür. Genel olarak mavi bölge olmadığı için bölgede sel olma ihtimali azdır.

Figure 29: Caption