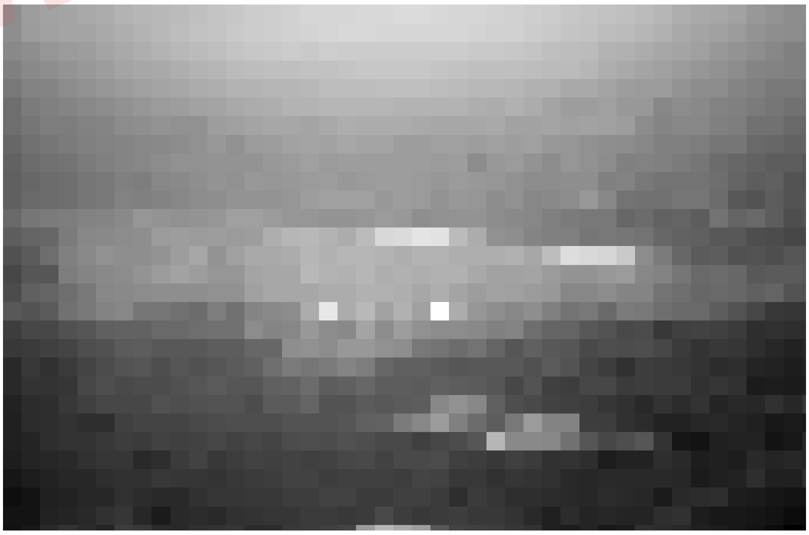
暗通道先验去雾实验 [redacted]

```
clc;clear;close all;


imag = imread('.\test images\haze4.jpg');

figure;

imshow(imag);title('原始图像','FontSize',15);

imagRGB = double(imag)./255;

window =10; eps = 0.03;alpha = 0.001;
```

Dark Channel

```
r = imagRGB(:,:,1); g = imagRGB(:,:,1); b = imagRGB(:,:,1);

[m,n] = size(r);
```

```
% Find the Minimum

dark_temp = zeros(m,n);

for i = 1:m

   for j = 1:n

      dark_temp(i,j) = min([r(i,j),g(i,j),b(i,j)]);

   end

end


% Minimum Filtering

d = ones(window,window);

fun = @(block_struct)min(min(block_struct.data))*d;

dark = blockproc(dark_temp, [window window], fun);

dark_channel = dark(1:m, 1:n);

figure;imshow(dark_channel,[]);

title('暗通道','FontSize',15);
```



暗通道

Guided Filter

```
% guidance image: I
% filtering input image: p
% local window radius: r
% regularization parameter: eps
r = window; I = dark_channel; p = dark_channel;
[hei, wid] = size(I);
N = boxfilter(ones(hei, wid), r);

mean_I = boxfilter(I, r) ./ N;
mean_p = boxfilter(p, r) ./ N;
mean_Ip = boxfilter(I.*p, r) ./ N;

cov_Ip = mean_Ip - mean_I .* mean_p;

mean_II = boxfilter(I.*I, r) ./ N;
var_I = mean_II - mean_I .* mean_I;

a = cov_Ip ./ (var_I + eps);
b = mean_p - a .* mean_I;

mean_a = boxfilter(a, r) ./ N;
mean_b = boxfilter(b, r) ./ N;

q = mean_a .* I + mean_b;
% figure;imshow(q)

% Keep a Very Small Amount of Haze For the Distant Objects
```

```
w = 0.95;

% Transmission

t_x = 1 - w*q;
```

Estimating the Atmospheric Light

```
% Pick the Top 0.1% Brightest Pixels in the Dark Channel

temp = sort(q(:),'descend');

THR_temp = temp(floor(hei*wid*alpha));

xy_List = zeros(floor(hei*wid*alpha),2);

List_index = 0;

for i=1:1:hei

    for j=1:1:wid

        if(q(i,j)>=THR_temp)

            List_index = List_index + 1;

            xy_List(List_index,1) = i;

            xy_List(List_index,2) = j;

        end

    end

end

xy_List_RGB = zeros(floor(hei*wid*alpha),3);

for i = 1:1:floor(hei*wid*alpha)

   xy_List_RGB(i,1) = imagRGB(xy_List(i,1),xy_List(i,2),1);

   xy_List_RGB(i,2) = imagRGB(xy_List(i,1),xy_List(i,2),2);

   xy_List_RGB(i,3) = imagRGB(xy_List(i,1),xy_List(i,2),3);

end

A = [sum(xy_List_RGB(:,1)) sum(xy_List_RGB(:,2)) sum(xy_List_RGB(:,3))] ...

                                        ./(floor(hei*wid*alpha))
```

```
A =

   0.5584    0.6133    0.6173
```

Recovering the Scene Radiance

```
Result_Imag = zeros(hei,wid,3);

for i = 1:1:hei

   for j = 1:1:wid

      for k = 1:1:3

      Result_Imag(i,j,k) = (imagRGB(i,j,k)-A(k)) ./ t_x(i,j) + A(k);

      end

   end

end

figure; imshow(Result_Imag,[]);

title('去雾图像','FontSize',15);
```