**作业1 图像增强**

高鹏昺 15051317

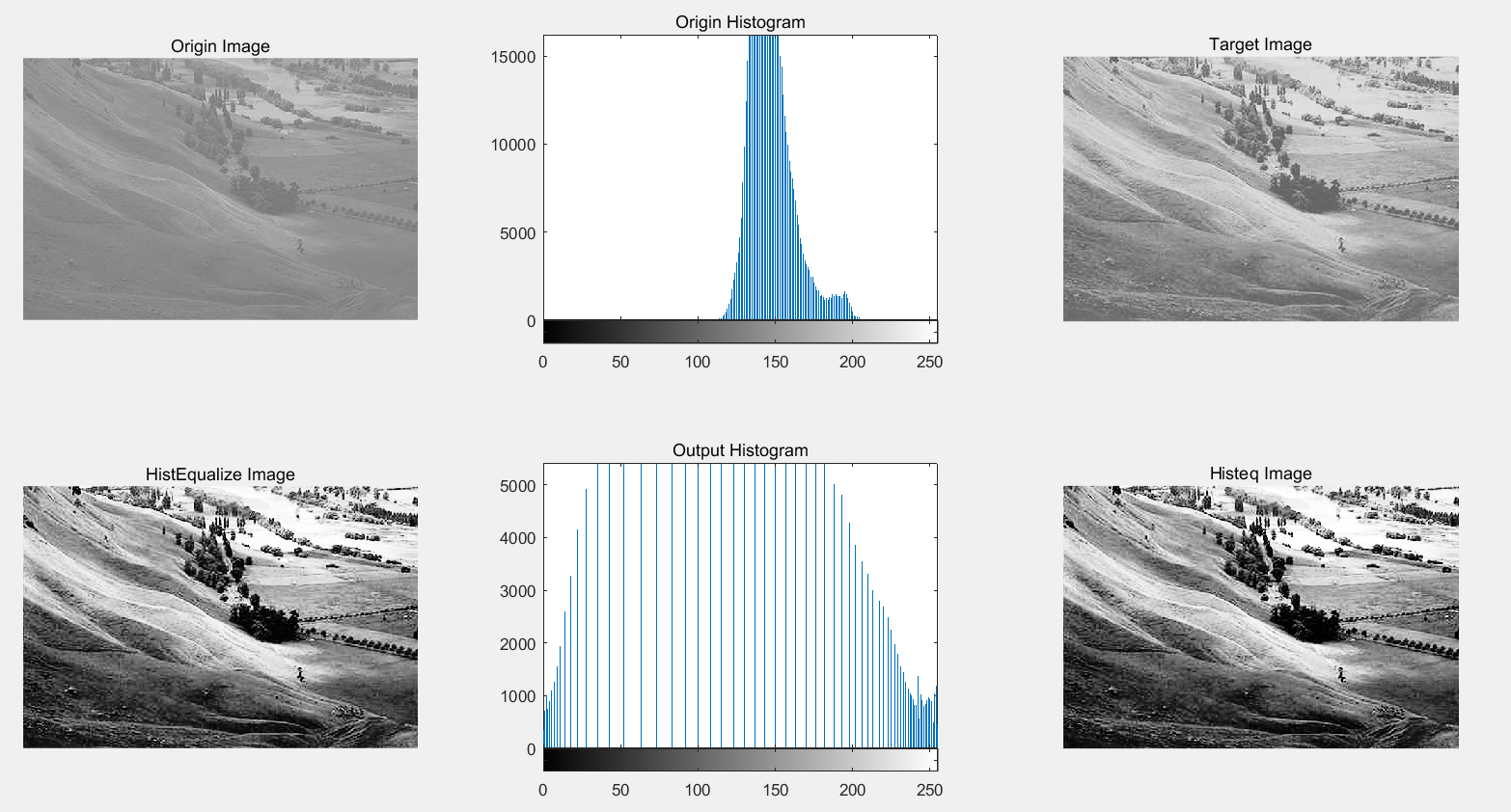
**题目1:**

**代码：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **程序编号** | **P1** | **文件名称** | **Program1.m** | **说明** | **题目1主代码** |
| Image1 = imread('Image1-1.1.png');  Image2 = imread('Image1-1.2.png');  Image3 = HistEqualize(I);  Image4 = histeq(I);    figure;  subplot(2,3,1); imshow(Image1); title('Origin Image');  subplot(2,3,2); imhist(Image1); title('Origin Histogram');  subplot(2,3,3); imshow(Image2); title('Target Image');  subplot(2,3,4); imshow(Image3); title('HistEqualize Image');  subplot(2,3,5); imhist(Image3); title('Output Histogram');  subplot(2,3,6); imshow(Image4); title('Histeq Image'); | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **程序编号** | **P2** | **文件名称** | **HistEqualize.m** | **说明** | **自编直方图均衡化函数** |
| function M = HistEqualize(Image)  n = 255;  C = zeros(1,n);  shape = size(Image);  H = shape(1);  L = shape(2);  for i = 1:H  for j = 1:L  C(Image(i,j)) = C(Image(i,j))+1;  end  end  C = C/sum(C);  P = cumsum(C);  M = uint8(zeros(H,L));  for i = 1:H  for j = 1:L  M(i,j) = uint8(n \* P(Image(i,j)));  end  end | | | | | |

**结果图：**





**题目2:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **程序编号** | **P3** | **文件名称** | **Program2.m** | **说明** | **题目2主代码** |
| Image1 = imread('Image1-2.1.jpg');  Image2 = imread('Image1-2.2.jpg');    h1 = fspecial('average', 3);  h2 = fspecial('average', 7);    Image1\_avg3 = Filter(h1, Image1);  Image1\_avg7 = Filter(h2, Image1);  Image2\_avg3 = Filter(h1, Image2);  Image2\_avg7 = Filter(h2, Image2);    figure;  subplot(2,3,1); imshow(Image1); title('Pepper Noise Image');  subplot(2,3,2); imshow(Image1\_avg3, []); title('3\*3 Average Filter');  subplot(2,3,3); imshow(Image1\_avg7, []); title('7\*7 Average Filter');  subplot(2,3,4); imshow(Image2); title('Gaussian Noise Image');  subplot(2,3,5); imshow(Image2\_avg3, []); title('3\*3 Average Filter');  subplot(2,3,6); imshow(Image2\_avg7, []); title('7\*7 Average Filter');        Image3\_avg3 = RGBFilter(Image1, 3);  Image3\_avg7 = RGBFilter(Image1, 7);  Image4\_avg3 = RGBFilter(Image2, 3);  Image4\_avg7 = RGBFilter(Image2, 7);    figure;  subplot(2,3,1); imshow(Image1); title('Pepper Noise Image');  subplot(2,3,2); imshow(Image3\_avg3, []); title('3\*3 Average Filter');  subplot(2,3,3); imshow(Image3\_avg7, []); title('7\*7 Average Filter');  subplot(2,3,4); imshow(Image2); title('Gaussian Noise Image');  subplot(2,3,5); imshow(Image4\_avg3, []); title('3\*3 Average Filter');  subplot(2,3,6); imshow(Image4\_avg7, []); title('7\*7 Average Filter'); | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **程序编号** | **P4** | **文件名称** | **Filter.m** | **说明** | **滤波器函数** |
| function M = Filter(h, Image)  R = Image(:,:,1);  G = Image(:,:,2);  B = Image(:,:,3);    R\_ = filter2(h, R);  G\_ = filter2(h, G);  B\_ = filter2(h, B);    M(:,:,1) = uint8(R\_);  M(:,:,2) = uint8(G\_);  M(:,:,3) = uint8(B\_);    end | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **程序编号** | **P5** | **文件名称** | **RGBFilter.m** | **说明** | **自编彩图滤波器函数** |
| function M = RGBFilter(Image, h)  R = Image(:,:,1);  G = Image(:,:,2);  B = Image(:,:,3);    R\_ = AverageFilter(R, h);  G\_ = AverageFilter(G, h);  B\_ = AverageFilter(B, h);    M(:,:,1) = uint8(R\_);  M(:,:,2) = uint8(G\_);  M(:,:,3) = uint8(B\_);    end | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **程序编号** | **P6** | **文件名称** | **AverageFilter.m** | **说明** | **自编均值滤波器函数** |
| function M = AverageFilter(Image, h)  Conv = ones(h);  shape = size(Image);  H = shape(1);  L = shape(2);  H\_ = H - h + 1;  L\_ = L - h + 1;    Image = Image(:,:,1);    M = uint8(zeros(H\_, L\_));  for i = 1:H\_  for j = 1:L\_  s = uint16(0);  pi = i + floor(h/2);  pj = j + floor(h/2);  for p = -floor(h/2):floor(h/2)  for q = -floor(h/2):floor(h/2)  s = s + uint16(Image(pi+p,pj+q)\*1);  end  end  M(i,j) = uint8(s/(h\*h));  end  end | | | | | |

**结果图：**

**滤波器+MATLAB自带均值滤波器函数结果图**



**彩图滤波器+自编均值滤波器函数结果图**

