

# BIL 467 Görüntü İşleme

## Ödev #1

Kağan Hamzaçebi

151101064

06/09/2021

### 1. Genel Açıklama

Ödev klasörü içerisinde 3 farklı script dosyası, kullanılan görüntülerin bulunduğu images isimli bir klasör ve outputs isimli çıktıların bulunduğu bir klasör bulunmaktadır.

Ödev 3 farklı script dosyası içerisine dağıtılarak yapılmıştır. Bu bölümler;

- gray\_interpolation.m (4 numaralı soru)
- rgb\_interpolation.m (5 numaralı soru)
- distance\_transform.m (6 numaralı soru)

**Not:** Ek olarak 3 numaralı soru (thresholder) distance\_transform bölümü içerisinde yer almaktadır.

**Not2:** Sonuçların birbiriyle karşılaştırılabilir olması için outputs içerisinde yer alan interpolation çıktıları(fig uzantılı matlab figure) ratio: 2 kullanılarak elde edilmiştir. (input: 512x512 output: 1024x1024)

#### 1.1. Kullanım

Daha öncesinde Matlab tecrübem olmadığından ve belirli bir yapı istenmediğinden ötürü yapılması istenilen işlemleri fonksiyonlar olarak tanımladım.

Çalıştırılması istenilen fonksiyonu çağırıp diğer fonksiyonları comment out yaparak testleri gerçekleştirebilirsiniz. Varsayılan(default) olarak nearest neighbor metodu belirlenmiştir.

## 2. Grayscale Interpolation Modülü (4 numaralı Soru)

Bu script dosyasının içerisinde grayscale image'lara uygulanabilen input olarak **image** ve **ratio (scale ratio)** alan 3 farklı interpolation fonksiyonu tanımlanmıştır.

- nearestNeighborGRAY (Satır: 12 – 32 arası)
- bilinearGRAY (Satır 35 – 85 arası)
- bicubicGRAY (Satır 88 – 186 arası)

### 2.1. Sonuç

Nearest Neighbor Interpolation (Grayscale) fonksiyonunda beklenen sonuç elde edilmiştir ve bir problem ile karşılaşılmamıştır.

Bilinear ve Bicubic Interpolation (Grayscale) fonksiyonlarında Nearest Neighbor tekniğine göre daha yumuşak(smooth) geçişler olduğu gözlemlenmiştir. Ancak bu iki fonksiyonda image'da soluk kare şekiller belirmiştir. Pixellerin tek tek intensity değerlerine bakıldığında bir problem gözükmemekte ancak figure üzerinde bu problem görülebilmektedir. Ek olarak image daha büyük bir çözünürlüğe büyütülmek istendiğinde örn: 4096x4096 belirtilen sorun yok olmaktadır.

## 3. RGB Interpolation Modülü (5 numaralı Soru)

Bu script dosyasının içerisinde rgb image'lara uygulanabilen input olarak **image** ve **ratio (scale ratio)** alan 3 farklı interpolation fonksiyonu tanımlanmıştır.

- nearestNeighborRGB (Satır: 12 – 38 arası)
- bilinearRGB (Satır 41 – 96 arası)
- bicubicGRAY (Satır 99 – 201 arası)

### 3.1. Sonuç

Bu bölümde karşılaşılan sonuçlar bir önceki modül olan Grayscale Interpolation kısmı ile aynıdır. Nearest Neighbor başarılı bir şekilde çalışmaktadır. Bilinear ve Bicubic fonksiyonlarında da istenilen yumuşuk geçiş sağlanmıştır ancak sebebini belirleyemediğim karelenmeler mevcuttur.

#### 4. Distance Transform Modülü (6 numaralı Soru)

Bu script dosyasının içerisinde binary image'lara uygulanabilen distance transform fonksiyonları yer almaktadır.

- thresholder (Satır 19 – 31 arası)
- boundry\_detection (Satır 35 – 85 arası)
- adjacency\_deteciton (Satır 64 – 116 arası)

##### 4.1. Sonuç

Thresholder fonksiyonu ödev dökümanında istenilen 3. sorunun cevabıdır. Input olarak image ve threshold kabul eder. Belirtilen threshold değerinden yüksek olan pixel intensity değerlerini 255, düşük olanları ise 0 yapmaktadır. Problemsiz bir şekilde çalışmaktadır ve istenilen görevi yerine getirebilmektedir.

Boundry Detection fonksiyonu input olarak bir binary image kabul etmektedir ve bu image içerisinde yer alan boundry pixelleri tespit ederek input ile aynı büyüklükte, boundry pixellerin 1 intensity'e sahip olduğu bir maske döndürmektedir.. Test amaçlı kullanılan image'larda sorunsuz bir şekilde boundry pixelleri tespit etmiştir.

Adjacency Detection fonksiyonu input olarak Boundry Detection'dan gelen maskeyi kabul etmektedir(image) ve bu image içerisinde bulunan 1 intensity'e sahip pixellerden başlayarak pixel'in 8 komşusuna(8-adjacency) o pixelin sahip olduğu intensity değerinin 1 üstünü vererek devam eder. Bu fonksiyonda karşılaştığım problem iterasyonun hangi koşulda tamamlanacağını(terminate) belirleyemediğimden ötürü 1500 iterasyon sonrası durdurmayı tercih ettim. Bir önceki aşamada belirlenmiş olan köşe noktaları(boundry pixels) kullanarak cisimleri işaretlemeyi başardım ancak tam olarak başarılı çalıştığı söylenemez.

Son olarak Adjacency Detection sonrası 12. ve 14. satırlar arasında normalize işlemi gerçekleştirilmiştir.

## 5. Karşılaşılan Problemler – Zorluklar

İlk olarak matlab’i ilk defa kullandığım için syntax v.b. aşamalarda sorunlar yaşadım. Matlab’in kendi dökümantasyonuna bakarak ve forumlardaki cevaplar ile karşılaştığım sorunları çözdüm.

Yazmış olduğum kısımlarda index hatalarını çözerken problemler yaşadım. Out of index hatalarını aşağıdaki kod parçasına benzer şekilde çözdüm.

```
x(x < 1) = 1;  
x(x > cols - 1) = cols - 1;  
y(y < 1) = 1;  
y(y > rows - 1) = rows - 1;
```

Interpolation modüllerinde bilinear ve bicubic interpolation fonksiyonlarında uzaklık ağırlıklı(distance weighted) hesaplamaları yaparken zorlandım.

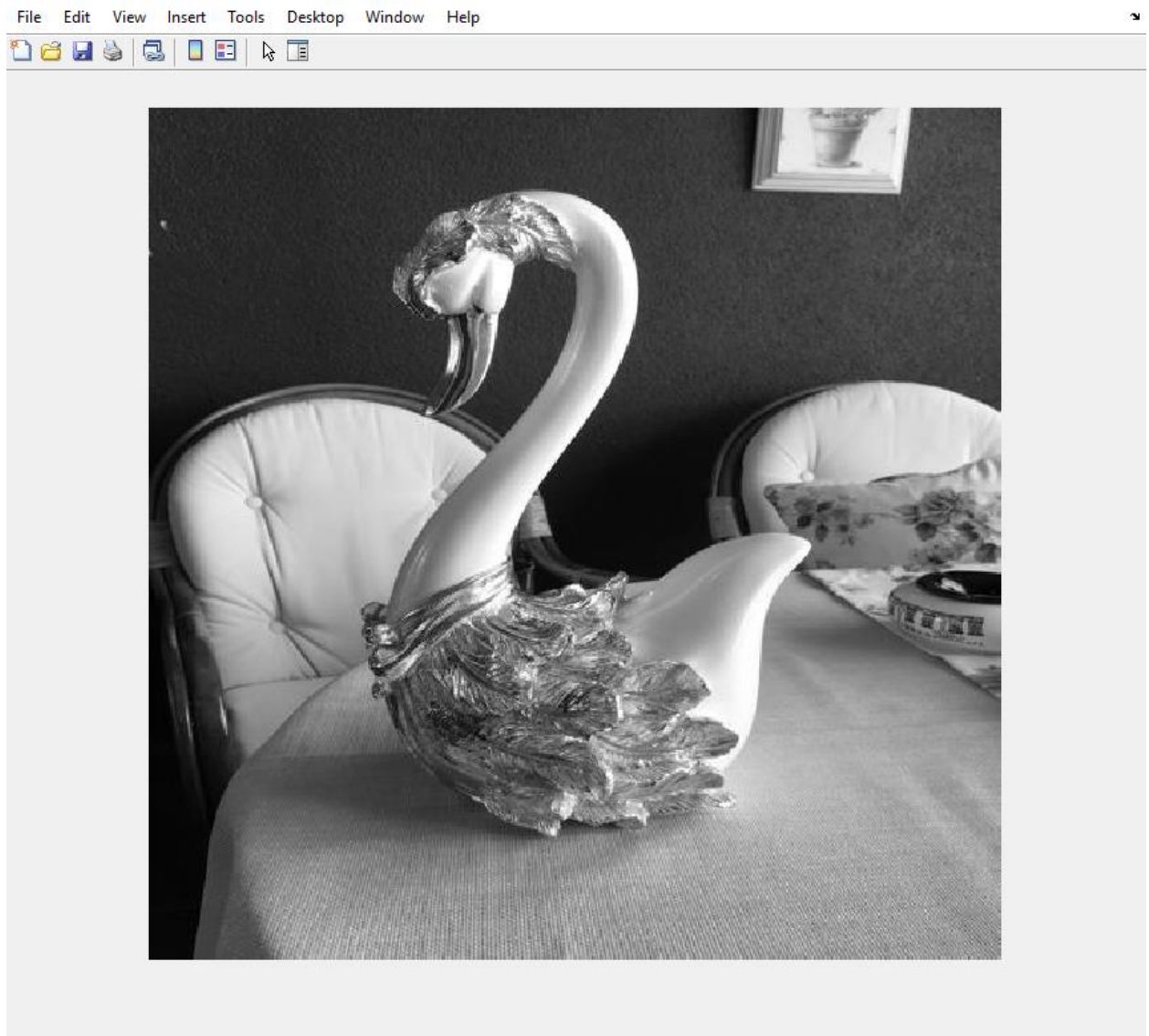
Bilinear ve Bicubic Interpolation fonksiyonlarında sebebini belirleyemediğim bir karelenme sorunu yaşadım.

Distance Transform modülünde iterasyon sonlandırmayı nasıl yapacağımı bulamadım. İlk olarak en sol üst, sol alt, sağ üst ve sağ alt pixellerin değerlerinin 0 olmadığını kontrol ettim ama istediğim sonucu vermedi. Bazı durumlarda döngü sonlanmadı. Bunun sonucunda bu yöntemden vazgeçtim ve iterasyon sayısını tutarak 1500 iterasyon sonucu döngüyü sonlandırdım. İdeal çözüm yolu olmasa da en iyi sonucu bu şekilde aldım.

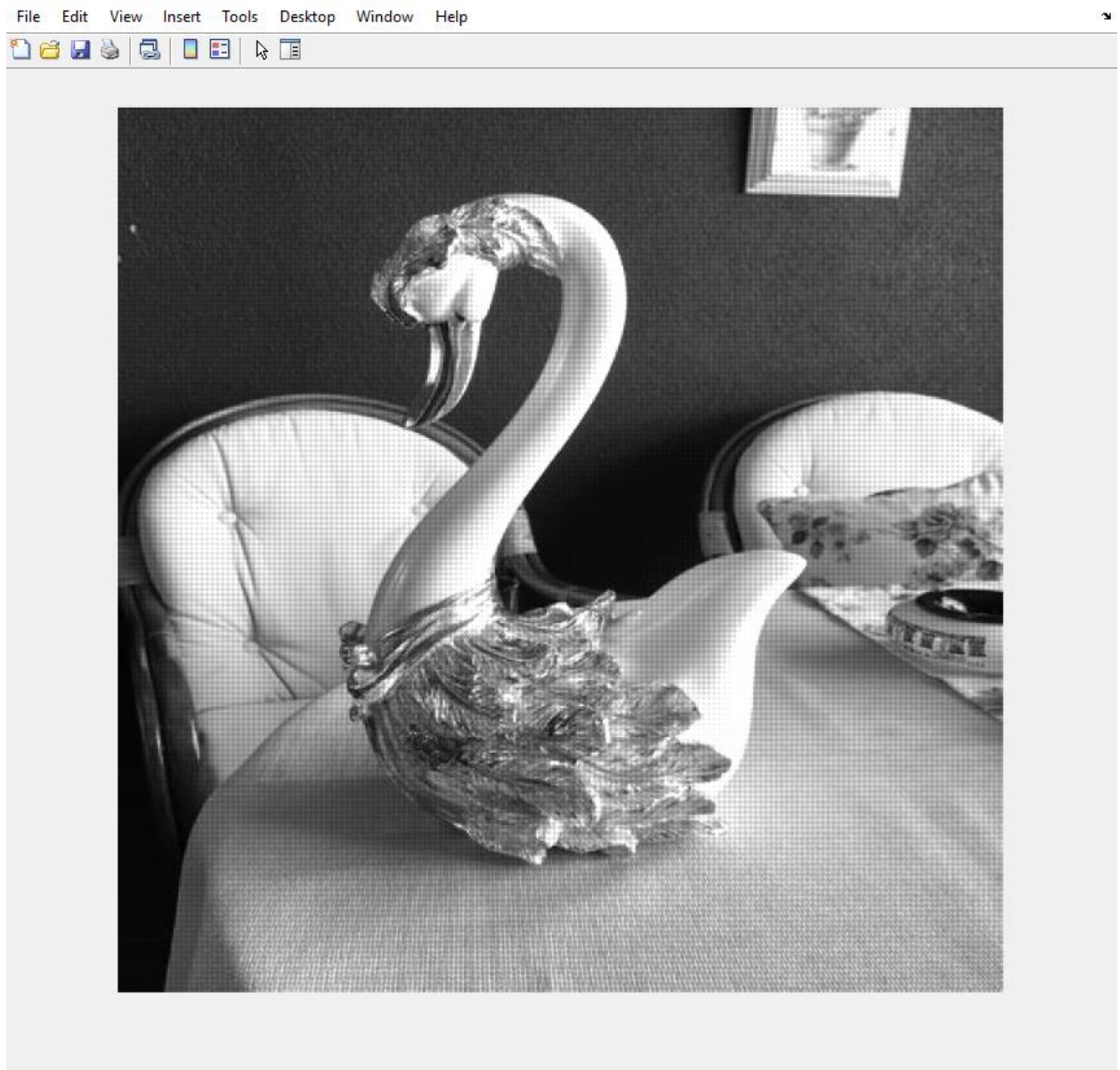
## 6. Girdi ve Çıktılar



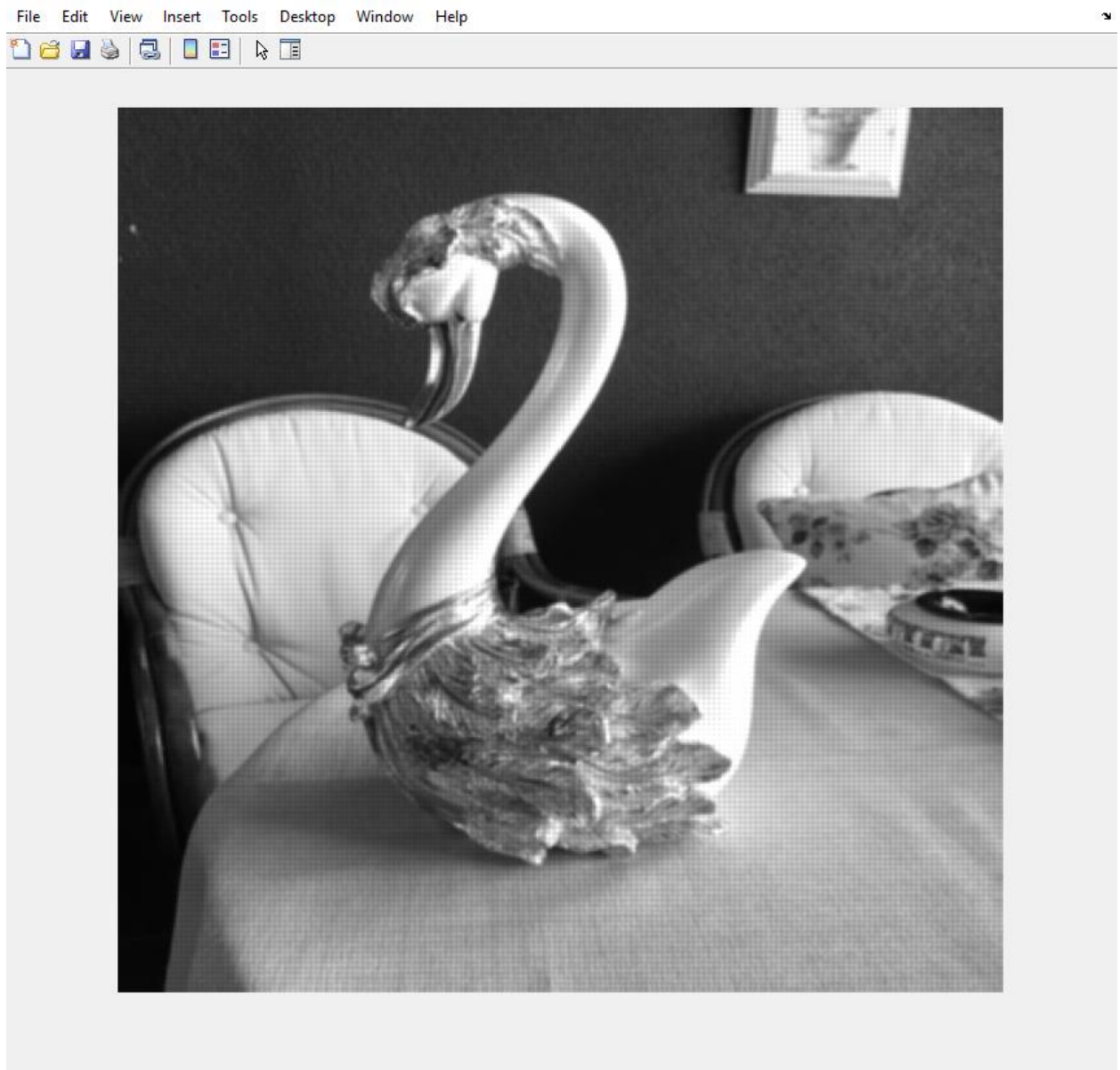
Original Grayscale Interpolation Grayscale (512x512)



Nearest Neighbor Interpolation Grayscale (ratio 2)



Bilinear Interpolation Grayscale (ratio 2)

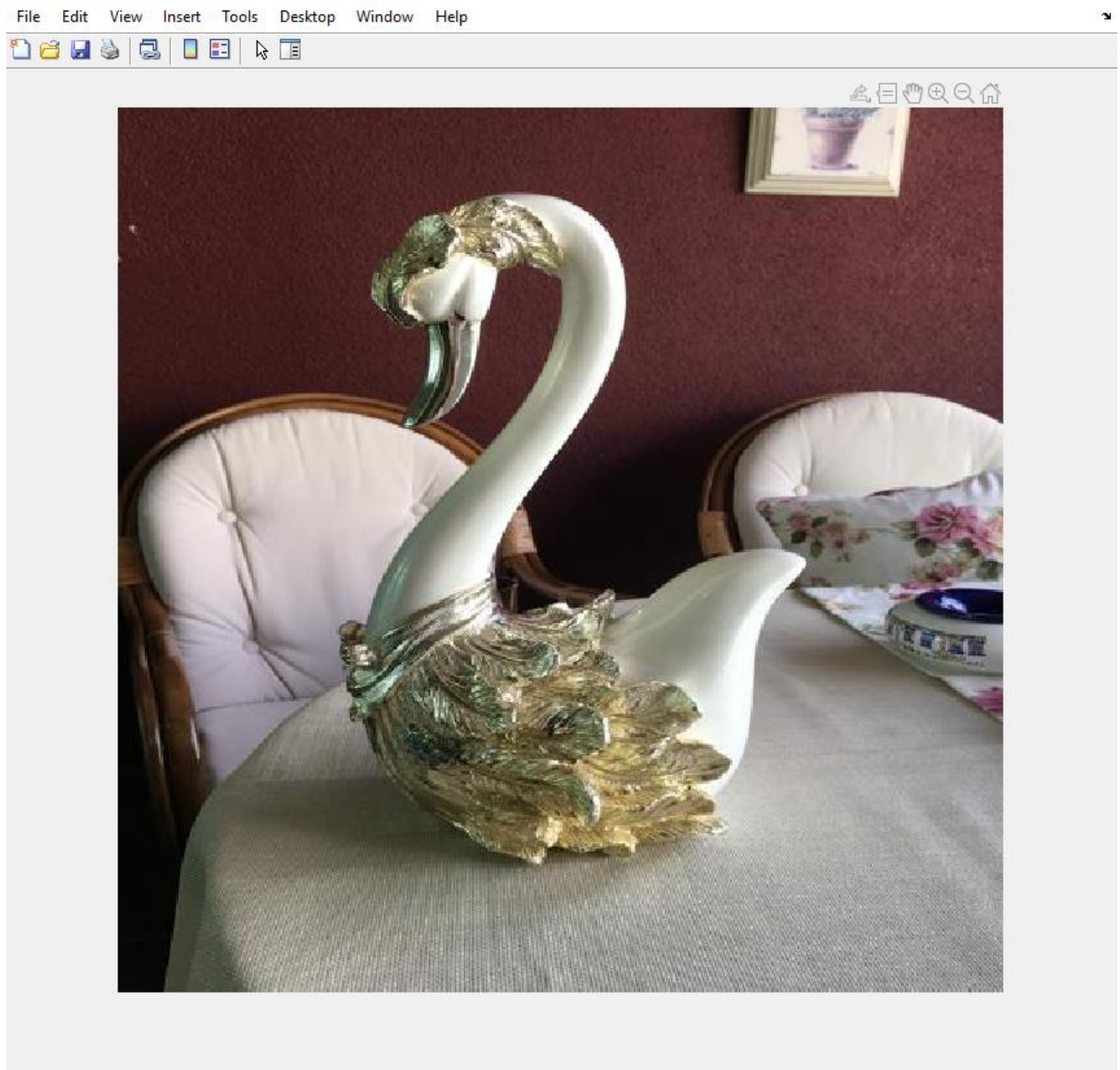


Bicubic Interpolation Grayscale (ratio 2)

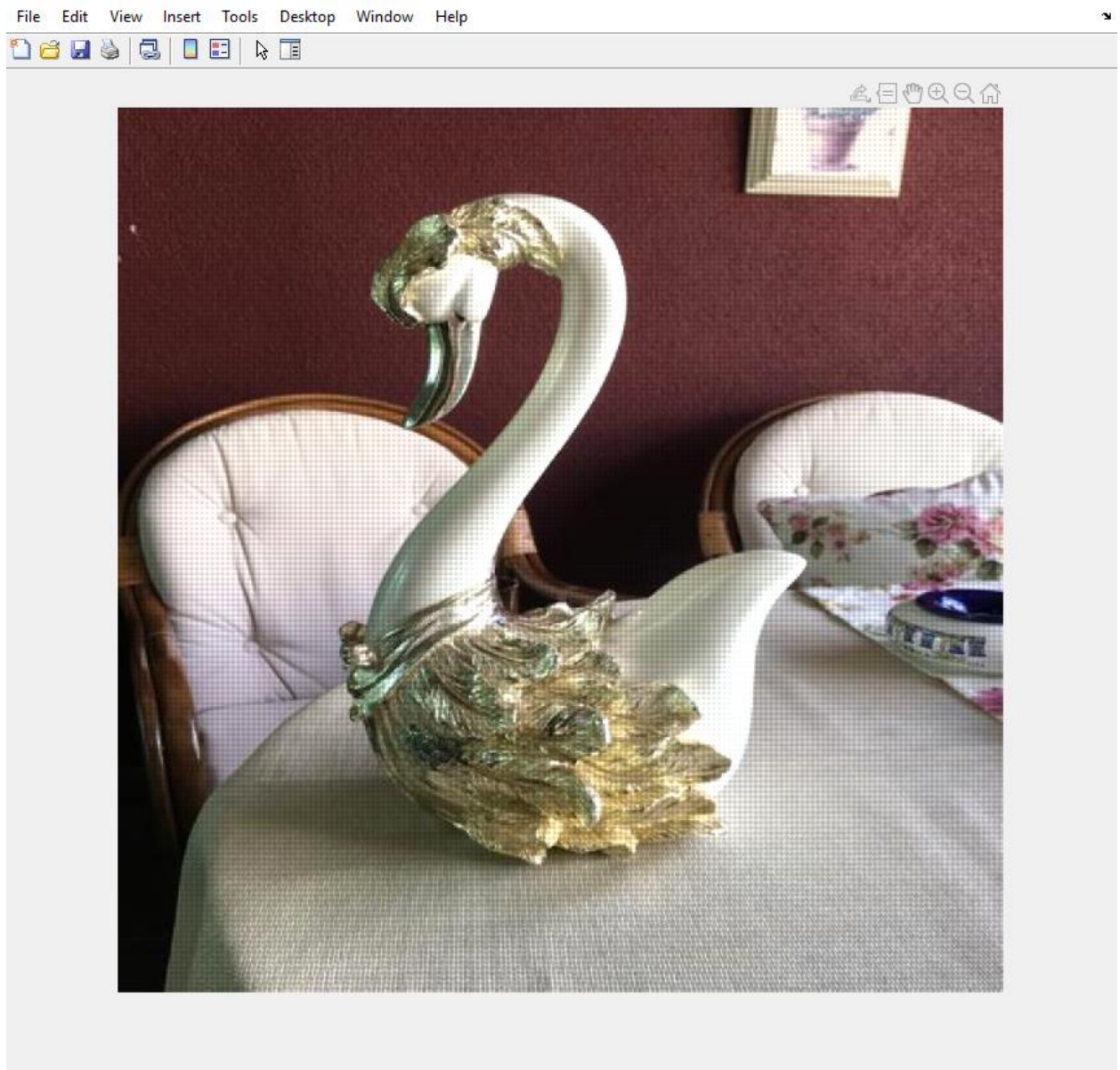




Original RGB Interpolation Grayscale (512x512)

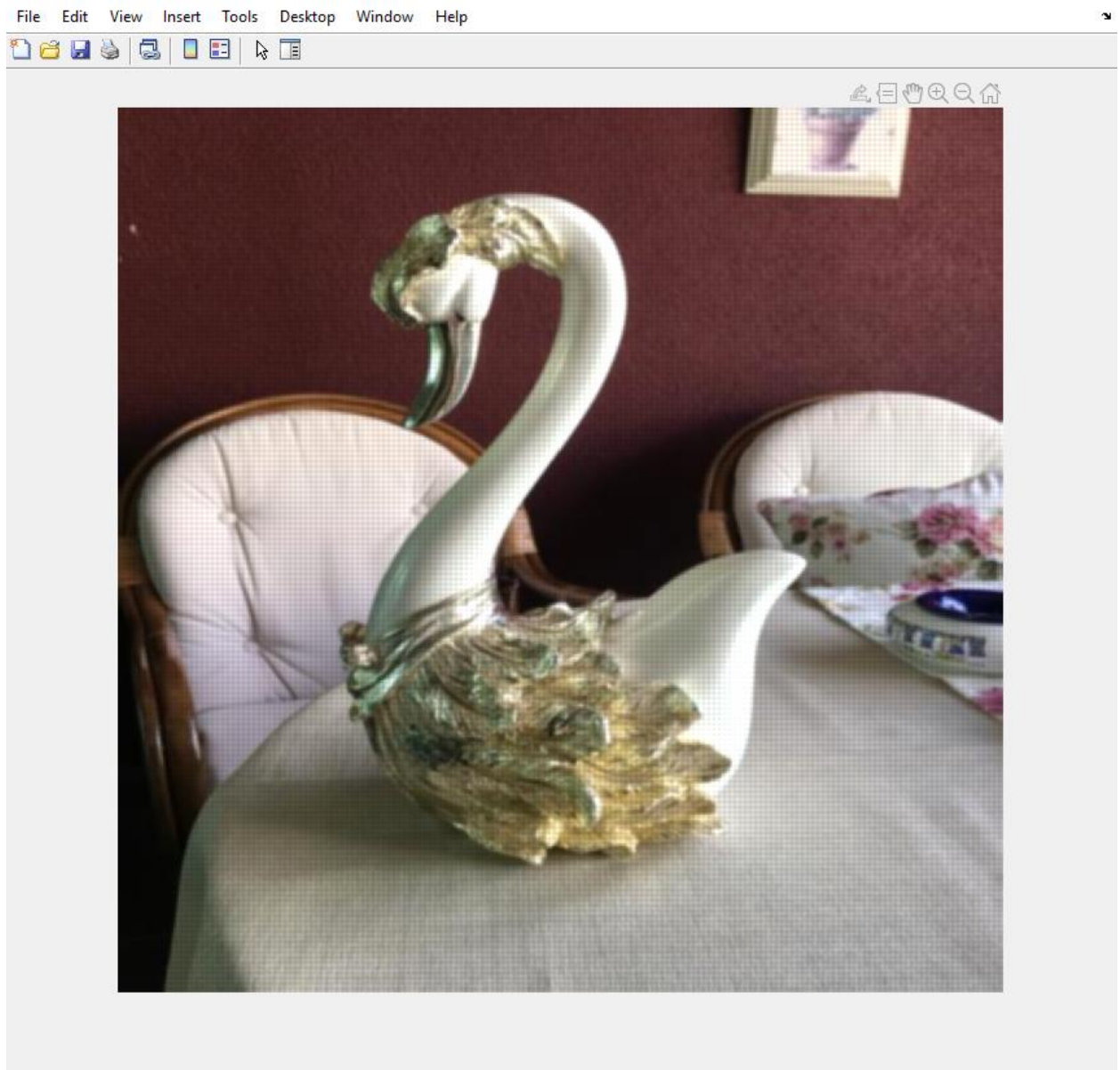


Nearest Neighbor Interpolation RGB (ratio 2)



Bilinear Interpolation RGB (ratio 2)





Bicubic Interpolation RGB (ratio 2)

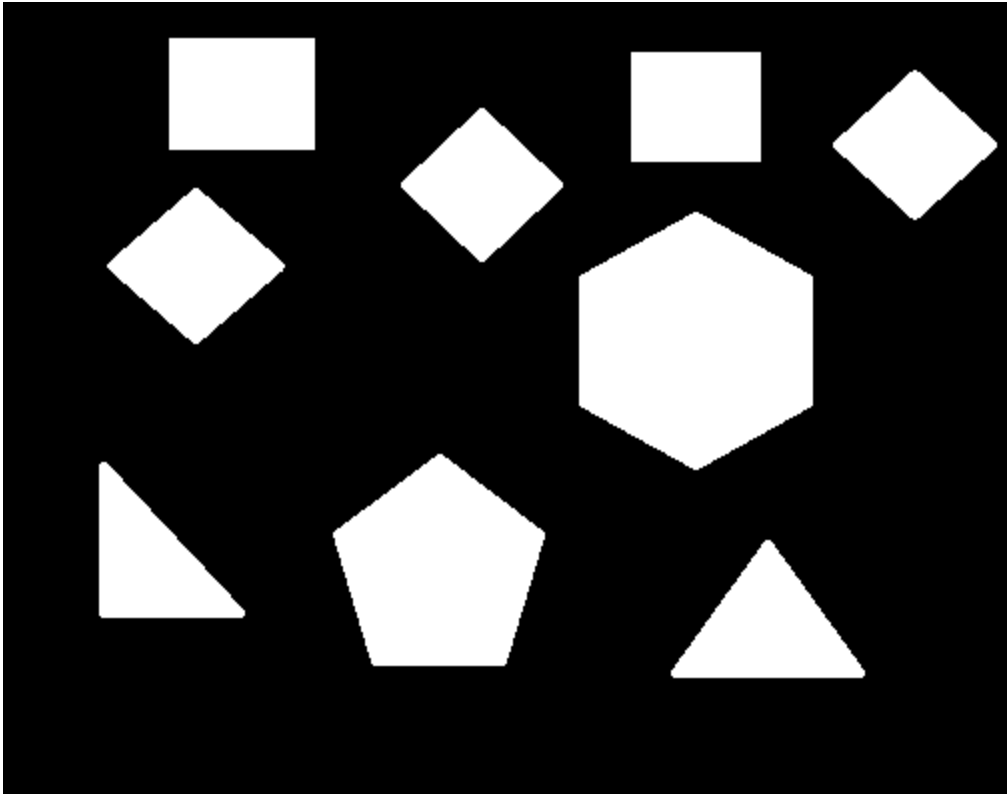


Image 1 – Original Image Distance Transform

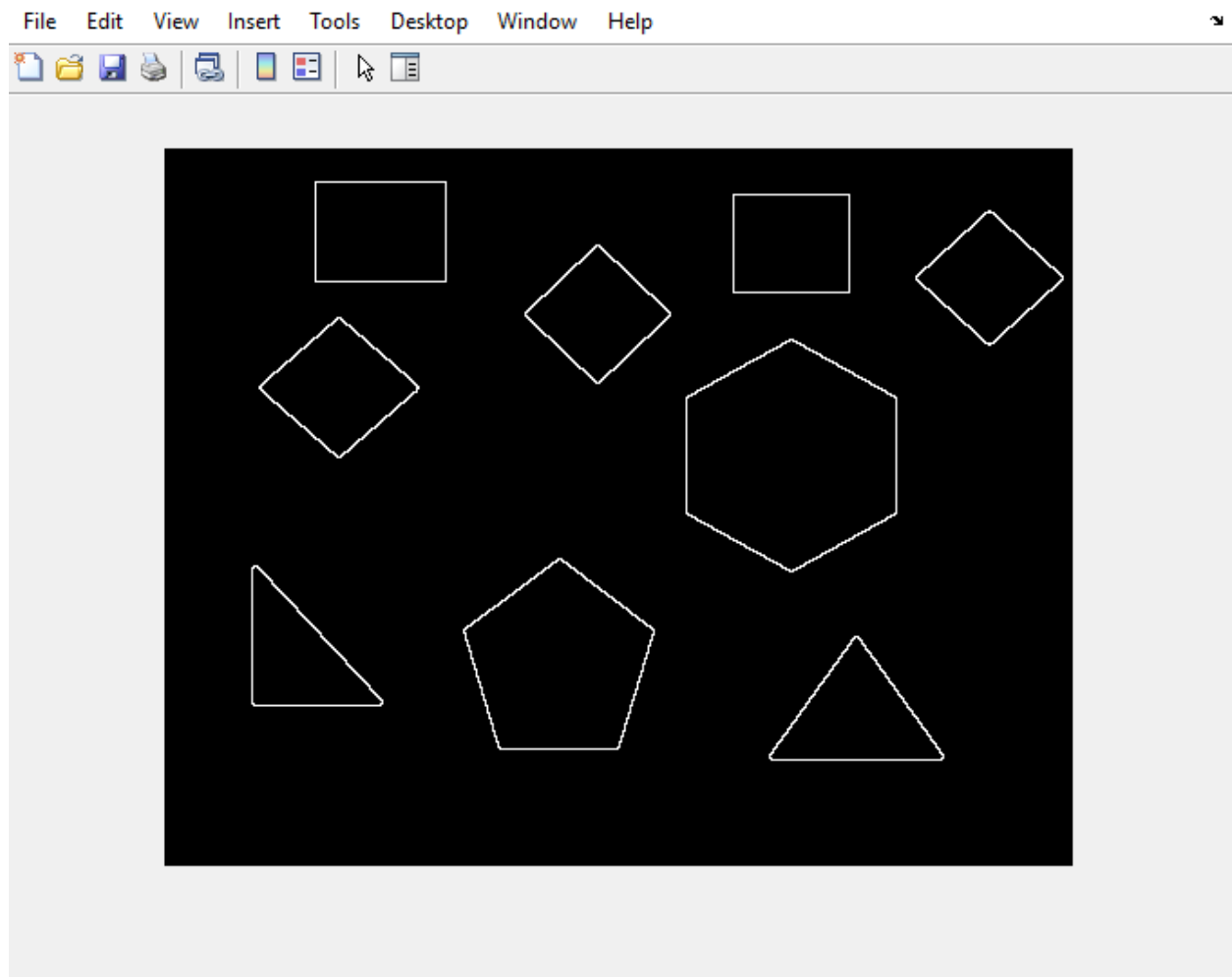


Image 1 – Boundry Pixel Detection

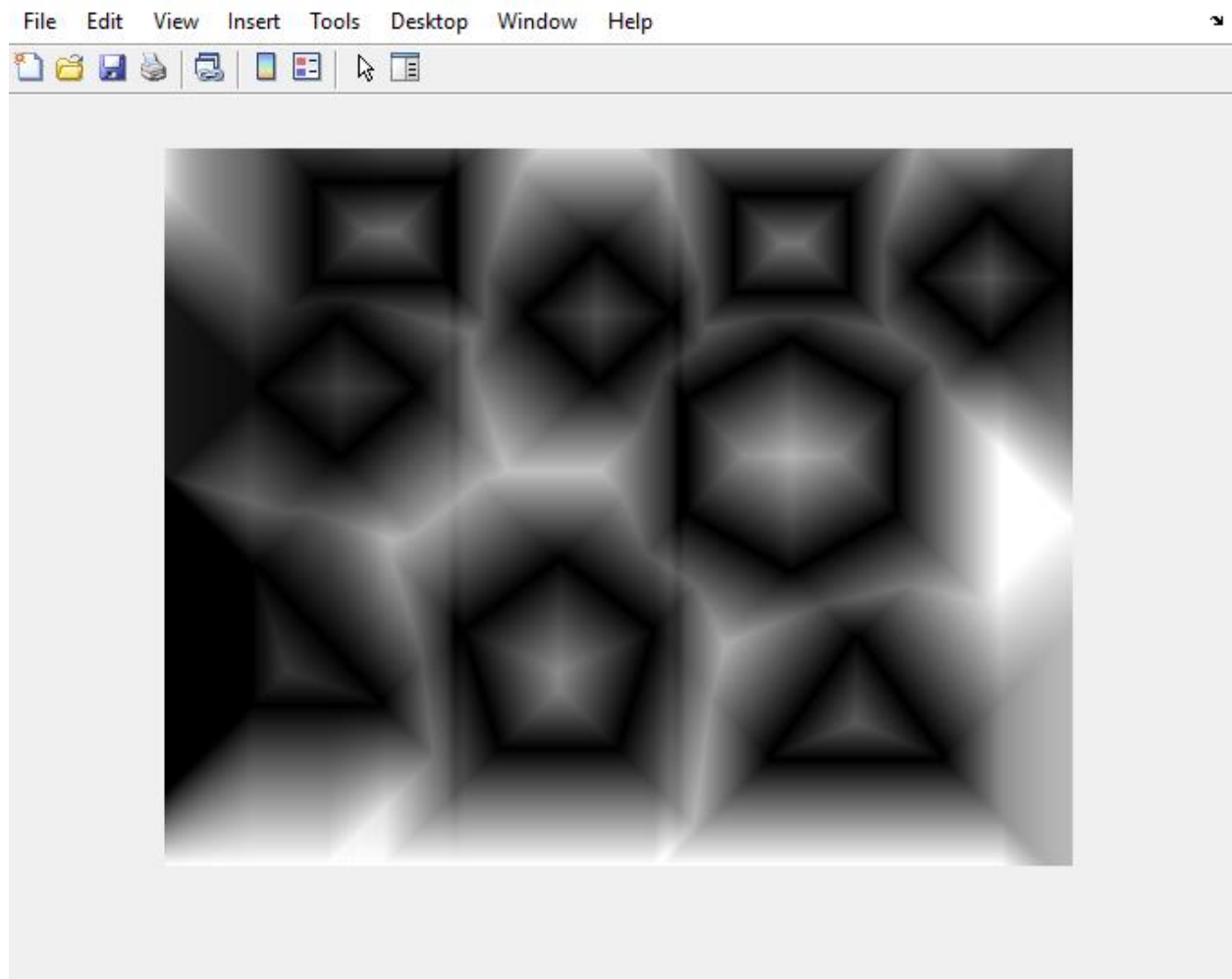


Image 1 – After Distance Transform

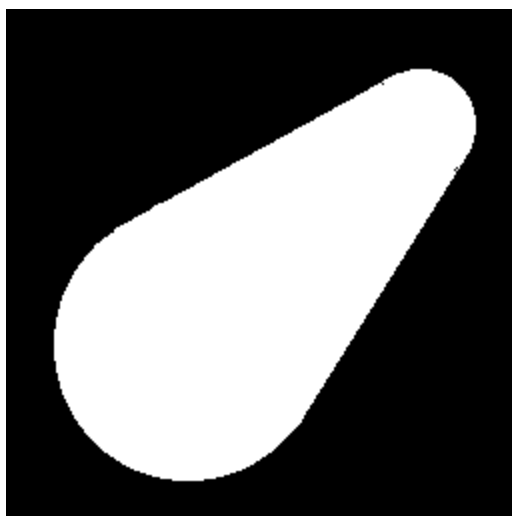


Image 2 – Original Image Distance Transform

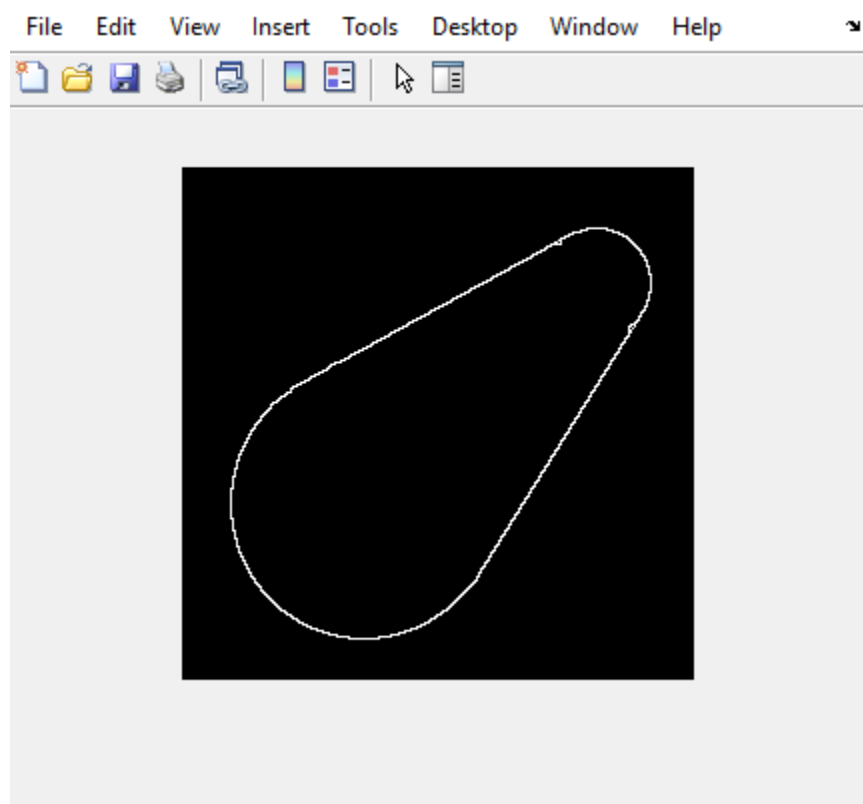


Image 2 – Boundry Pixel Detection



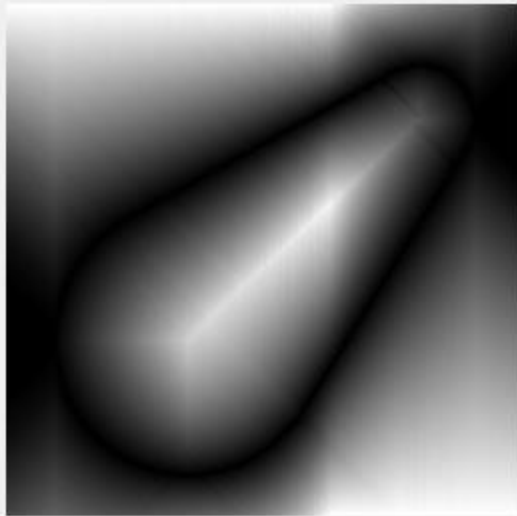


Image 2 – After Distance Transform

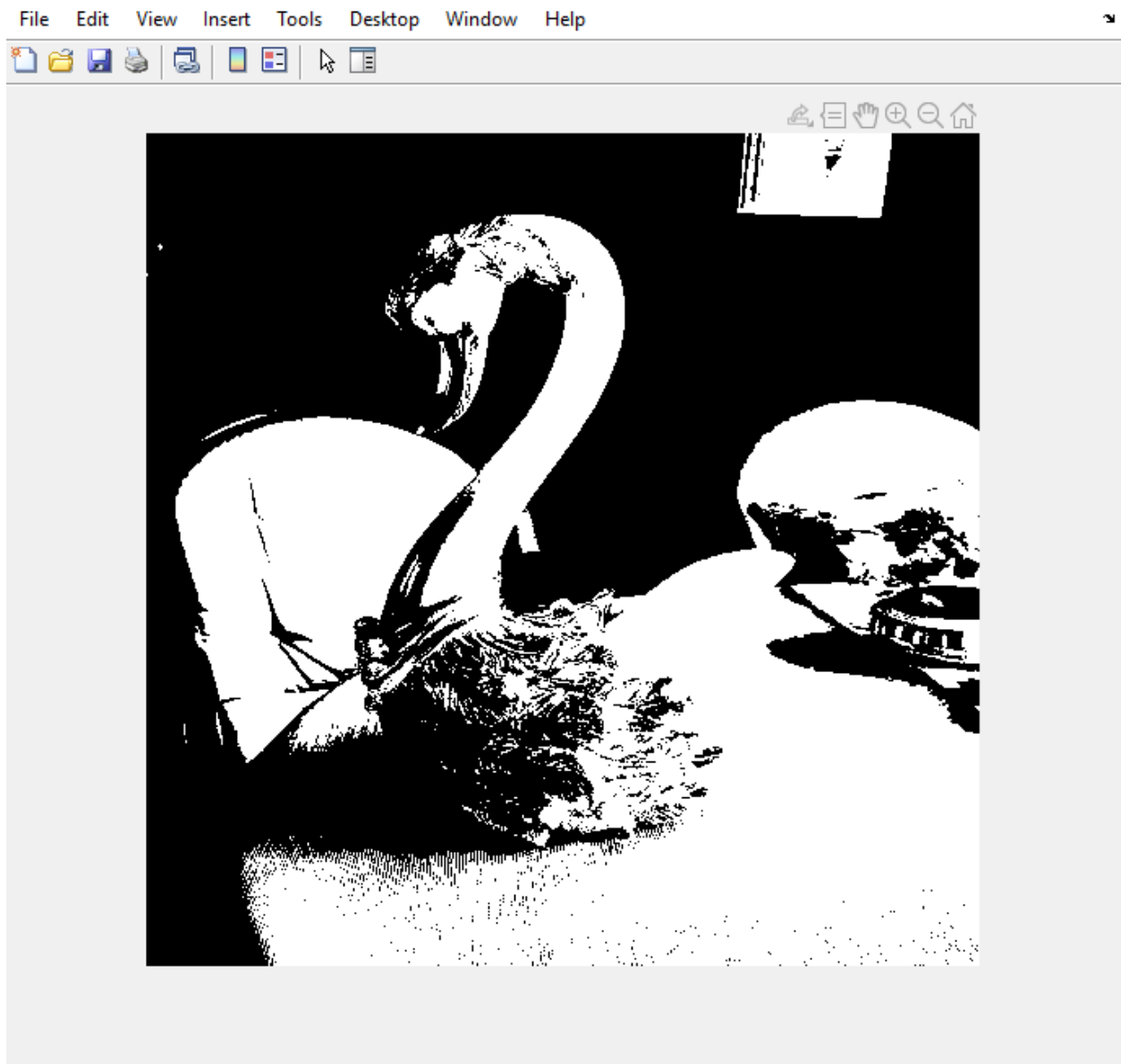


Image 3 – Thresholder Used for Distance Transform

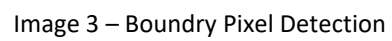


Image 3 – Boundry Pixel Detection

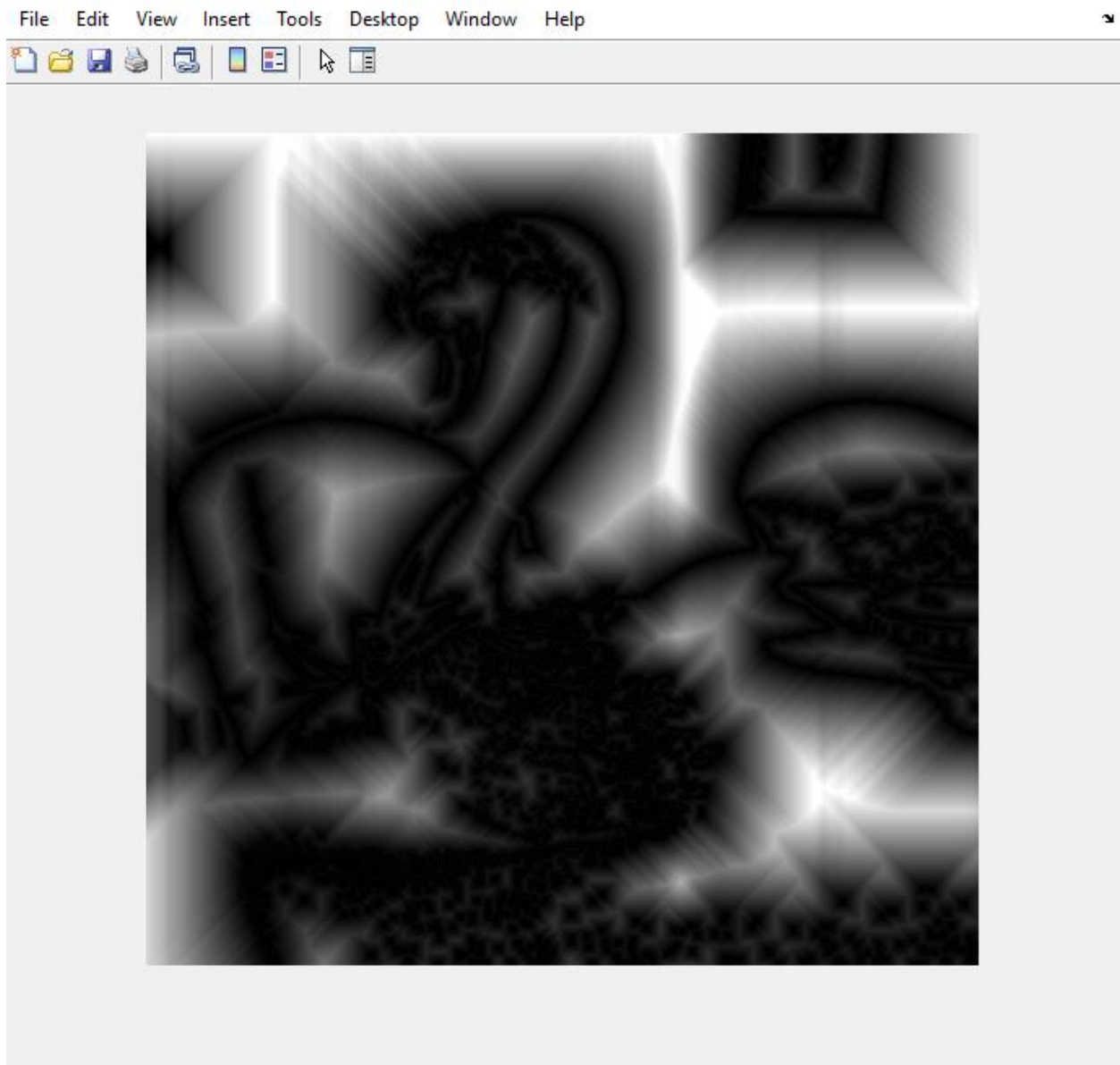


Image 3 – After Distance Transform