

ΓΑΒΡΙΗΛΙΔΗΣ ΣΟΦΟΚΛΗΣ-ΦΙΛΑΡΕΤΟΣ 2014030062

ΧΟΥΡΔΑΚΗΣ ΓΕΩΡΓΙΟΣ 2014030231

## ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΑΝΑΦΟΡΑ 4ΗΣ ΑΣΚΗΣΗΣ

### Για την 1η άσκηση:

Αρχικά διαβάζουμε την εικόνα και δηλώνουμε για Kernel 3x3 έπειτα καλούμε την συνάρτηση Compute\_Median με im την εικόνα και K ως το Kernel

```
im= imread('noisyimg.png');  
figure  
imshow(im)  
k3 = ones(3);  
filtered1 = Compute_Median(im, k3);
```

```
function medi = Compute_Median(image, kernel)  
[r, c] = size(image);  
[n, n] = size(kernel);  
  
siz = 2*floor(n/2);  
I = padarray(image , [siz,siz], 'replicate');  
    for i=1:(r-1+siz)  
        for j=1:(c-1+siz)  
            patch= [];  
            for k=1:n  
                for l=1:n  
                    patch = [patch , I(i+l,j+k)];  
                end  
            end  
            patch = sort(patch);  
            m =median(patch);  
            I(i-1+siz,j-1+siz) = m;  
        end  
    end  
medi = I;  
end
```

Στην συνέχεια εμφανίζουμε την filtered εικόνα με τις παρακάτω εντολές  
figure  
imshow(filtered)



Επειτα κάνουμε για Kernel 5x5

```
k5 = ones(5);
```

```
filtered2 = Compute_Median(im, k5);
```

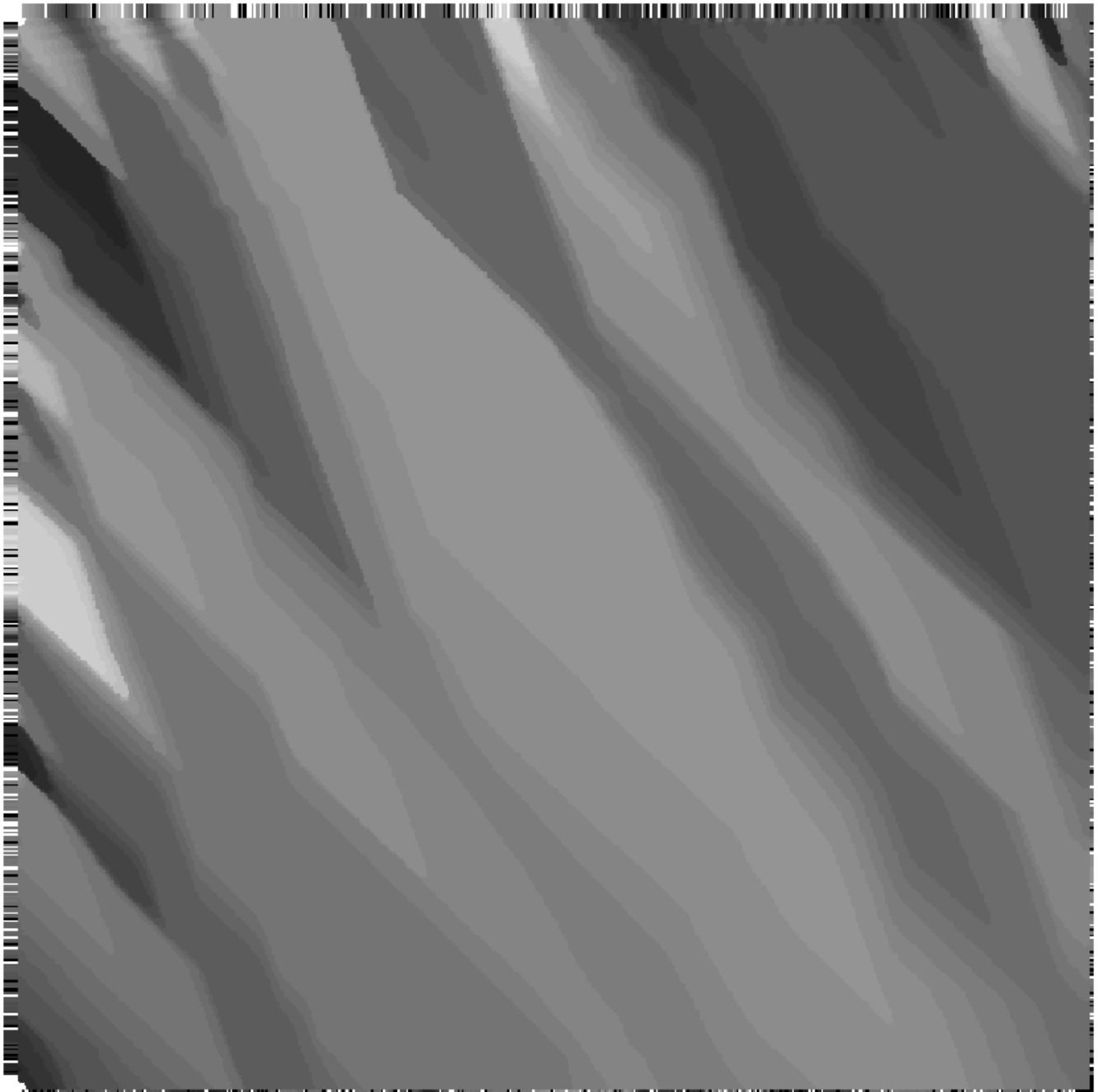
```
figure
```

```
imshow(filtered2)
```



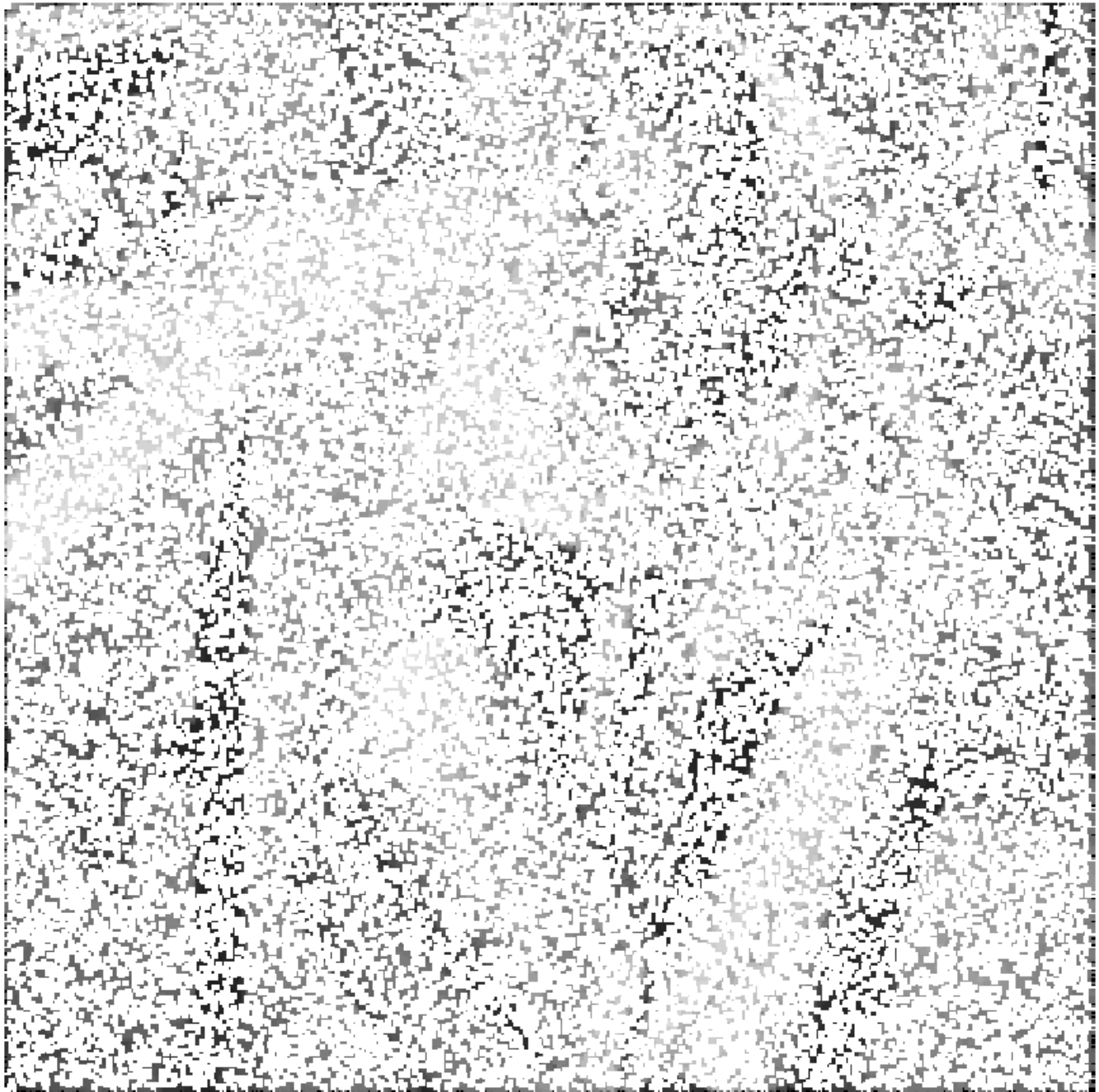
Επειτα κανουμε για Kernel 9x9

```
k9 = ones(9);  
filtered3 = Compute_Median(im, k9);  
figure  
imshow(filtered3)
```



Για την Max με Kernel 3x3:

```
max1 = Compute_Max(im, k3);  
figure  
imshow(max1)
```



με function Compute\_max:

```
function maxi = Compute_Max(image, kernel)  
[r, c] = size(image);  
[n, n] = size(kernel);  
  
siz = 2*floor(n/2);
```

```

I = padarray(image ,[siz,siz],'replicate');
    for i=1:(r-1+siz)
        for j=1:(c-1+siz)
            patch= [];
            for k=1:n
                for l=1:n
                    patch = [patch , I(i+k,j+l)];
                end
            end
            patch = sort(patch);
            m =patch(9);
            I(i-1+siz,j-1+siz) = m;
        end
    end
maxi = I;
end

```

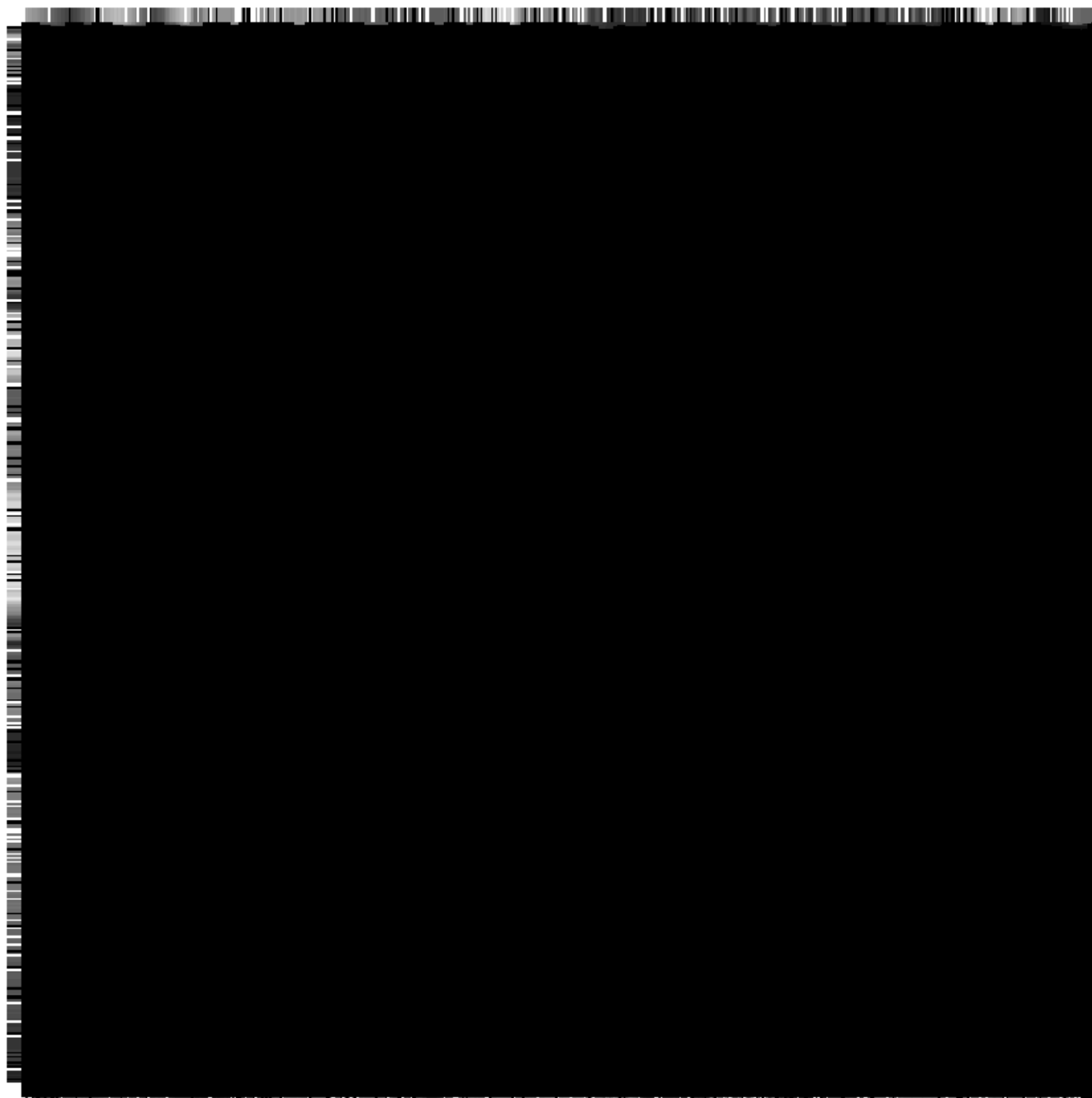
Για την Max με Kernel 5x5:

```
max2= Compute_Max(im, k5);  
figure  
imshow(max2)
```



Για την Max με Kernel 9x9:

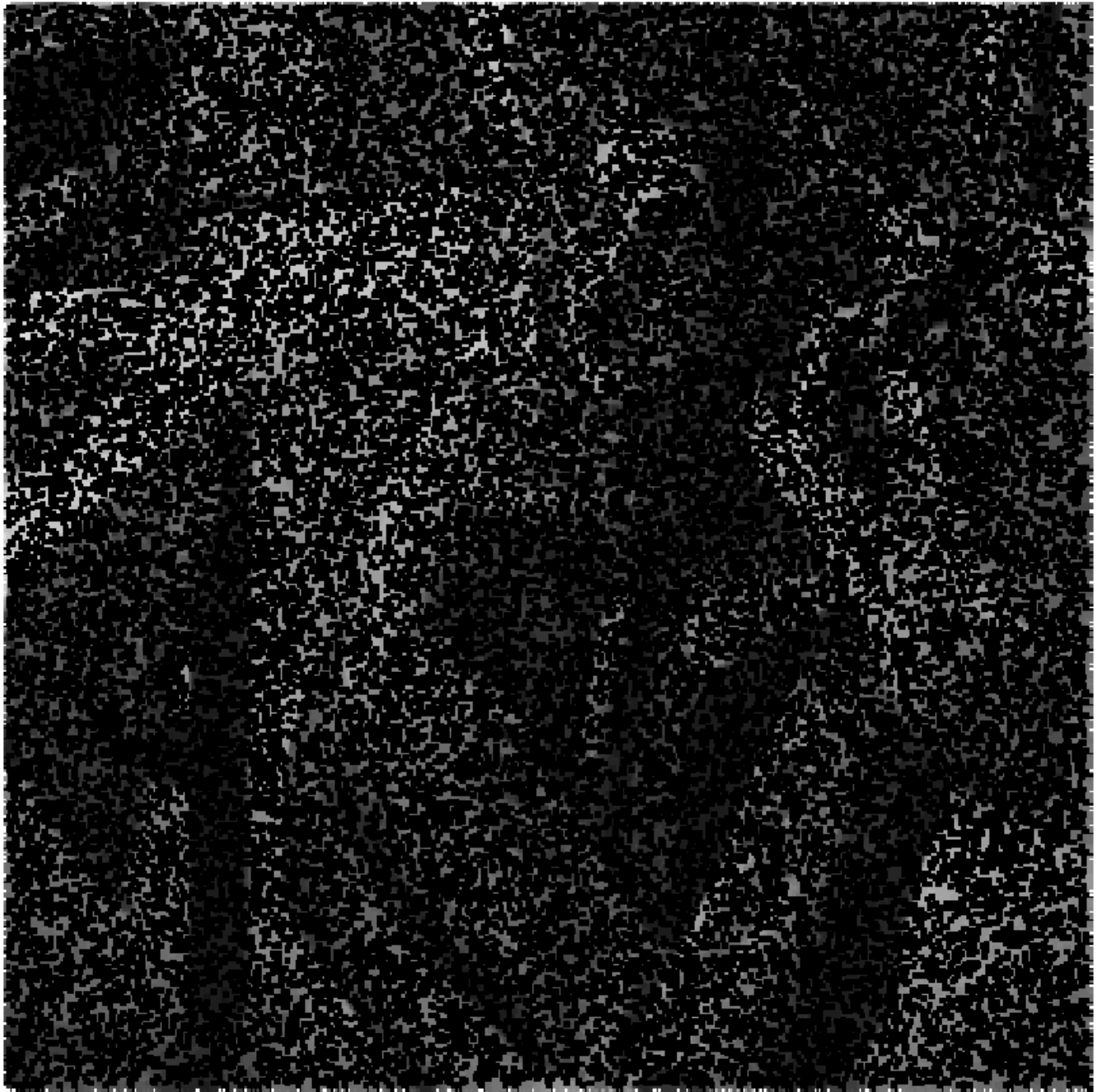
```
max3 = Compute_Max(im, k9);  
figure  
imshow(max3)
```





Για την Min με Kernel 3x3:

```
min1 = Compute_Min(im, k3);  
figure  
imshow(min1)
```



με function Compute\_min:

```
function mini = Compute_Min(image, kernel)  
[r, c] = size(image);  
[n, n] = size(kernel);  
  
siz = 2*floor(n/2);  
I = padarray(image, [siz, siz], 'replicate');  
for i=1:(r-1+siz)
```

```

        for j=1:(c-1+siz)
            patch= [];
            for k=1:n
                for l=1:n
                    patch = [patch , I(i+k,j+l)];
                end
            end
            patch = sort(patch);
            m =patch(1);
            I(i-1+siz,j-1+siz) = m;
        end
    end
    mini = I;
end

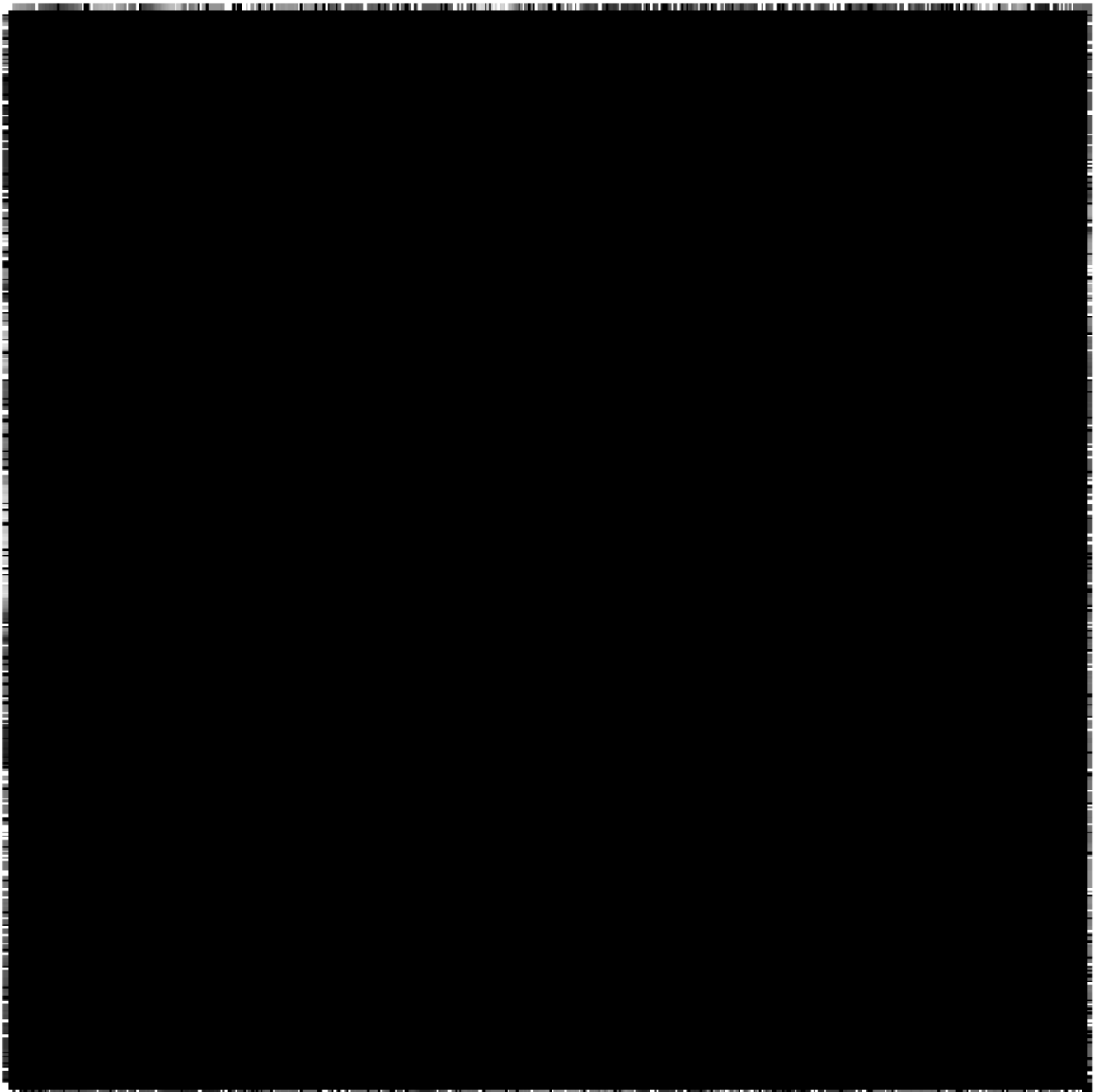
```

Για την Min με Kernel 5x5:

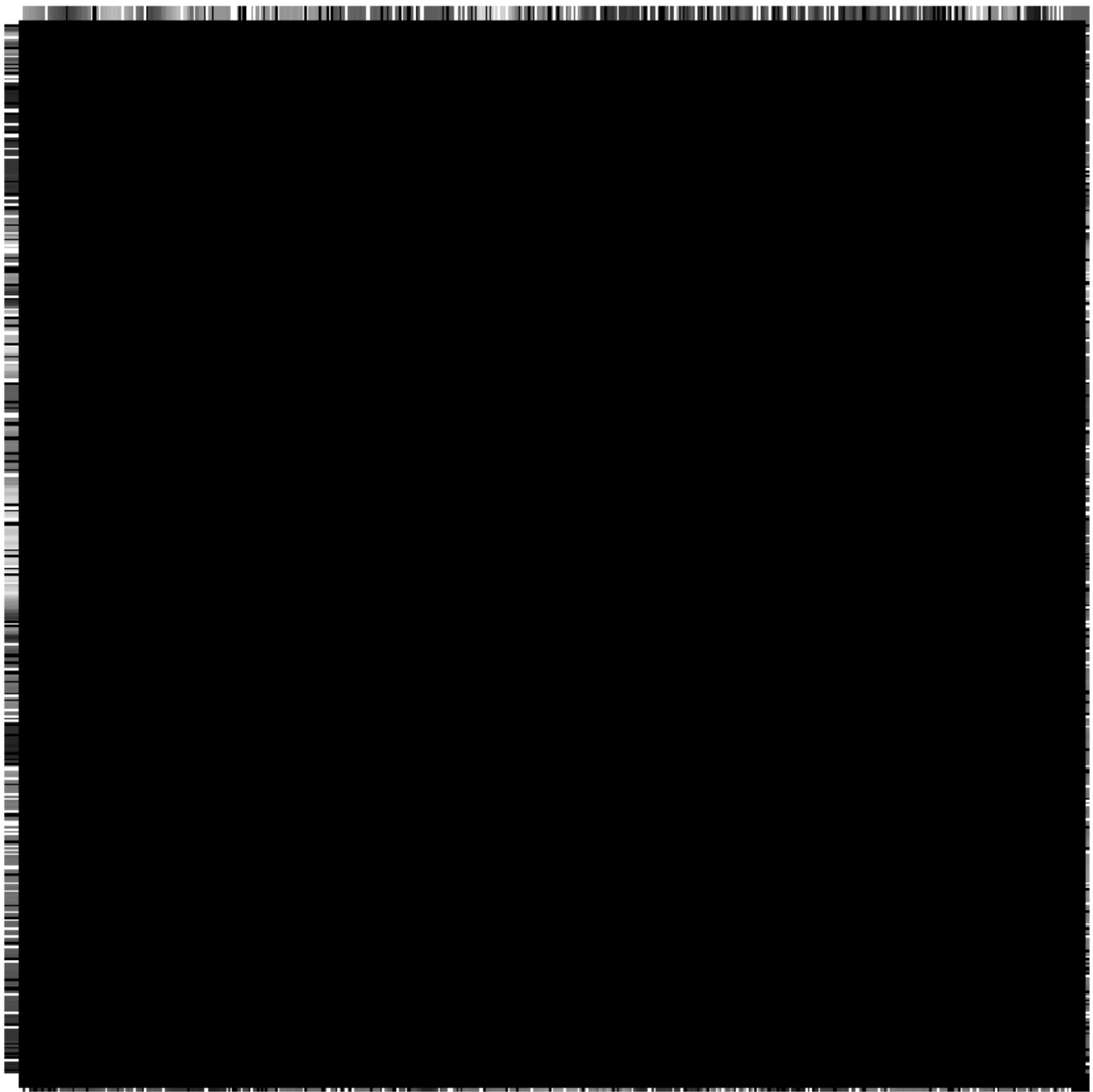
```

min2 = Compute_Min(im, k5);
figure
imshow(min2)

```



Για την Min με Kernel 9x9:  
`min3 = Compute_Min(im, k9);`  
`figure`  
`imshow(min3)`



Έπειτα τρέχουμε ακριβώς τον ίδιο κώδικα με 1 αλλαγή αντι για `im= imread('noisyimg.png');` βάζουμε `im= imread('noisyimg2.png');`

Median για Kernel 3x3



Median για Kernel 5x5



Median για Kernel 9x9



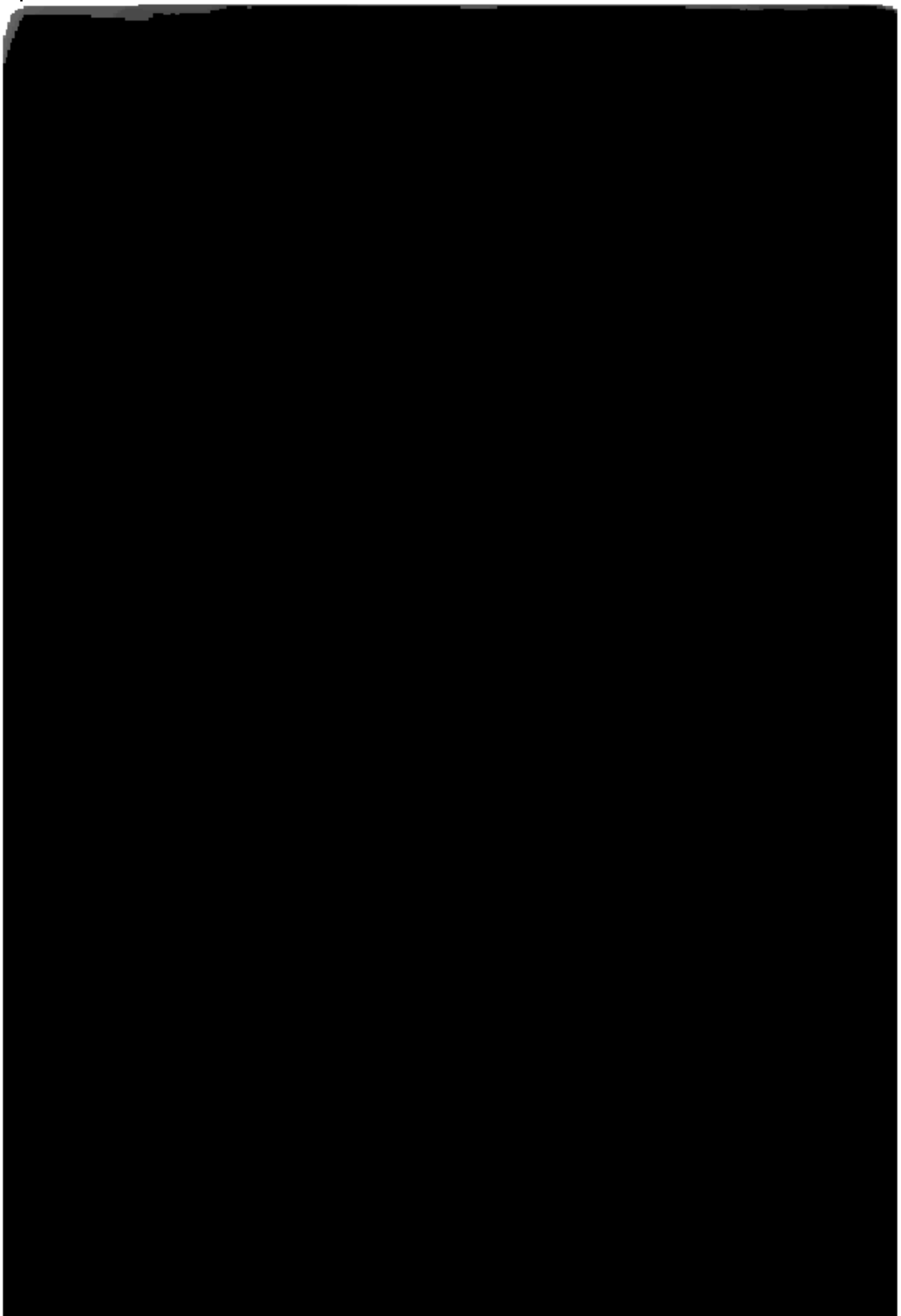
Max για Kernel 3x3



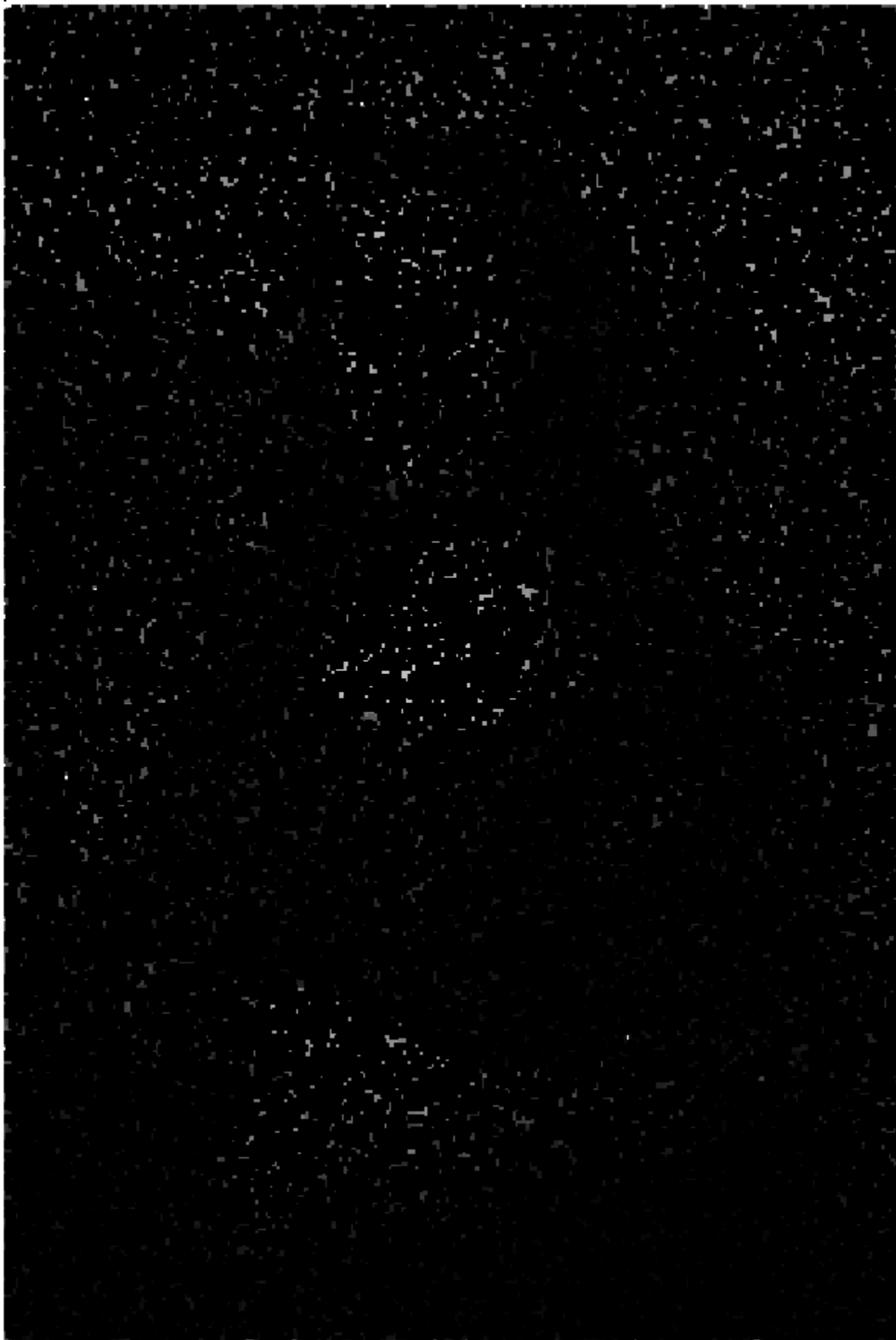
Max για Kernel 5x5



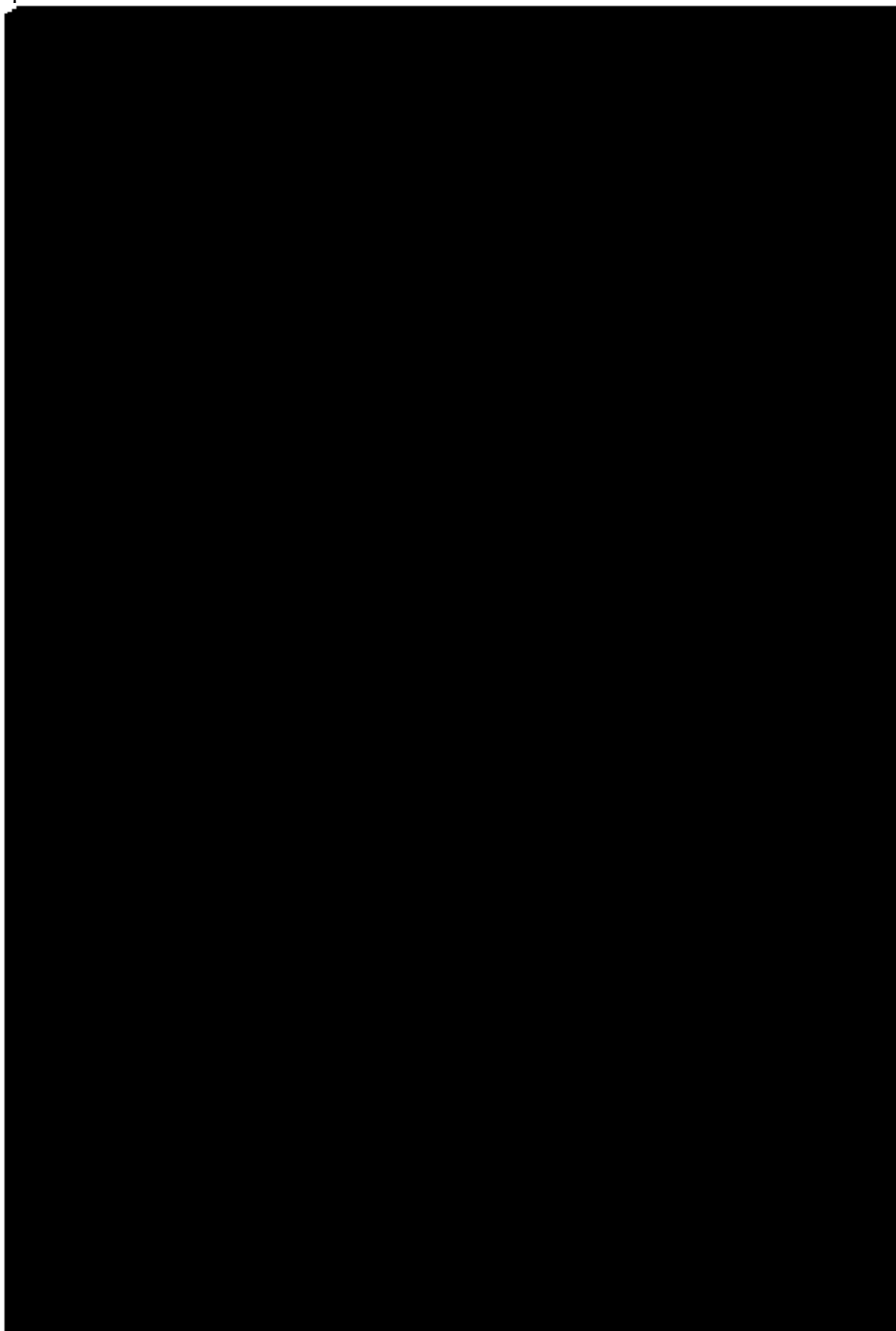
Max για Kernel 9x9



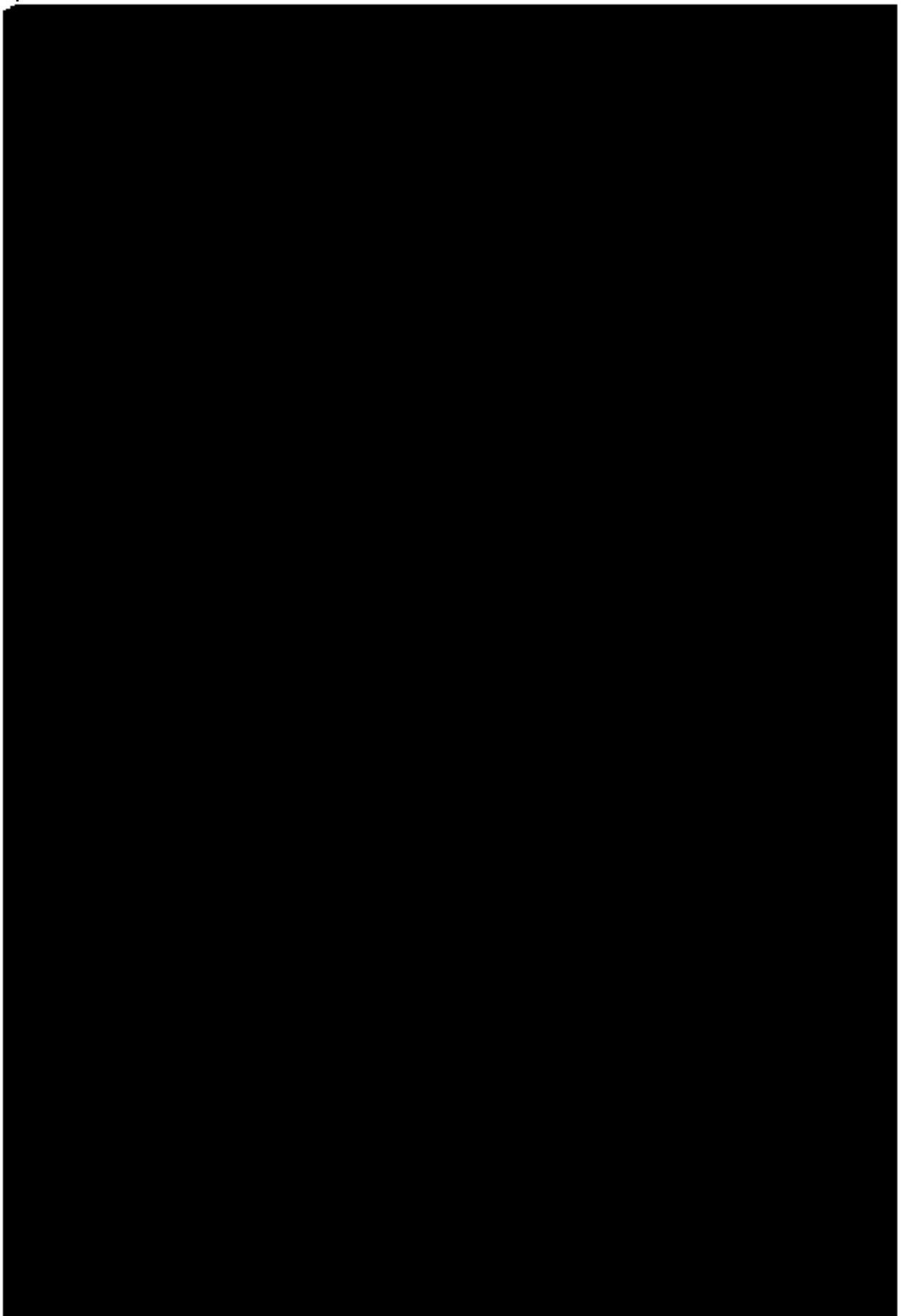
Min για Kernel 3x3



Min για Kernel 5x5



Min για Kernel 9x9



Ο κώδικας ίσως δεν είναι 100% αποτελεσματικός γιατί στο Kernel 9x9 θα έπρεπε σε median filter να μην είναι τόσο blurry. Στα max θα έπρεπε να είναι σχεδόν άσπρα αλλά δεν είναι όπως στο max 3x3. Τα min θα έπρεπε να είναι τελείως μαύρα