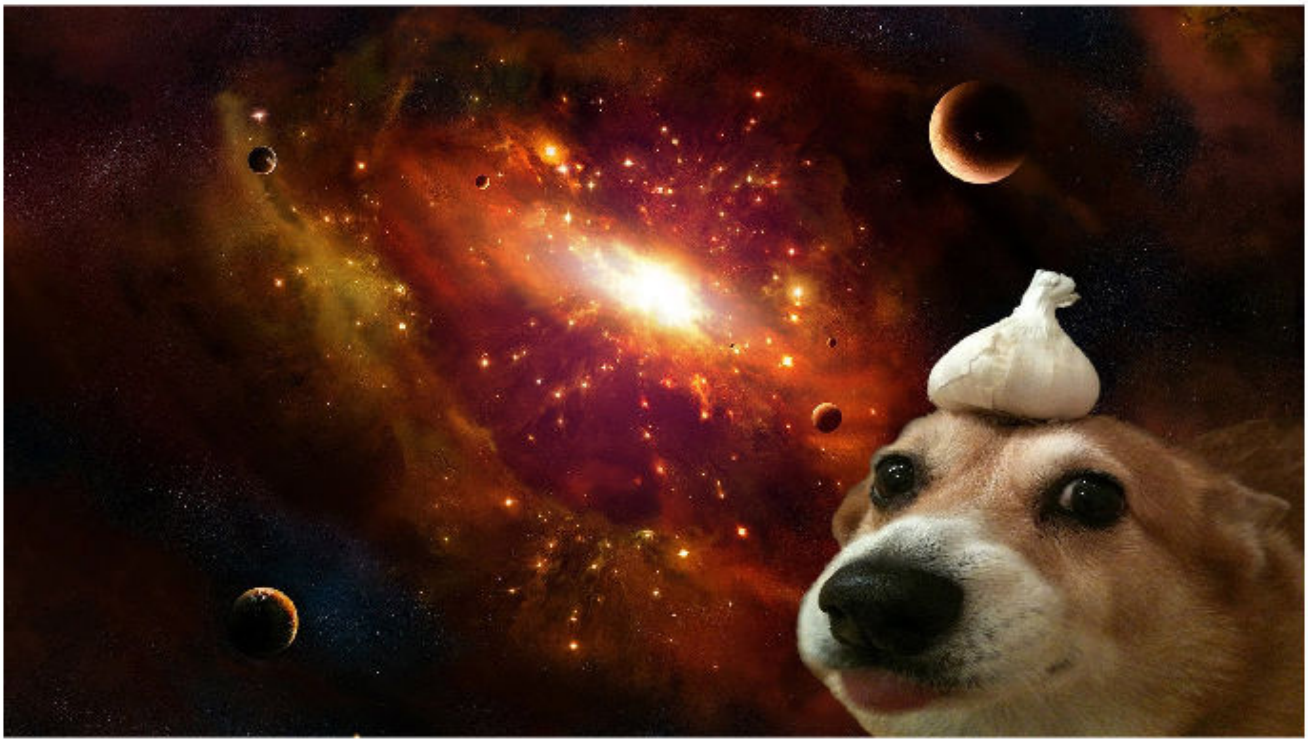
Lab 4 - Color Representation and Processing

Nick Snyder

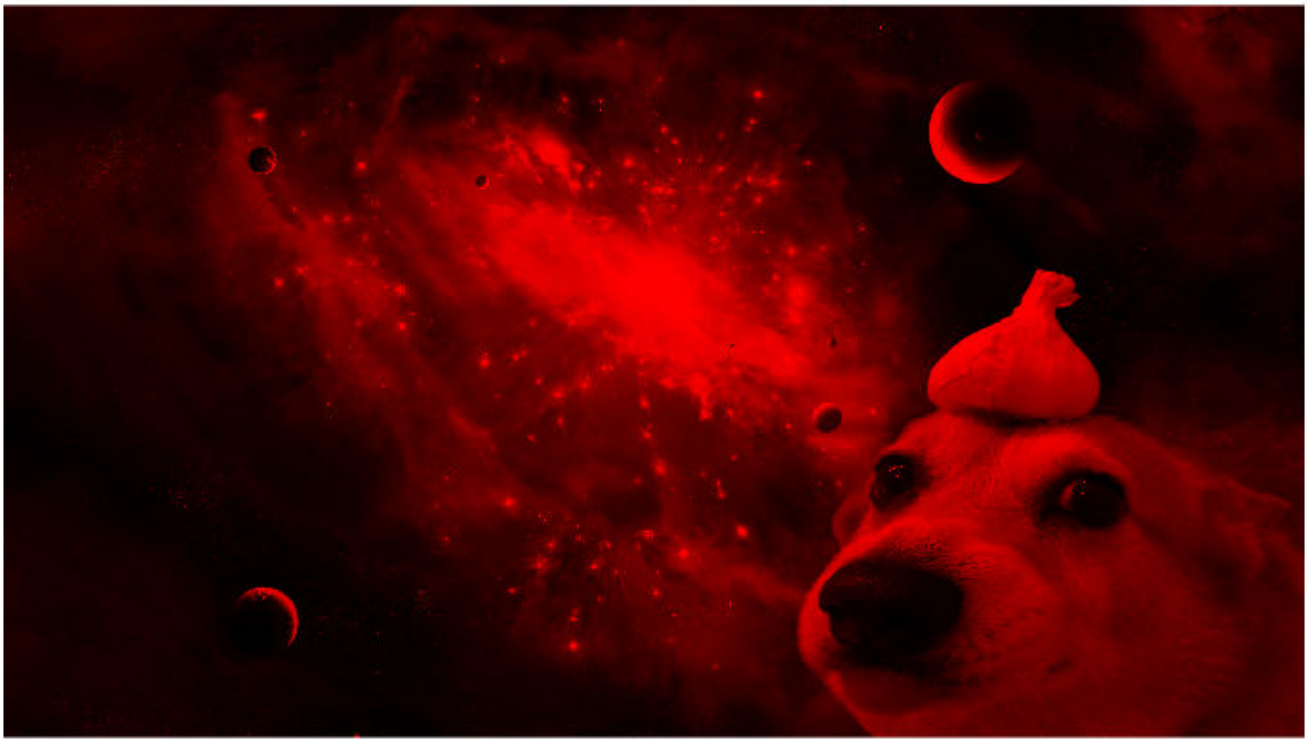
Table of Contents

Part 1: Dissecting a Color Image into its RGB Planes

```
im = imread('garlicdog.jpg');

imRed = im;
imRed(:, :, [2, 3]) = 0;
imRedGrey = im(:, :, 1);

imGreen = im;
imGreen(:, :, [1, 3]) = 0;
imGreenGrey = im(:, :, 2);

imBlue = im;
imBlue(:, :, [1, 2]) = 0;
imBlueGrey = im(:, :, 3);

imshow(im);
```



```
imshow(imRed);
```

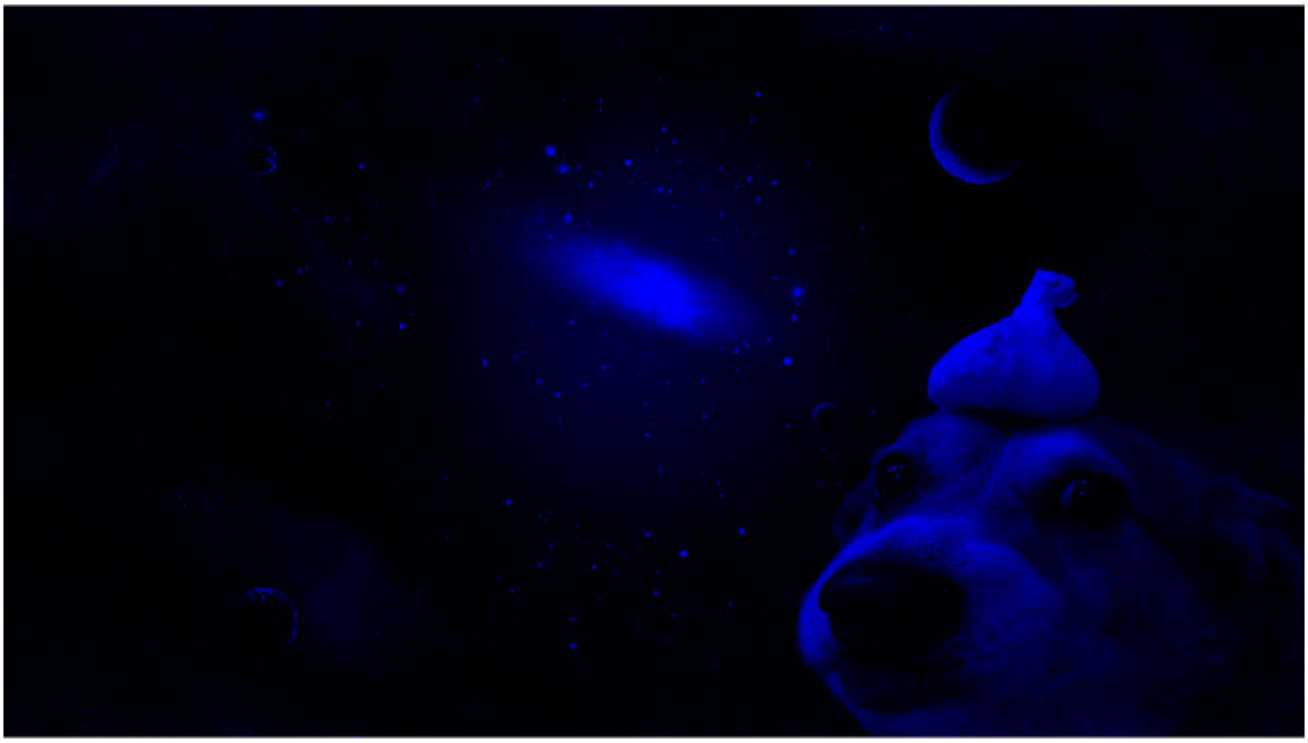


```
imshow(imRedGrey);
```

```
imshow(imGreen);
```

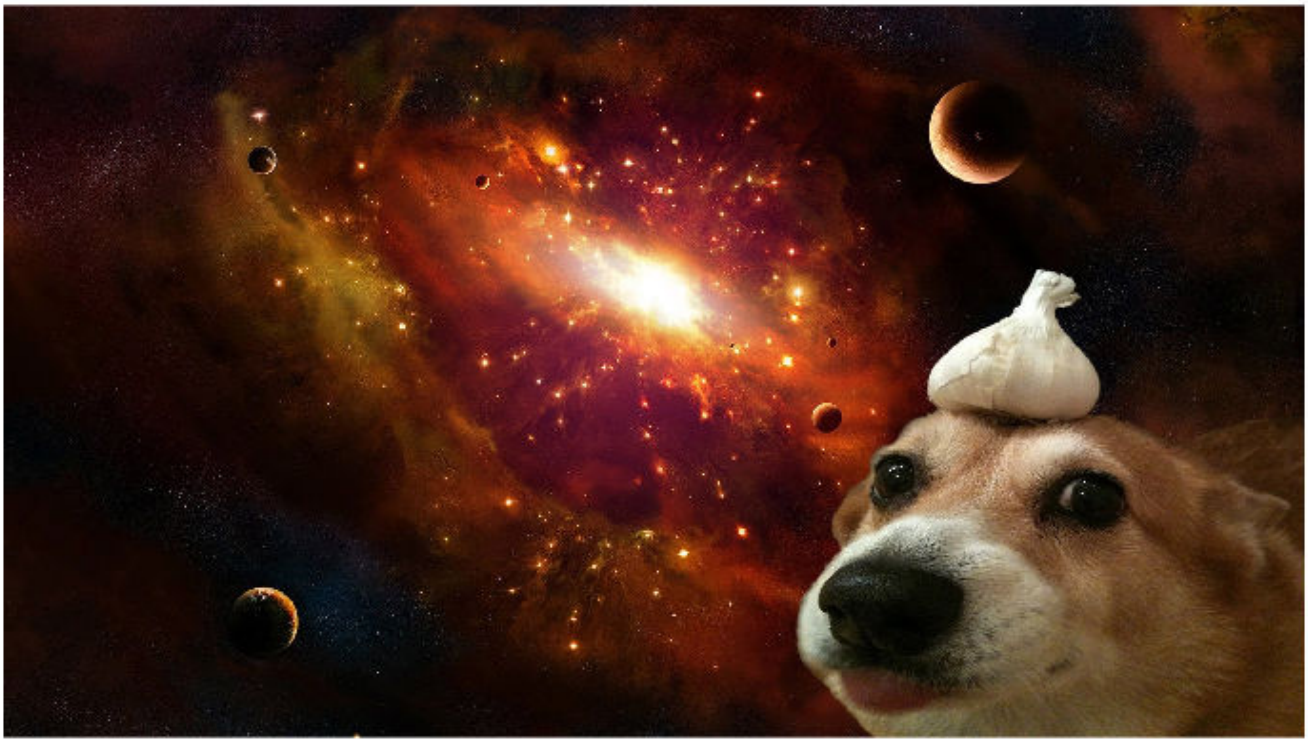
```
imshow(imGreenGrey);
```

```
imshow(imBlue);
```



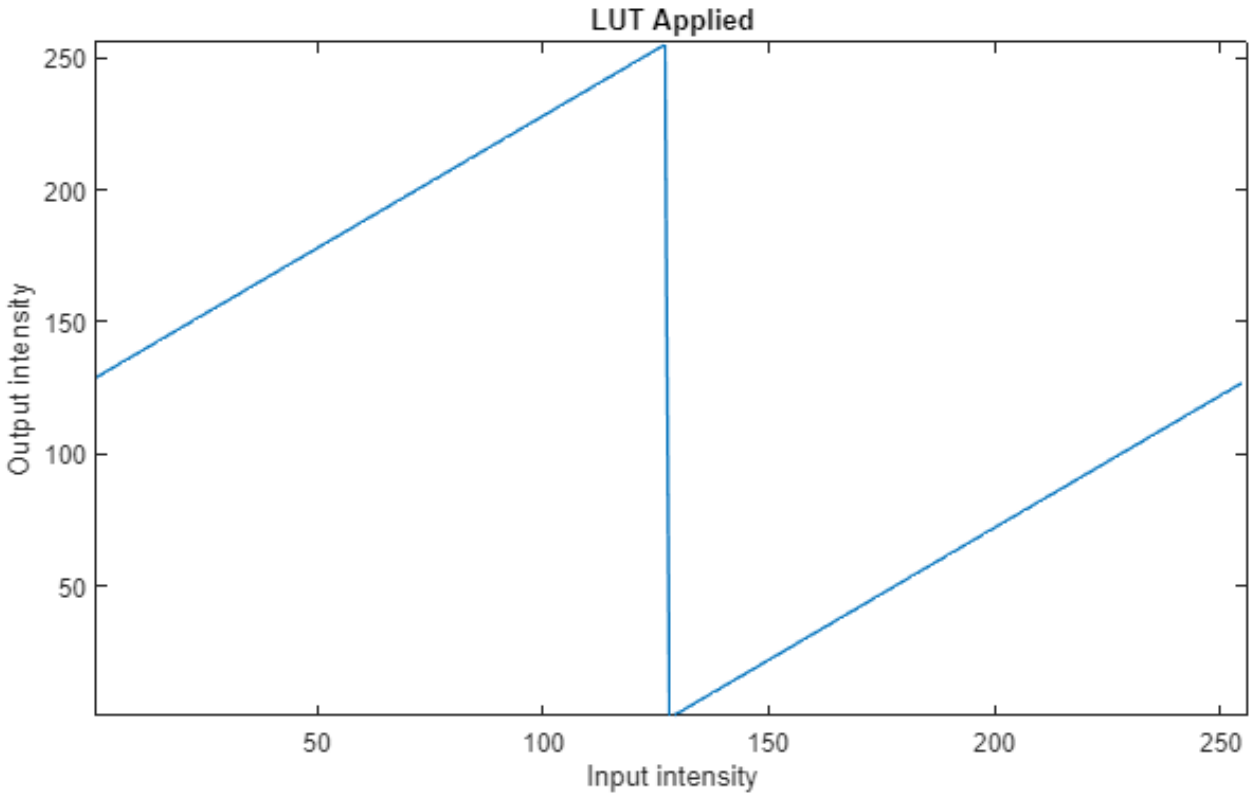
```
imshow(imBlueGrey);
```

```
imshow(imRed + imGreen + imBlue);
```

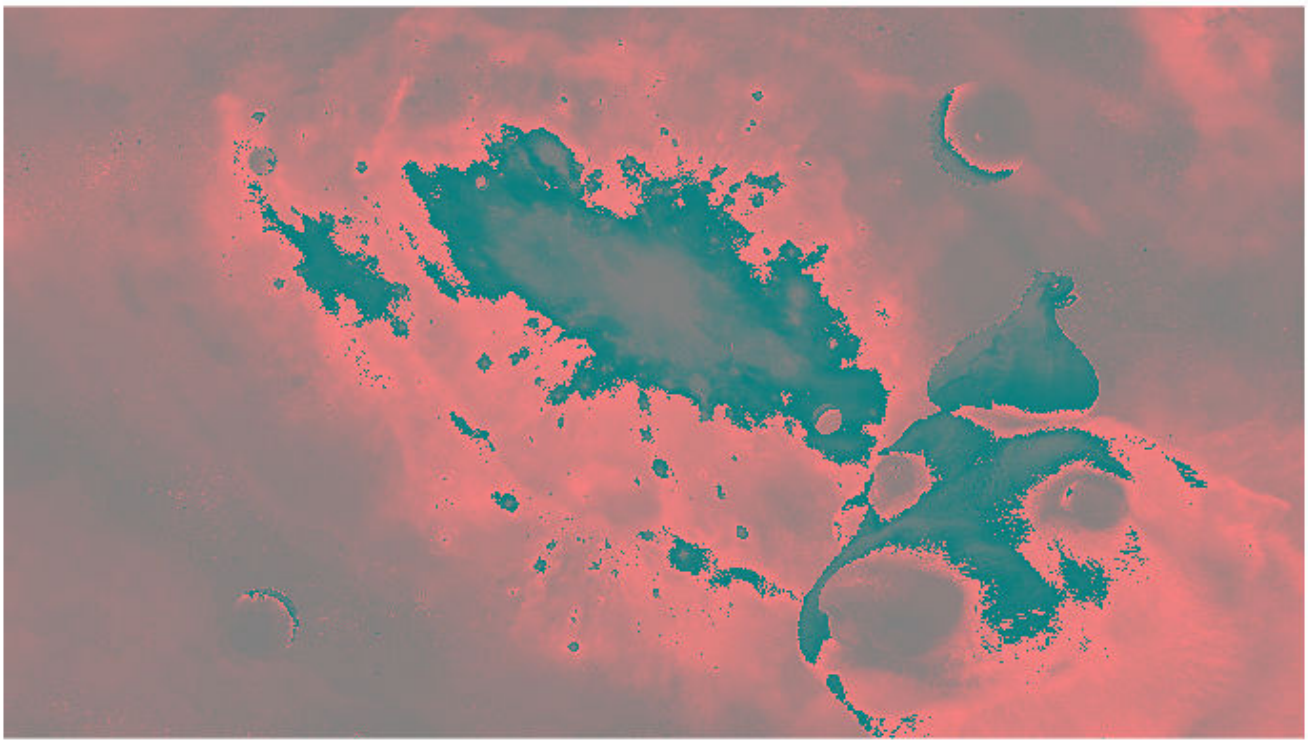


Part 2: Processing on Individual Color Channels in RGB

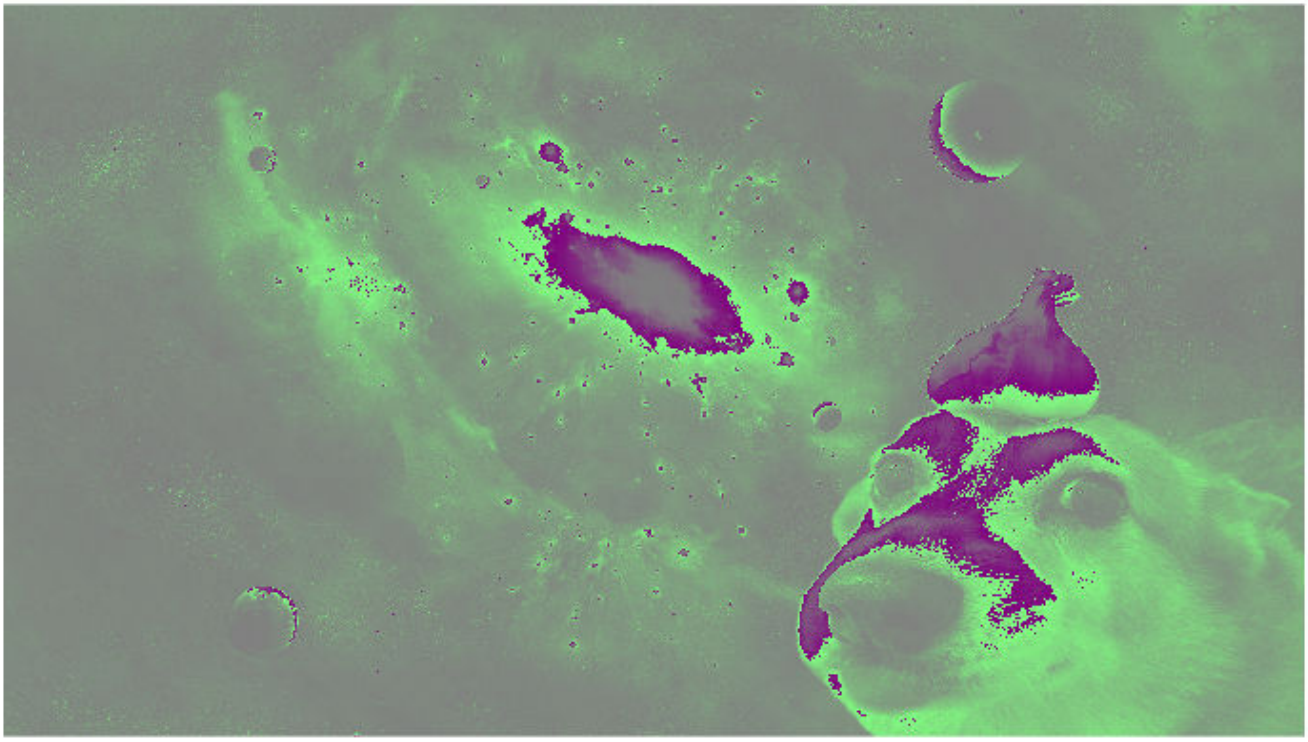
```
Ii = 0:255;
LUT = uint8(zeros([1 256]));
LUT(1:128) = Ii(1:128) + 128;
LUT(129:256) = Ii(129:256) - 128;

plot(Ii, LUT), axis([ 1 256 1 256 ]), title('LUT Applied'), xlabel('Input intensity'), ylabel('
```



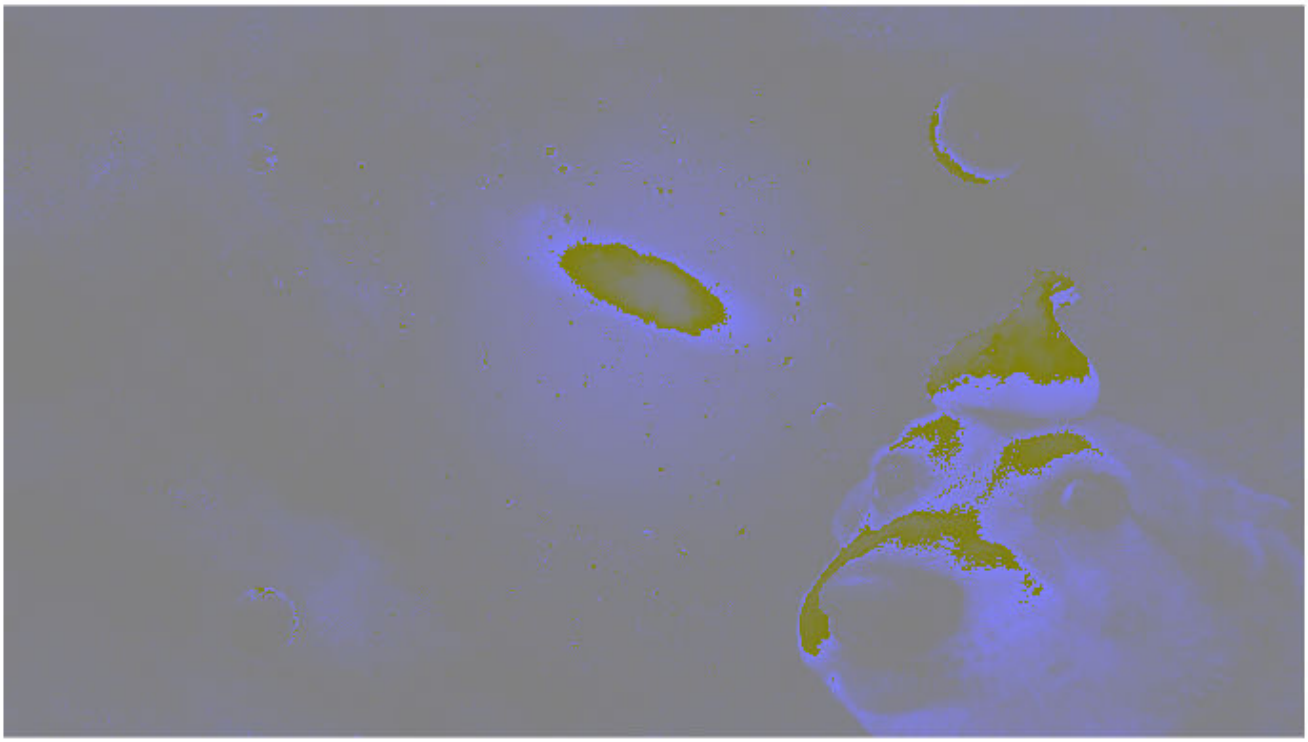

```
imshow(intlut(imRed, LUT));
```



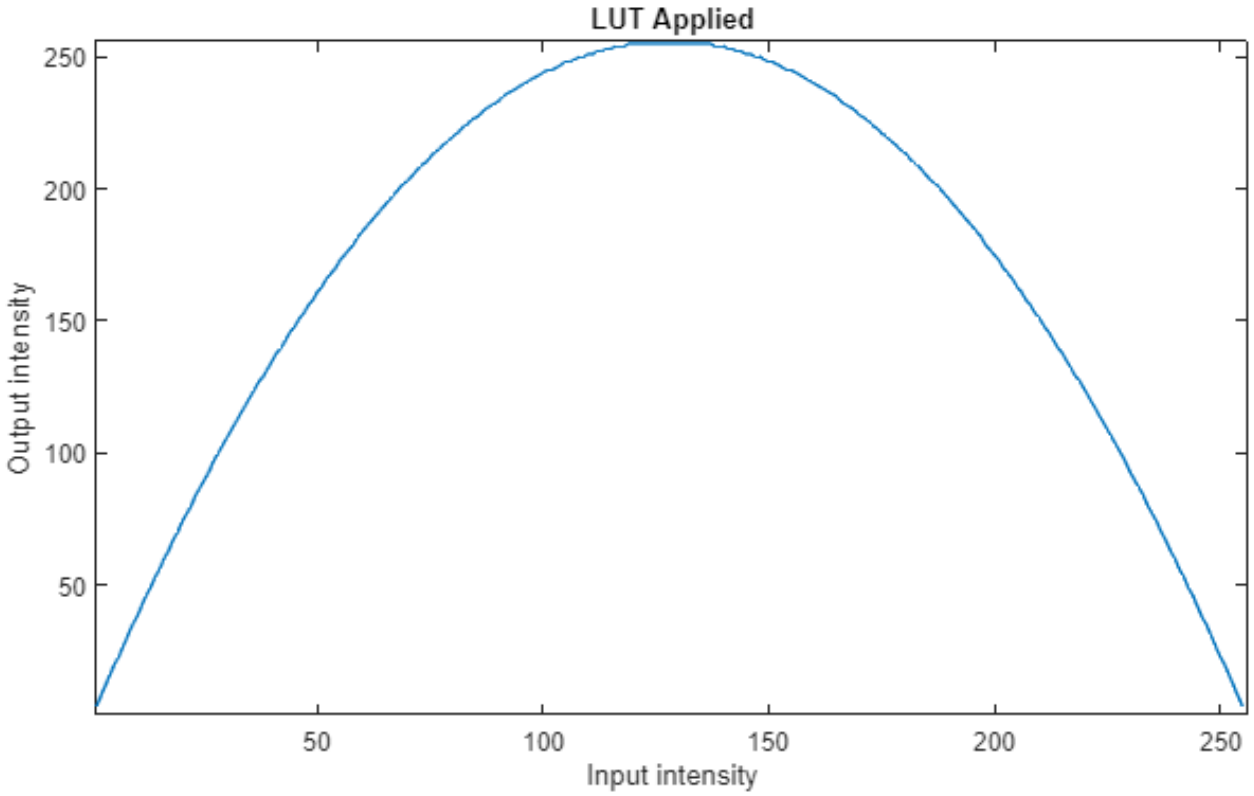
```
imshow(intlut(imGreen, LUT));
```



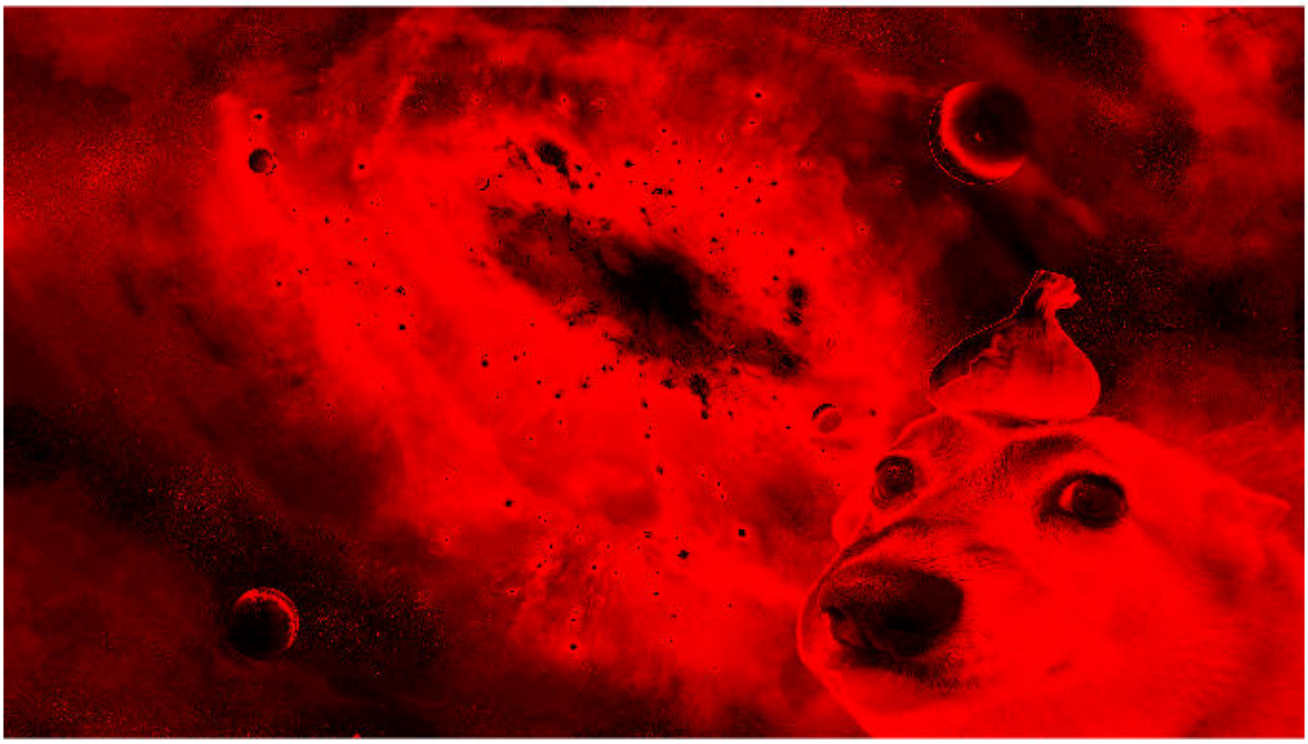
```
imshow(intlut(imBlue, LUT));
```



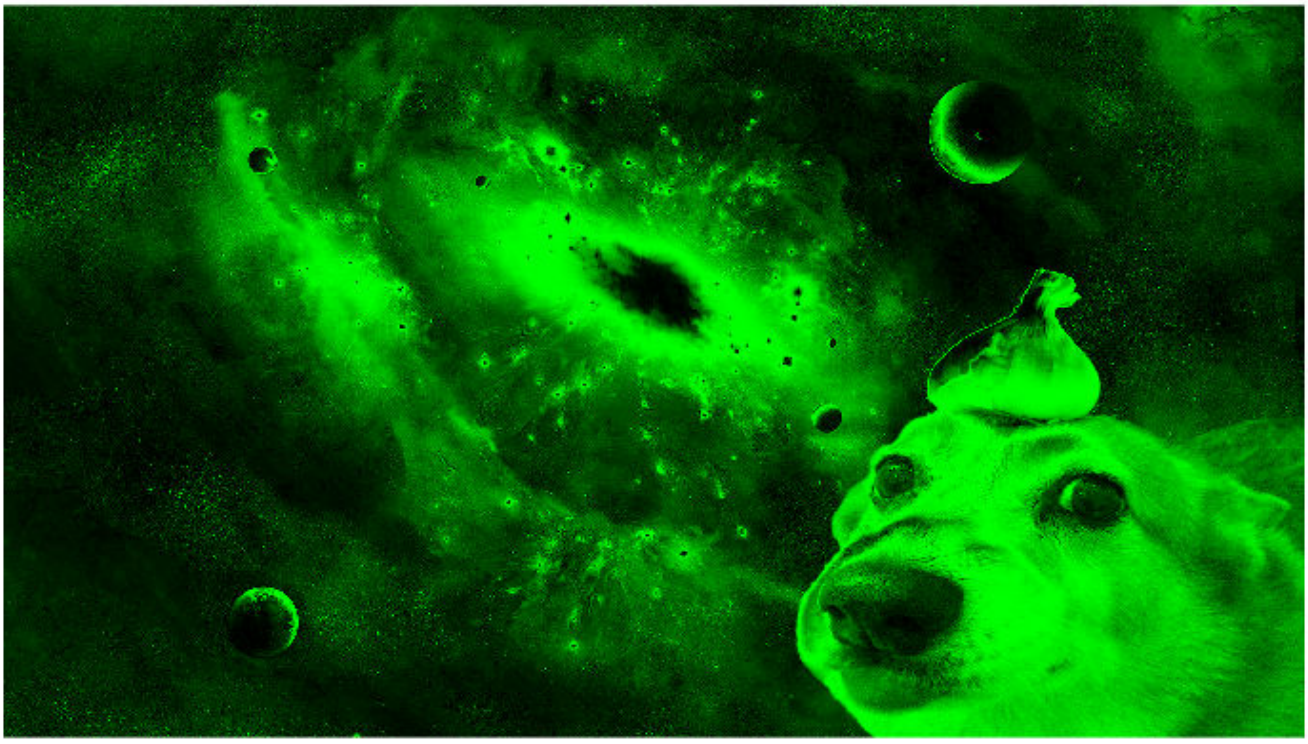
```
LUT = -(1 / 64) * (Ii .^ 2) + (4 * Ii);
LUT = uint8(LUT);

plot(Ii, LUT), axis([ 1 256 1 256 ]), title('LUT Applied'), xlabel('Input intensity'), ylabel('
```



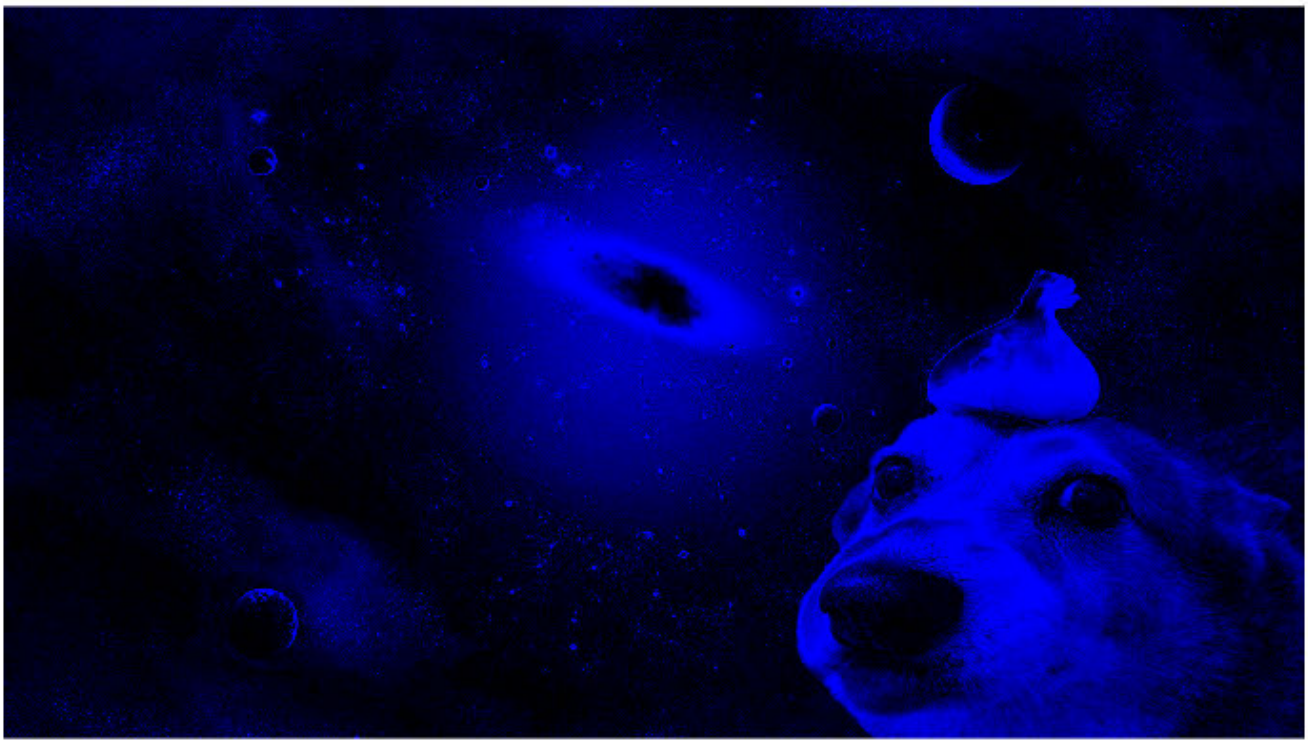

```
imshow(intlut(imRed, LUT));
```



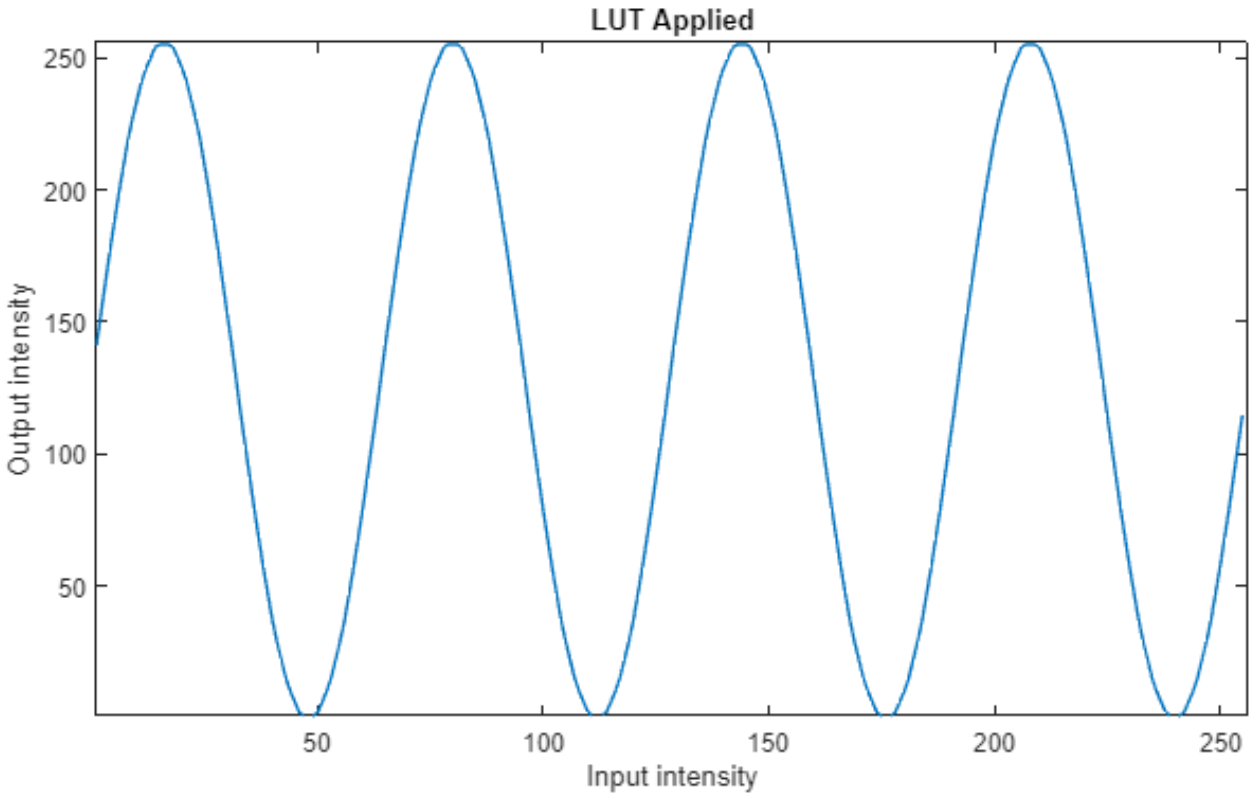
```
imshow(intlut(imGreen, LUT));
```



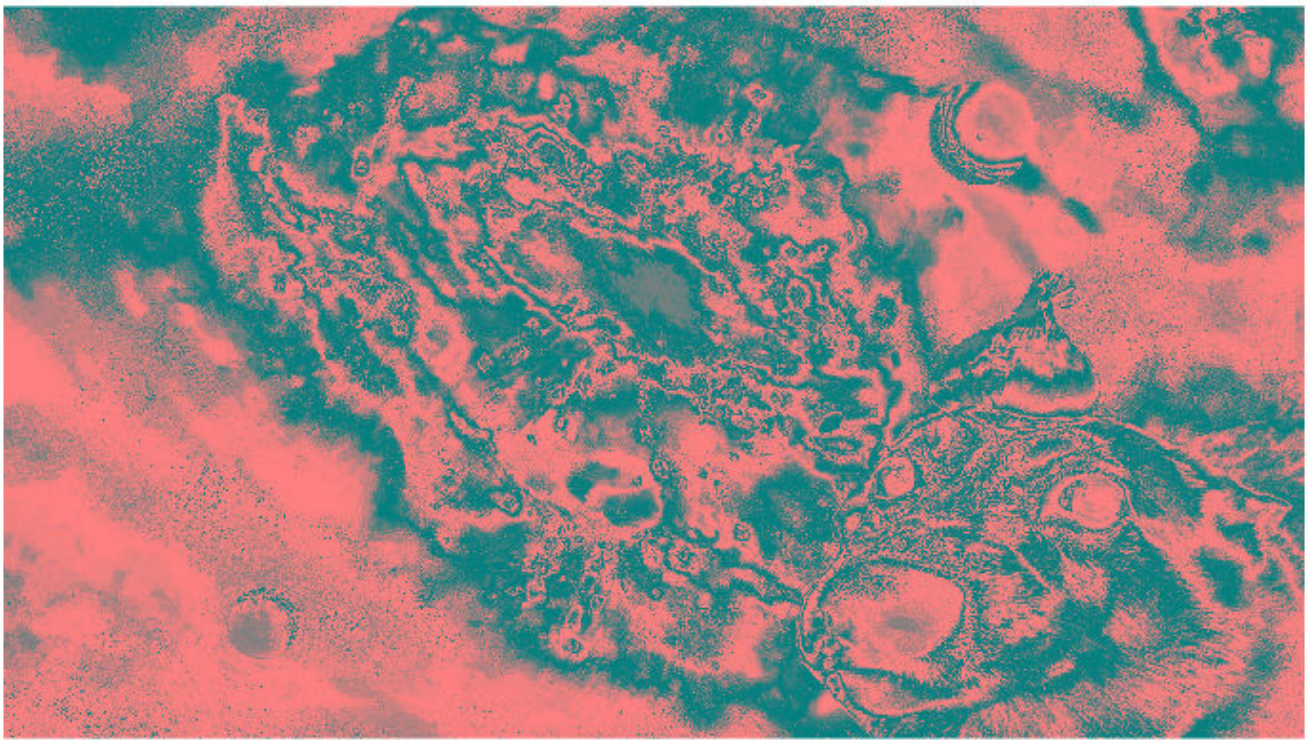
```
imshow(intlut(imBlue, LUT));
```



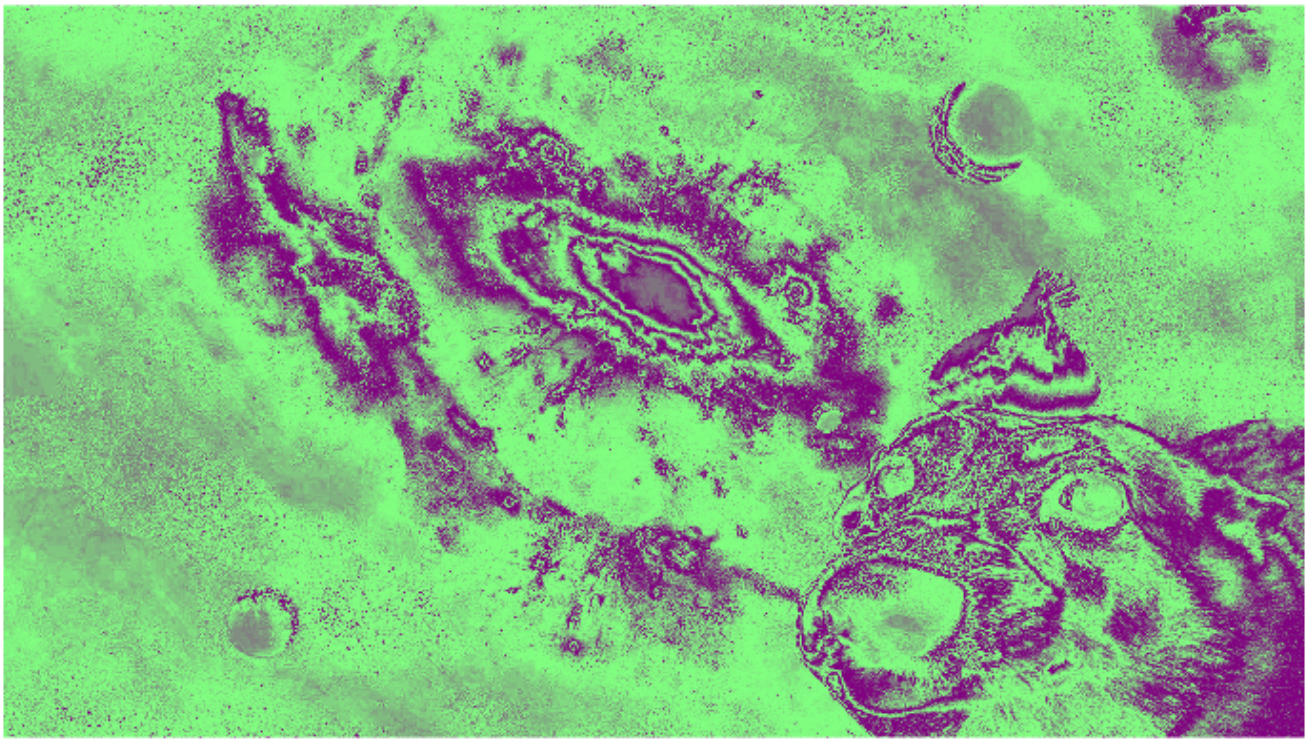
```
LUT = 128 * sin((pi / 32) * Ii) + 128;
LUT = uint8(LUT);

plot(Ii, LUT), axis([ 1 256 1 256 ]), title('LUT Applied'), xlabel('Input intensity'), ylabel('
```



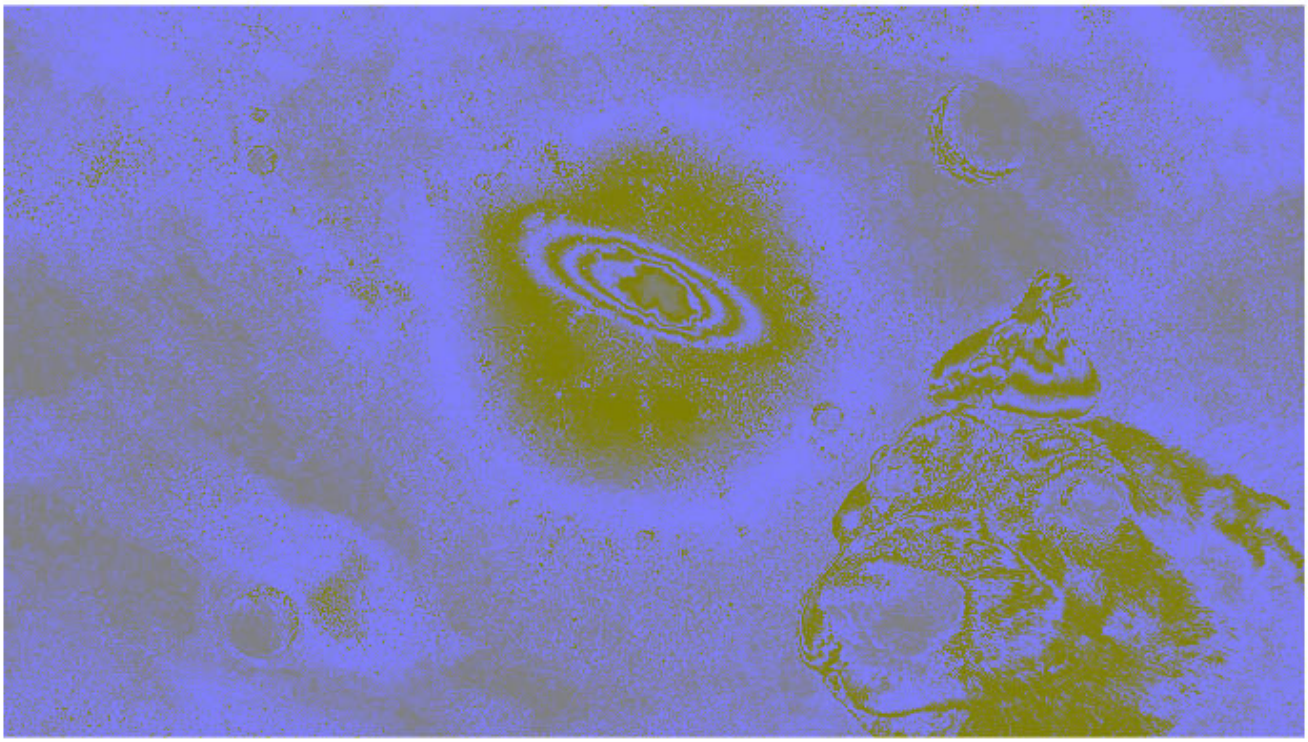

```
imshow(intlut(imRed, LUT));
```



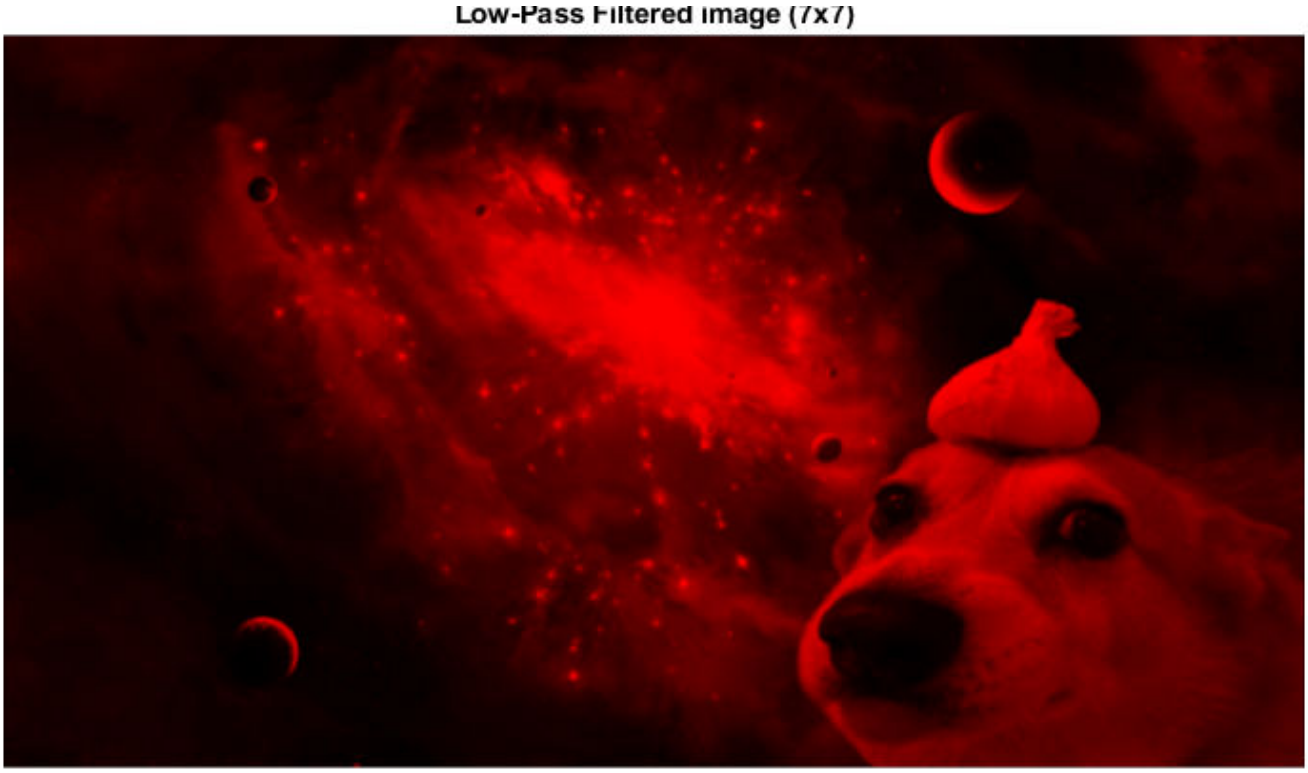
```
imshow(intlut(imGreen, LUT));
```



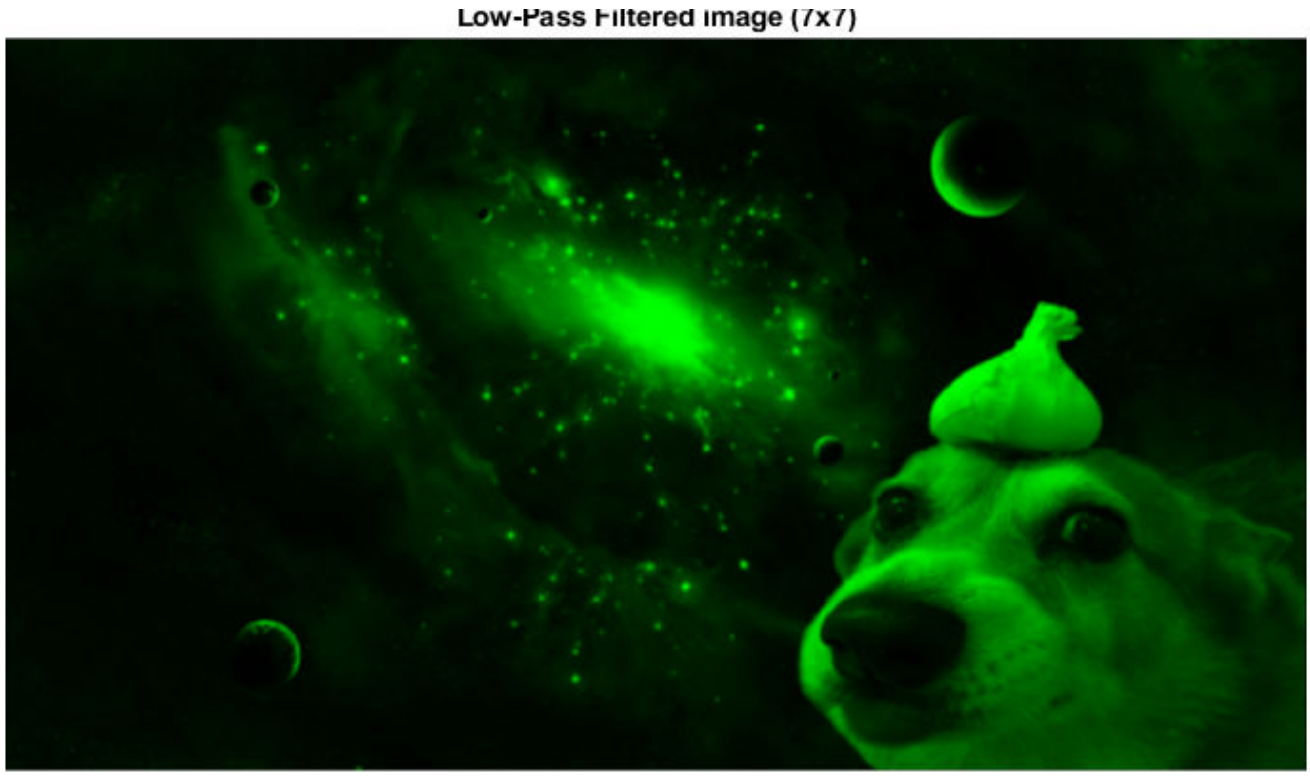
```
imshow(intlut(imBlue, LUT));
```



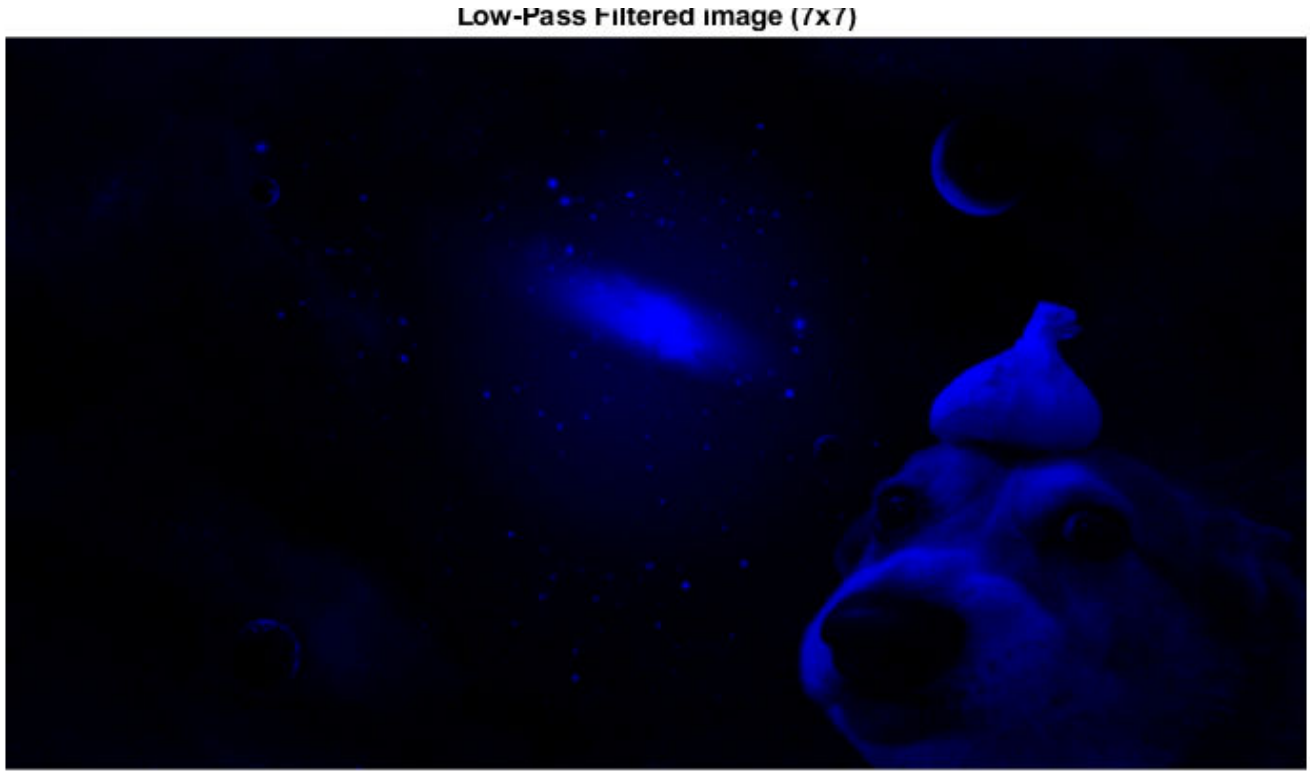
```
Lowpass7x7 = fspecial('average', 7);
Kernel = Lowpass7x7;

ImRedFiltered = imfilter(imRed, Kernel, 'conv');
ImGreenFiltered = imfilter(imGreen, Kernel, 'conv');
ImBlueFiltered = imfilter(imBlue, Kernel, 'conv');

imshow(ImRedFiltered), title('Low-Pass Filtered image (7x7)')
```



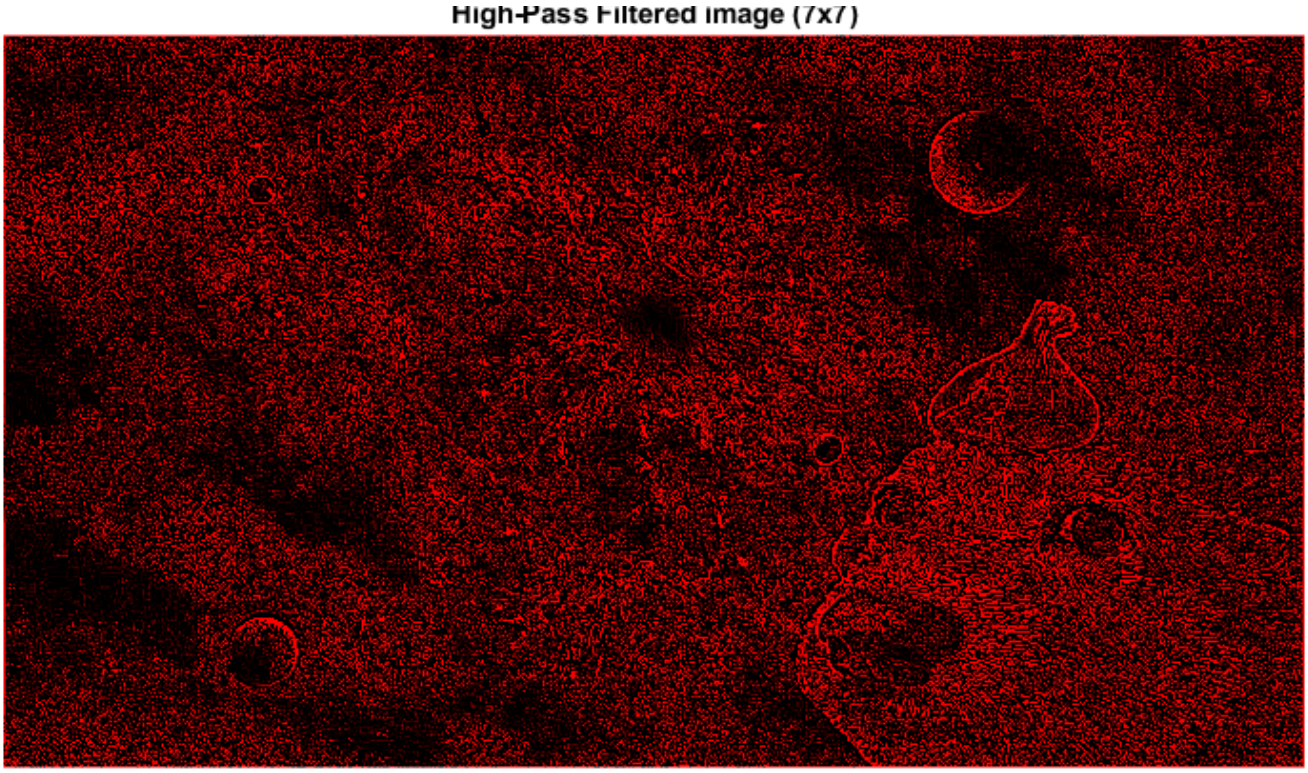

```
imshow(ImGreenFiltered), title('Low-Pass Filtered image (7x7)')
```



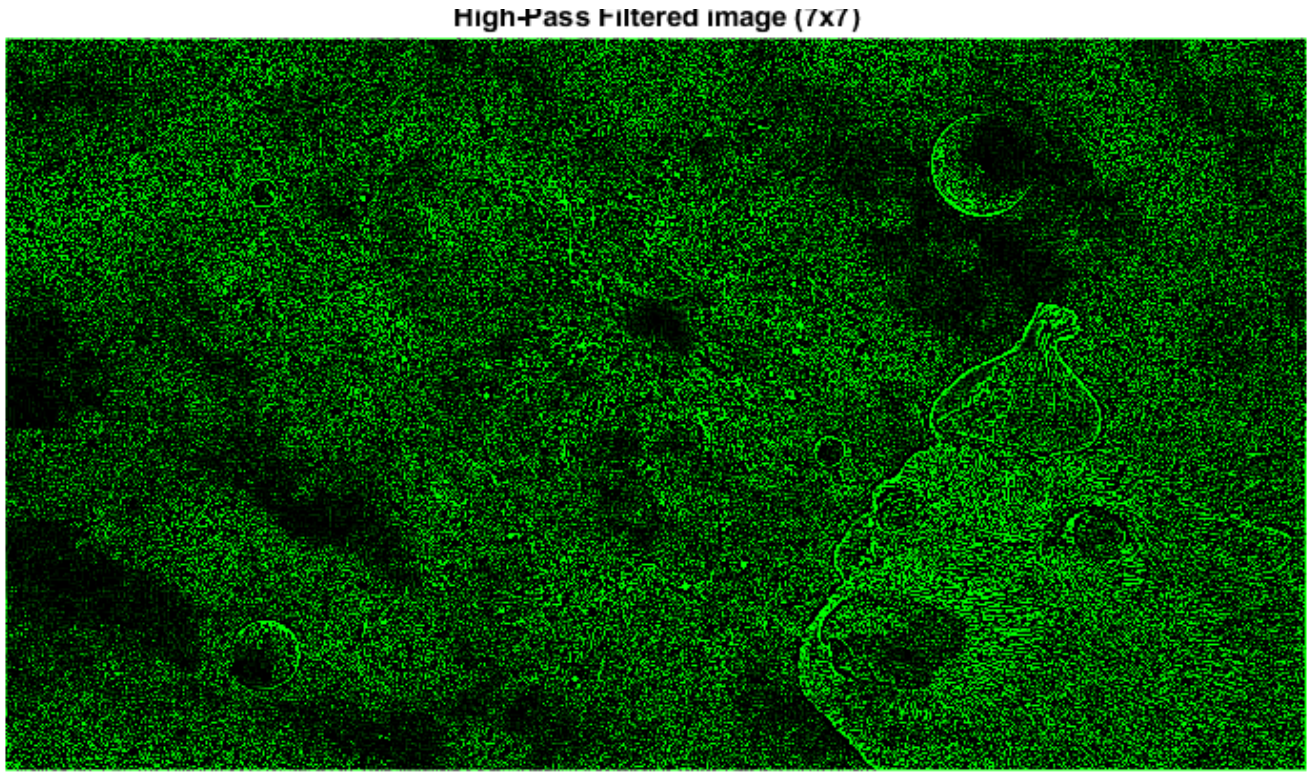

```
imshow(ImBlueFiltered), title('Low-Pass Filtered image (7x7)')
```



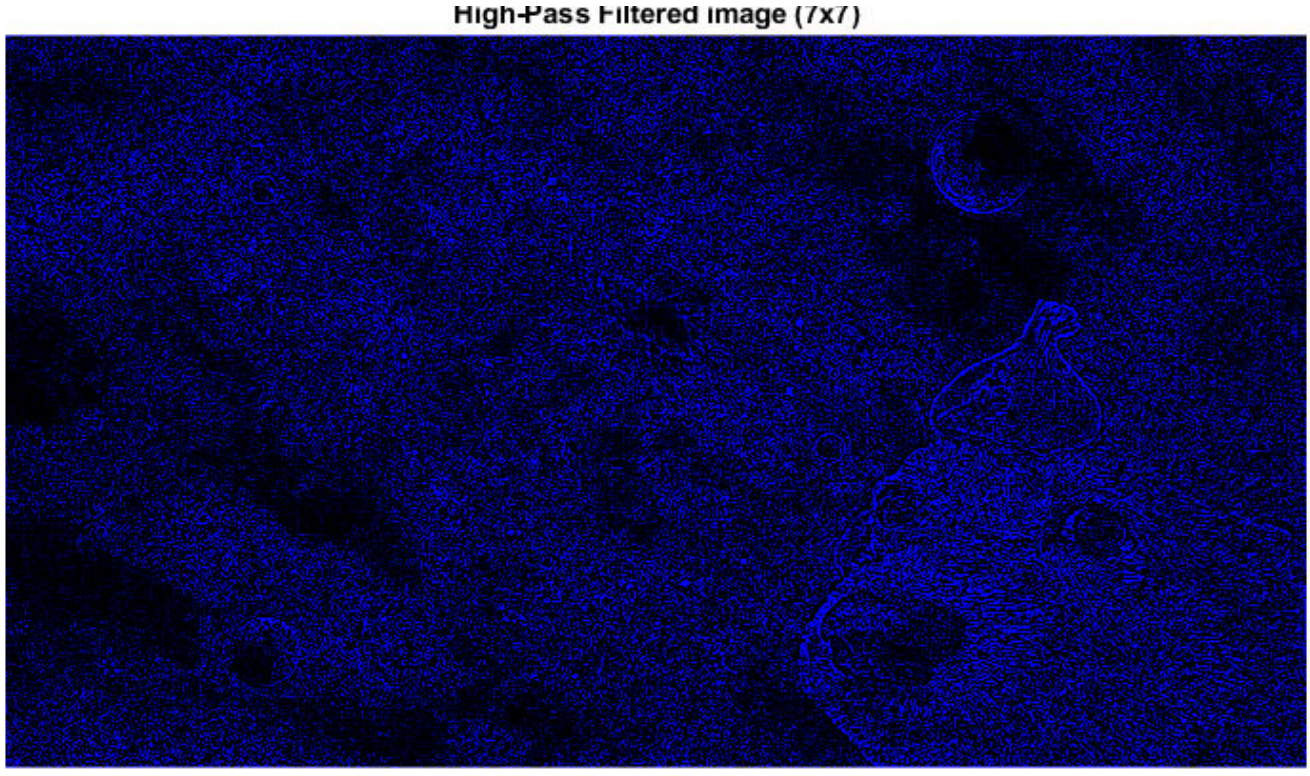
```
Highpass7x7 = [ -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, 48, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1 ];
Kernel = Highpass7x7;

ImRedFiltered = imfilter(imRed, Kernel, 'conv');
ImGreenFiltered = imfilter(imGreen, Kernel, 'conv');
ImBlueFiltered = imfilter(imBlue, Kernel, 'conv');

imshow(ImRedFiltered), title('High-Pass Filtered image (7x7)')
```




```
imshow(ImGreenFiltered), title('High-Pass Filtered image (7x7)')
```

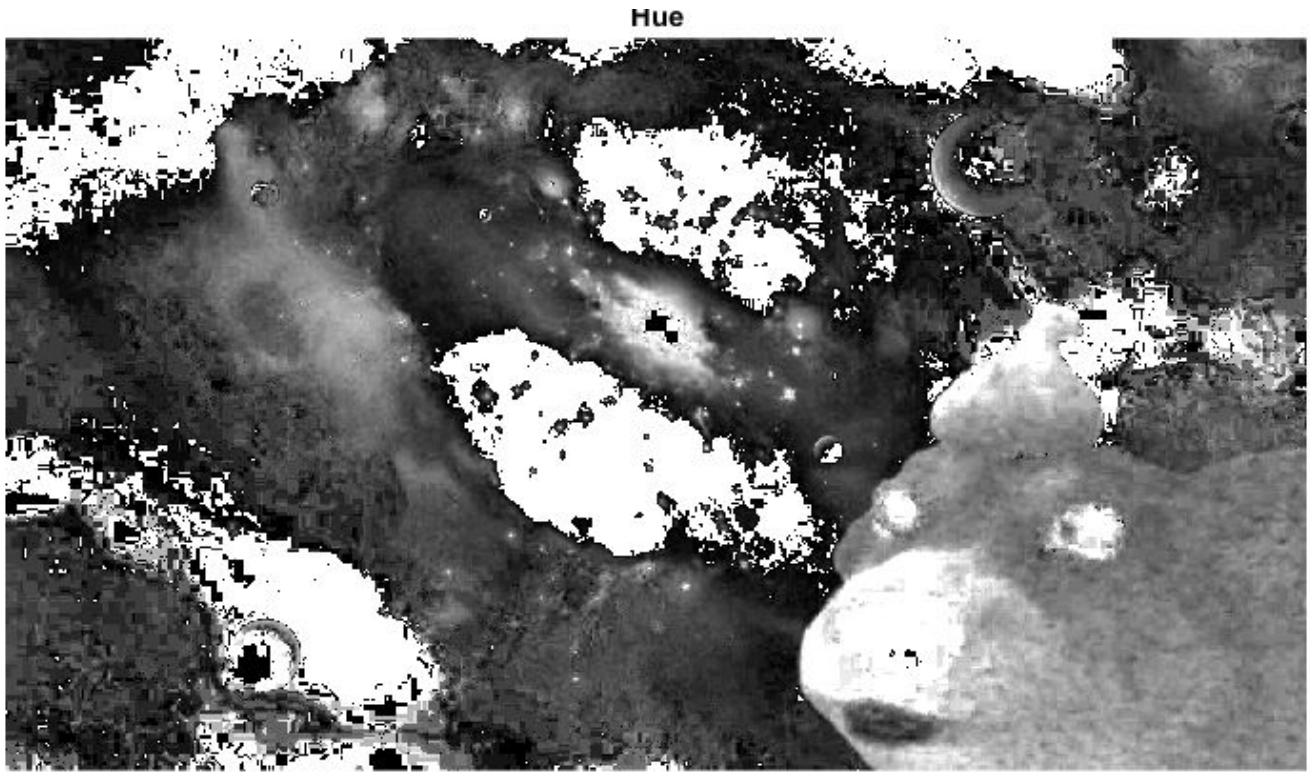



```
imshow(ImBlueFiltered), title('High-Pass Filtered image (7x7)')
```



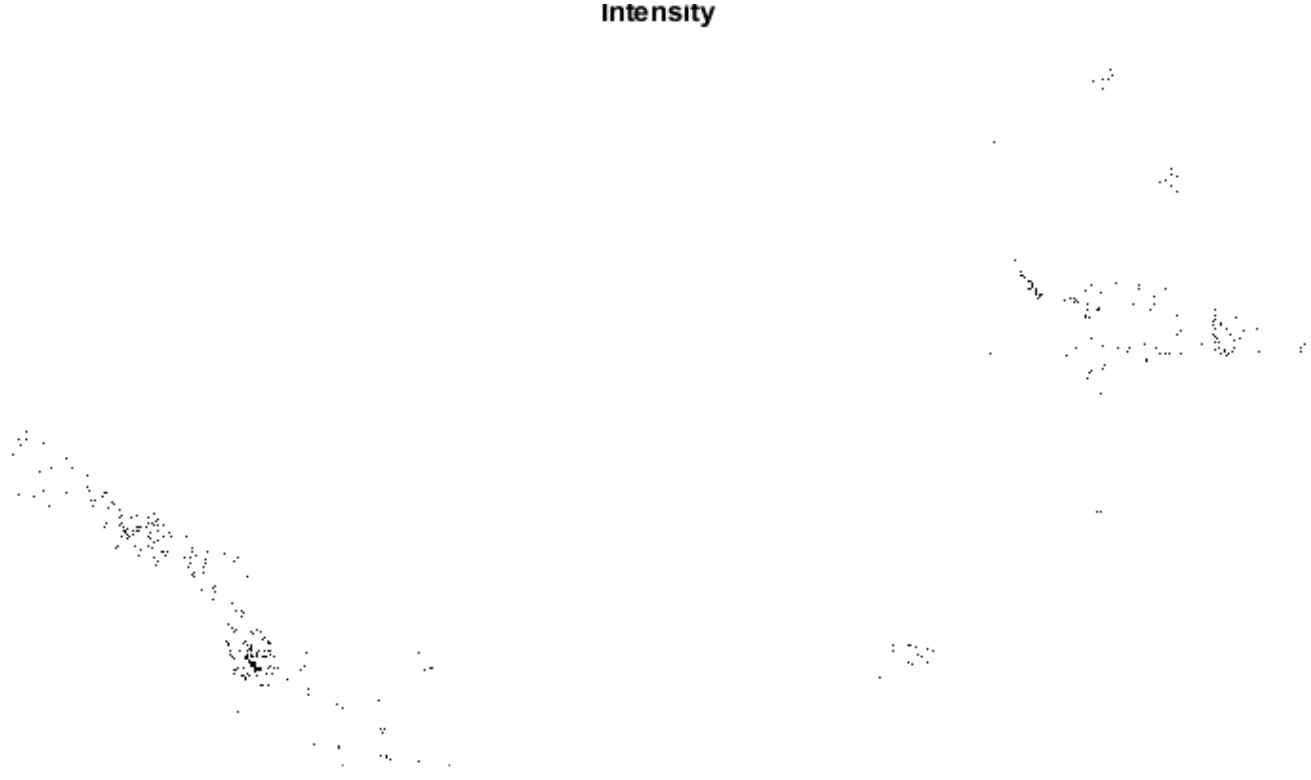
Part 3: Converting a 24-bit RGB Color Image into a 24-bit HSI Image

```
imRedGreyD = double(imRedGrey);
imGreenGreyD = double(imGreenGrey);
imBlueGreyD = double(imBlueGrey);

intensity = mean(im, 3);

saturation = 1 - (3 ./ (imRedGrey + imGreenGrey + imBlueGrey)) .* min(im, [], 3);

hue = acos((0.5 * ((imRedGreyD - imGreenGreyD) + (imRedGreyD - imBlueGreyD))) ./ (sqrt((imRedGr
hue(imBlueGreyD > imGreenGreyD) = 360 - hue(imBlueGreyD > imGreenGreyD);

HSI = zeros(size(im));
HSI(:, :, 1) = (hue);
HSI(:, :, 2) = (saturation);
HSI(:, :, 3) = uint8(intensity);

imshow(HSI(:, :, 1)), title('Hue');
```

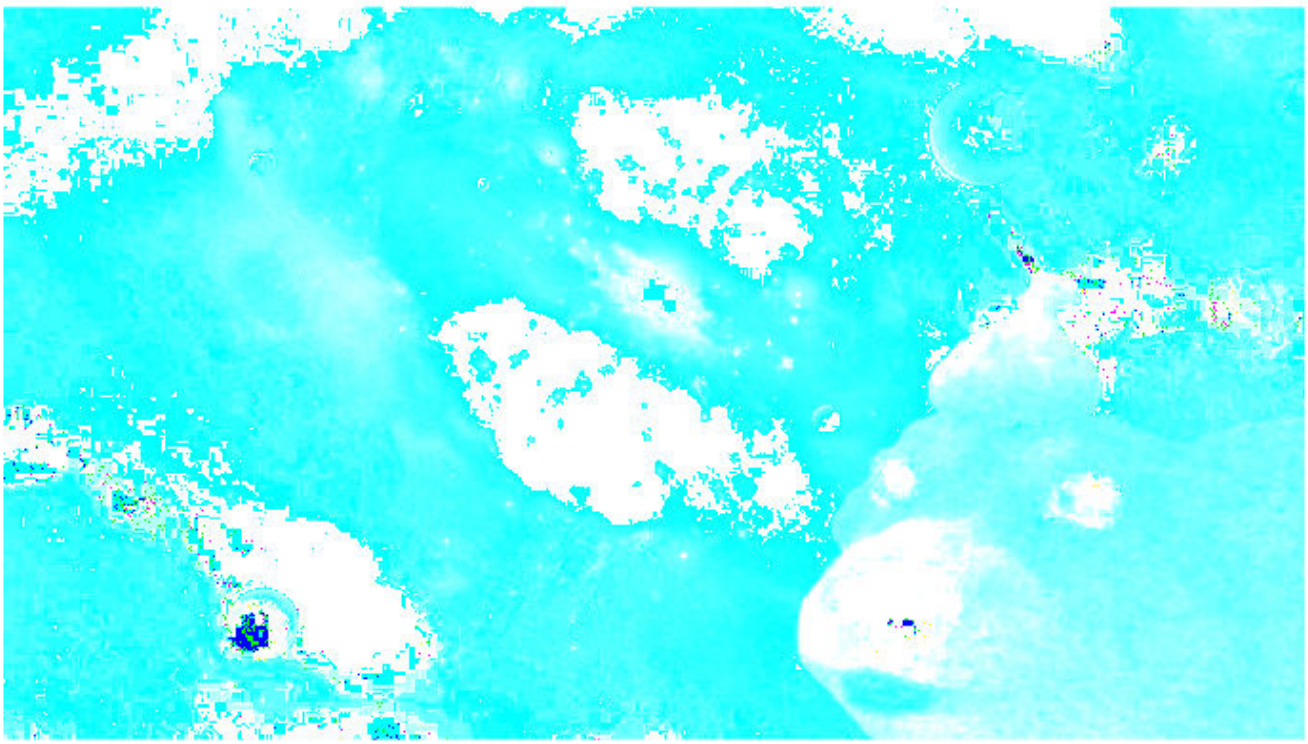



```
imshow(HSI(:, :, 2)), title('Saturation');
```


```
imshow(HSI(:, :, 3)), title('Intensity');
```

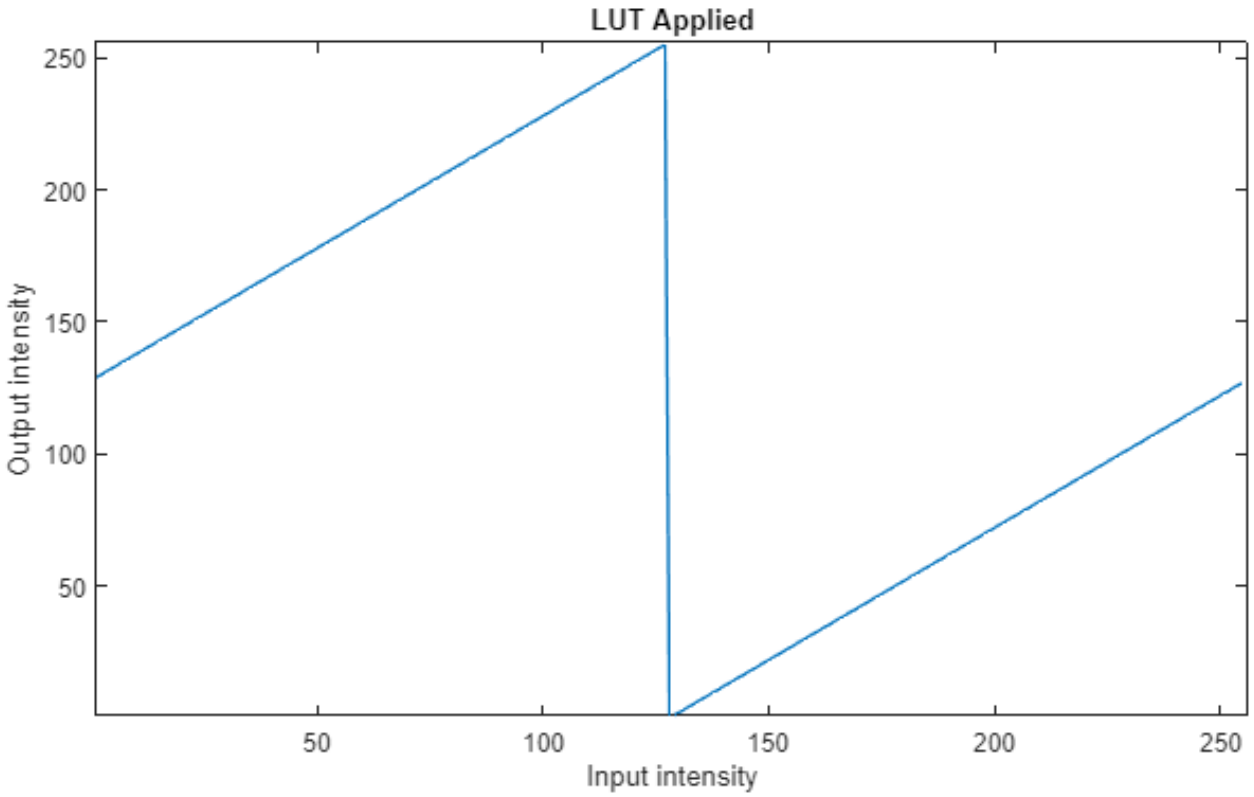


```
imshow(HSI)
```



Part 4: Processing in HSI Color Space

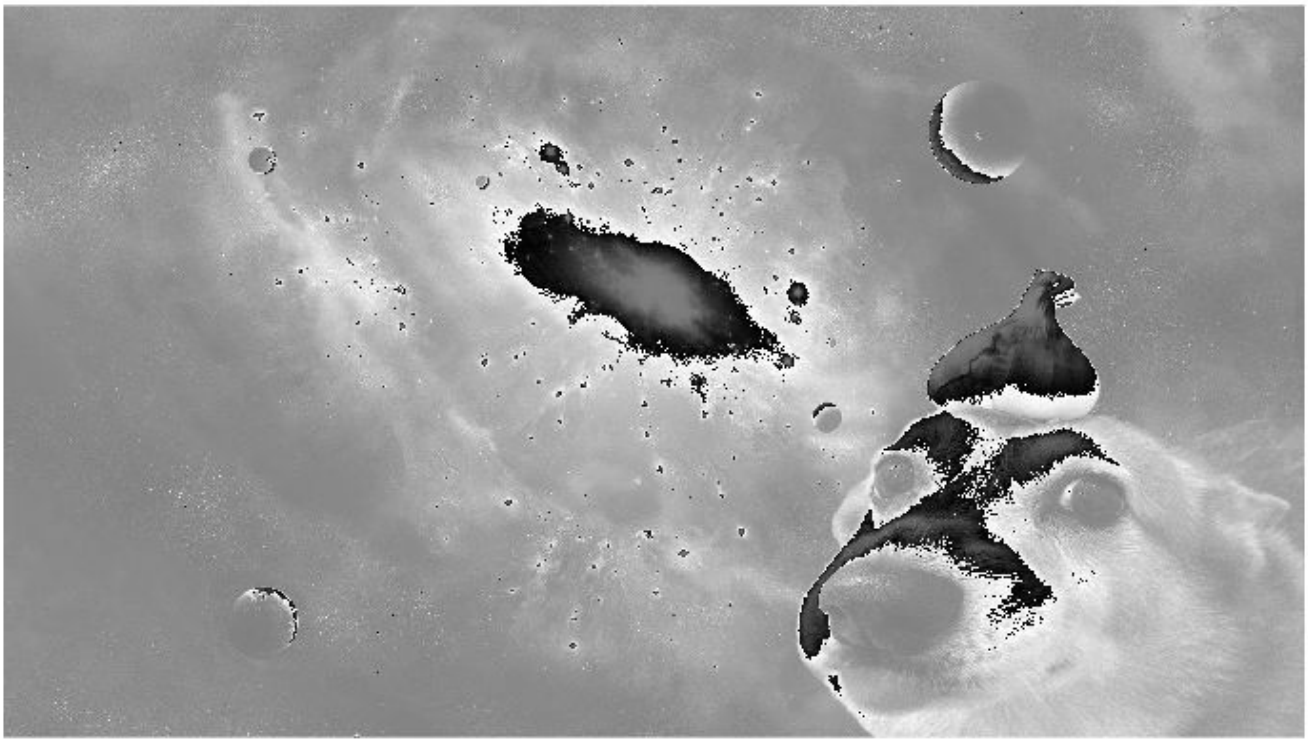
```
HSIhue = uint8(HSI(:, :, 1));
HSIsat = uint8(HSI(:, :, 2));
HSIint = uint8(HSI(:, :, 3));

LUT = uint8(zeros([1 256]));
LUT(1:128) = Ii(1:128) + 128;
LUT(129:256) = Ii(129:256) - 128;

plot(Ii, LUT), axis([ 1 256 1 256 ]), title('LUT Applied'), xlabel('Input intensity'), ylabel('
```



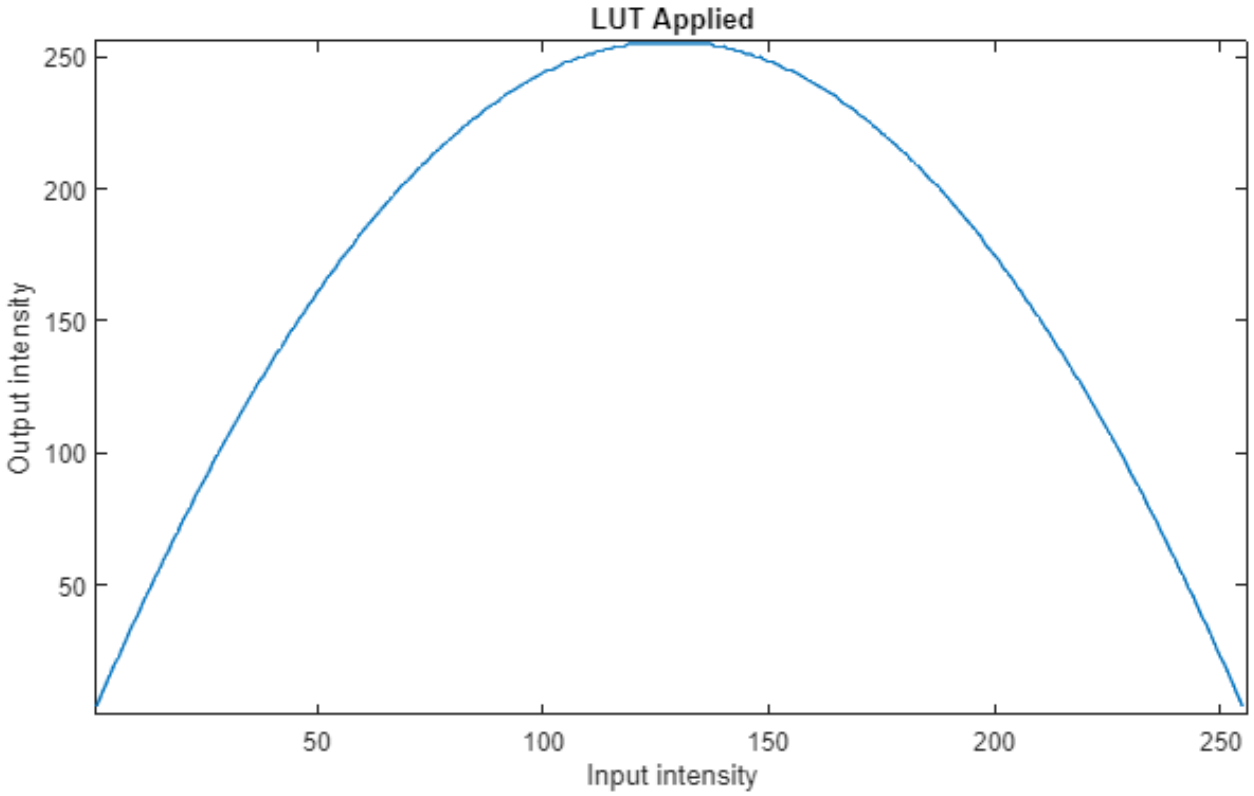

```
imshow(intlut(HSIhue, LUT));
```

```
imshow(intlut(HSIsat, LUT));
```

```
imshow(intlut(HSIint, LUT));
```



```
LUT = -(1 / 64) * (Ii .^ 2) + (4 * Ii);
LUT = uint8(LUT);

plot(Ii, LUT), axis([ 1 256 1 256 ]), title('LUT Applied'), xlabel('Input intensity'), ylabel('
```

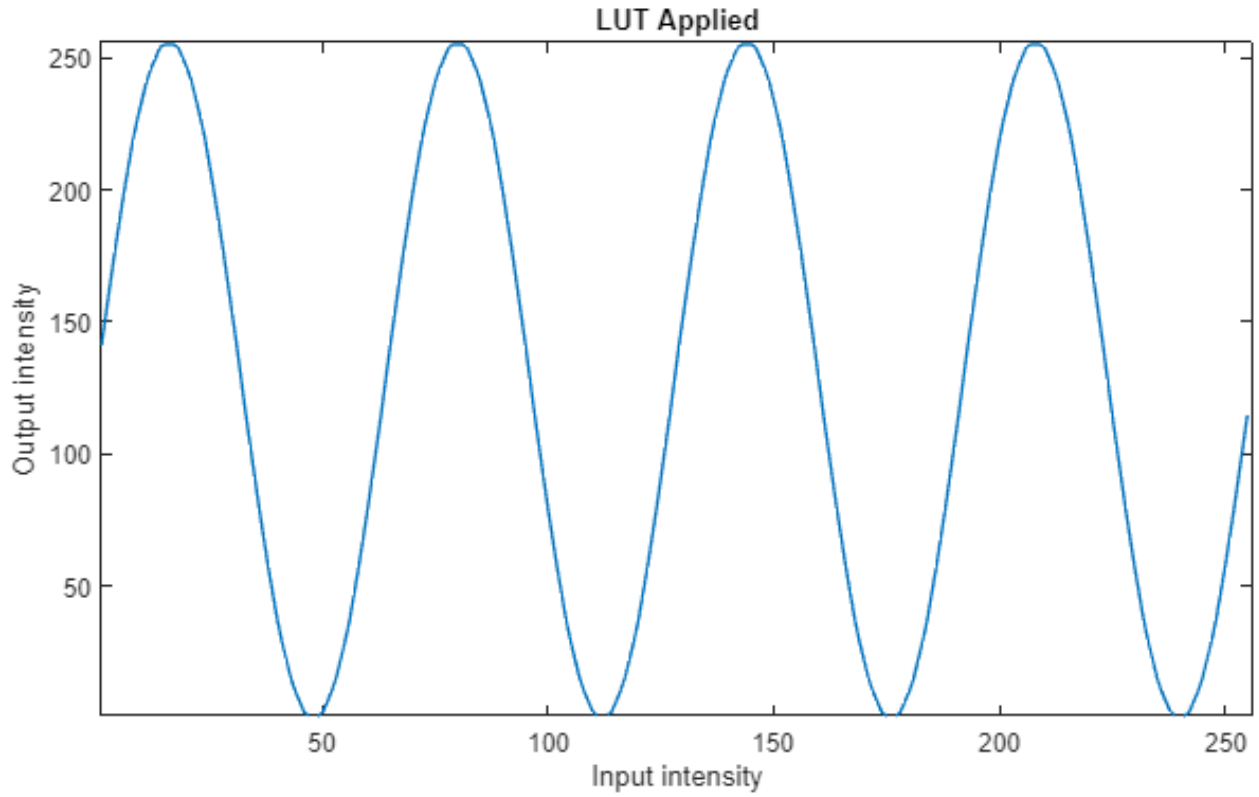



```
imshow(intlut(HSIhue, LUT));
```

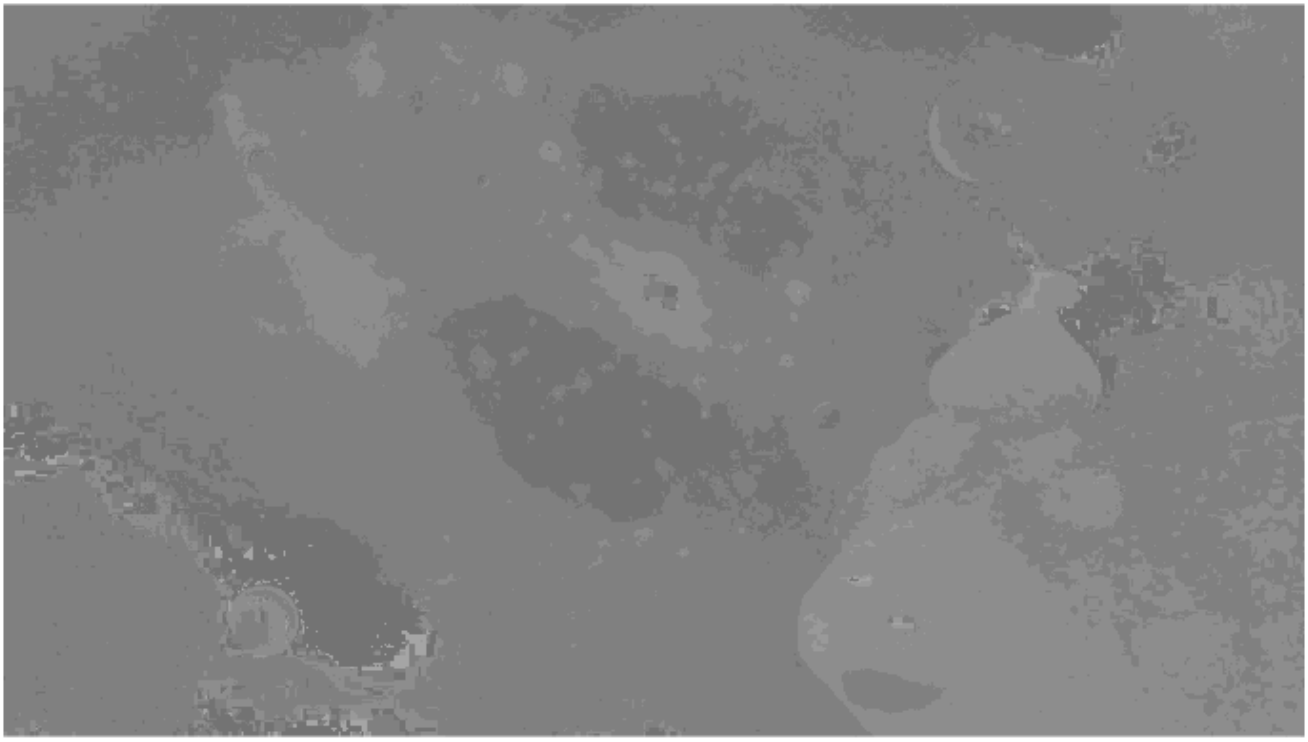
```
imshow(intlut(HSIsat, LUT));
```

```
imshow(intlut(HSIint, LUT));
```

```
LUT = 128 * sin((pi / 32) * Ii) + 128;
LUT = uint8(LUT);

plot(Ii, LUT), axis([ 1 256 1 256 ]), title('LUT Applied'), xlabel('Input intensity'), ylabel('
```

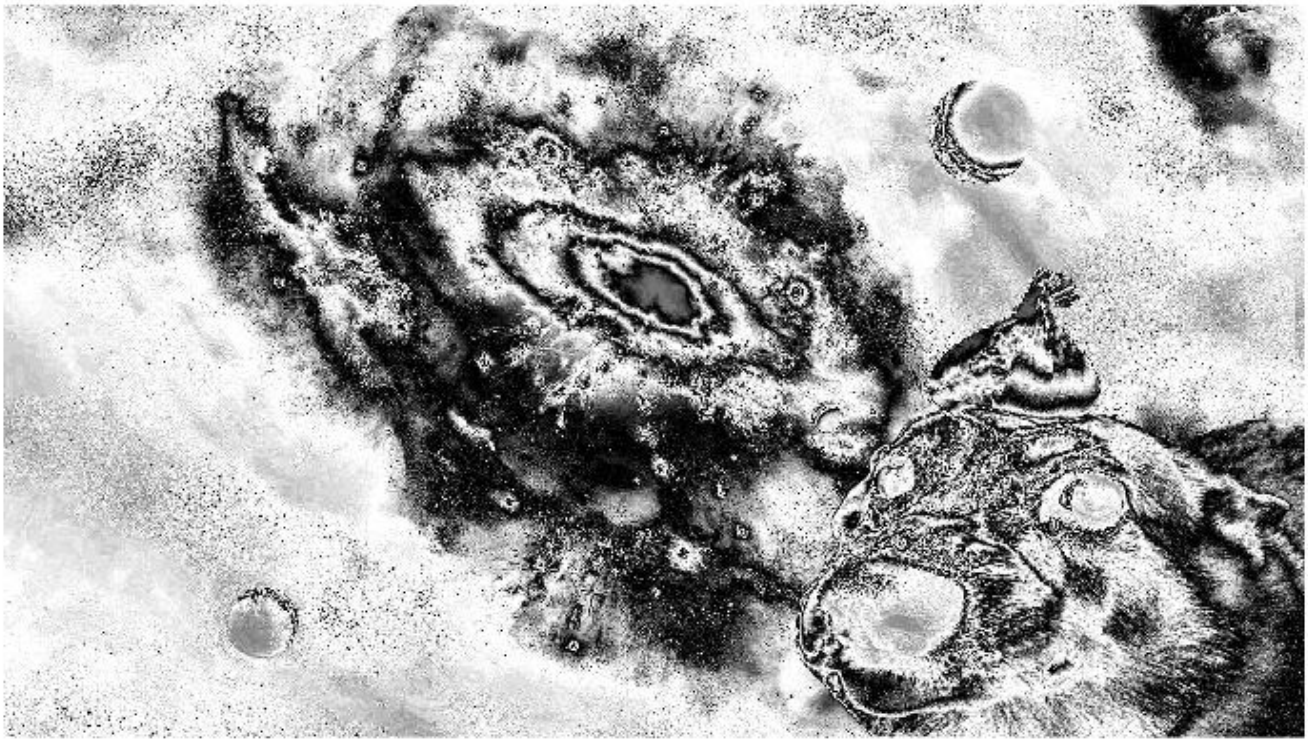



```
imshow(intlut(HSIhue, LUT));
```

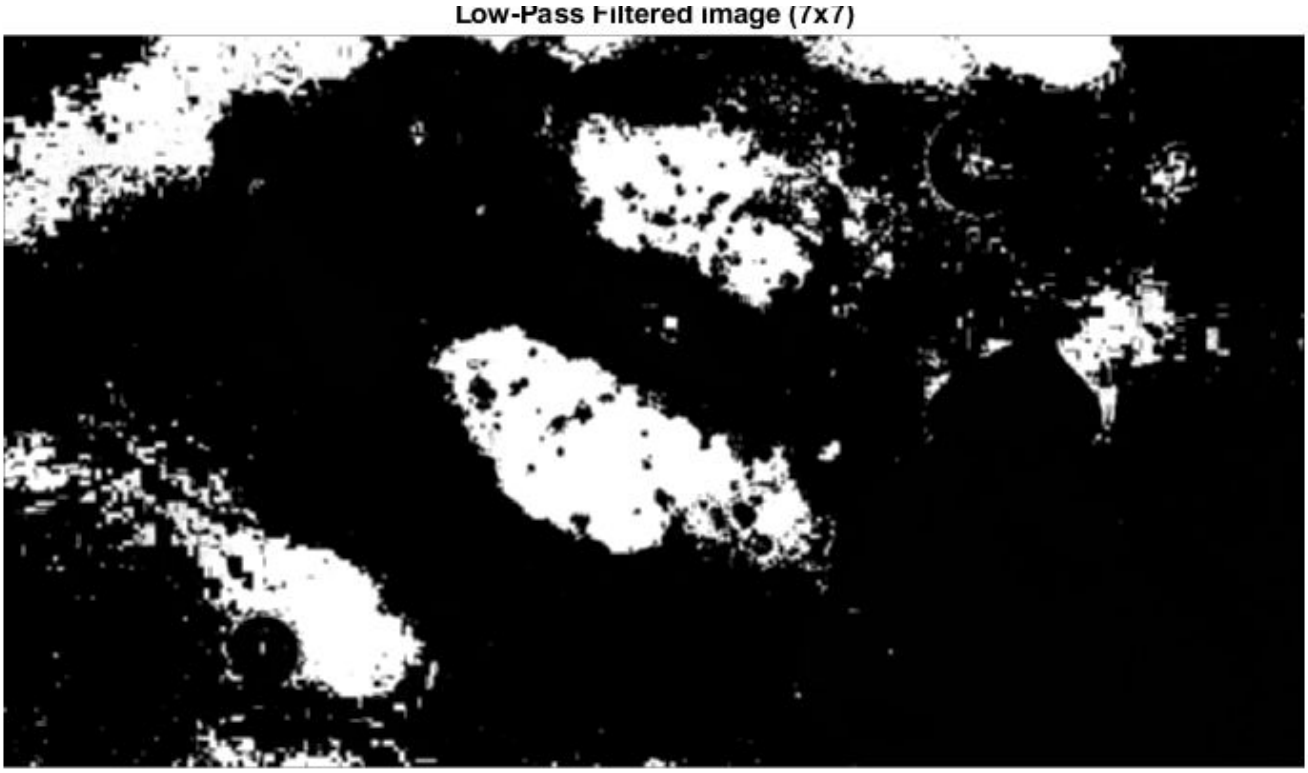


```
imshow(intlut(HSIsat, LUT));
```

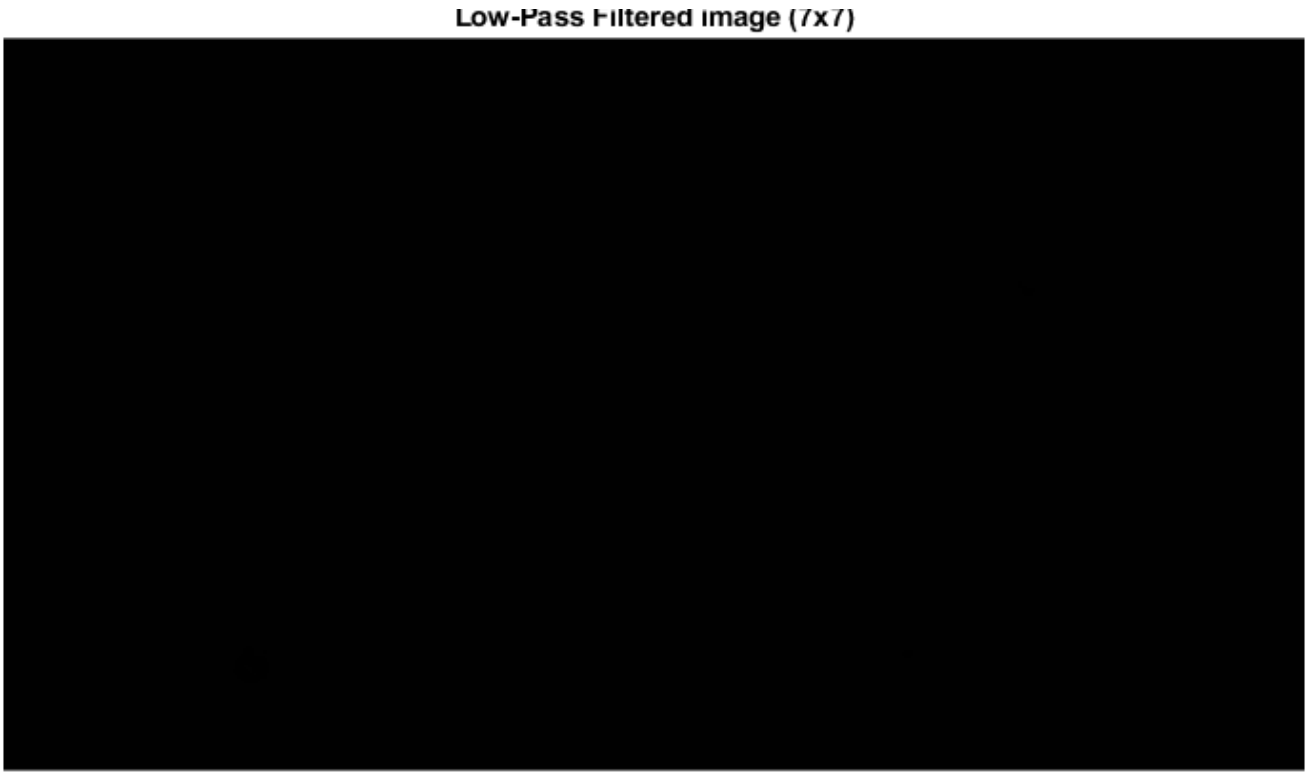
```
imshow(intlut(HSIint, LUT));
```



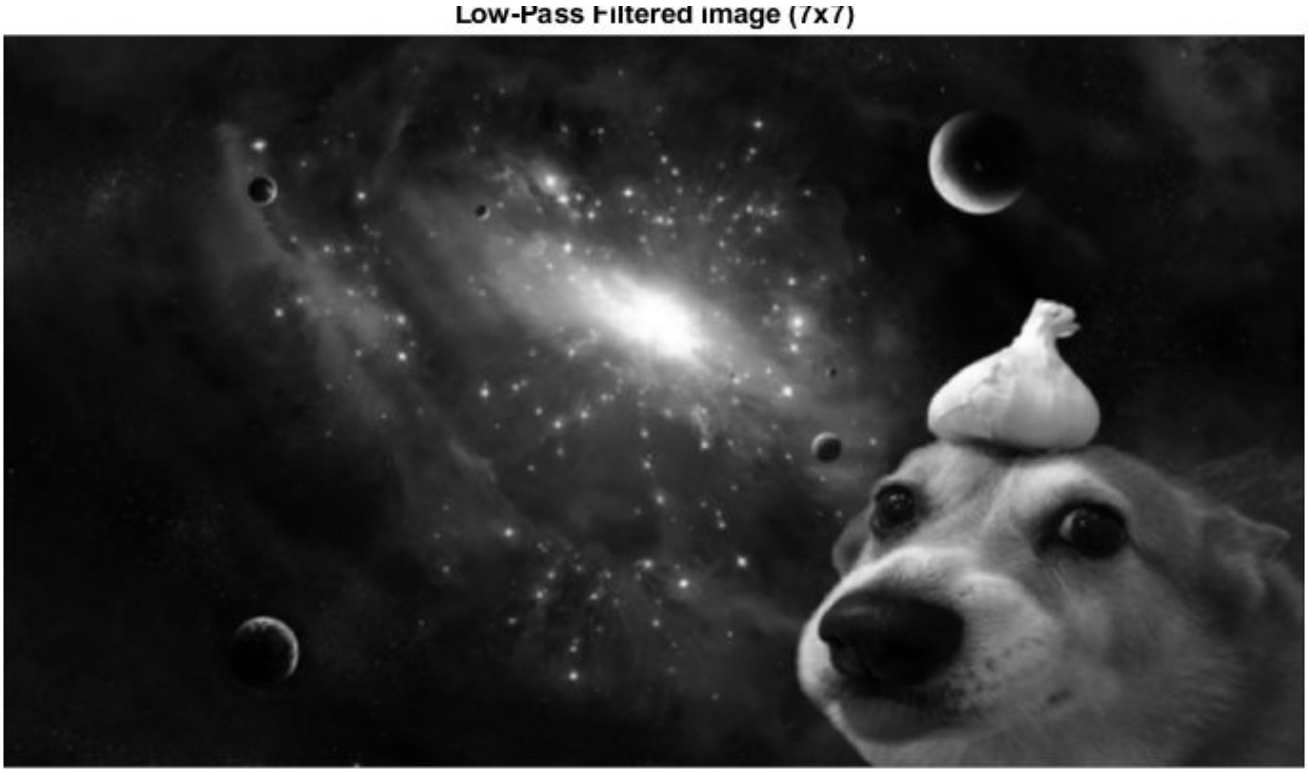
```
Lowpass7x7 = fspecial('average', 7);
Kernel = Lowpass7x7;

HSIhueFiltered = imfilter(HSIhue, Kernel, 'conv');
HSIsatFiltered = imfilter(HSIsat, Kernel, 'conv');
HSIintFiltered = imfilter(HSIint, Kernel, 'conv');

imshow(HSIhueFiltered), title('Low-Pass Filtered image (7x7)')
```



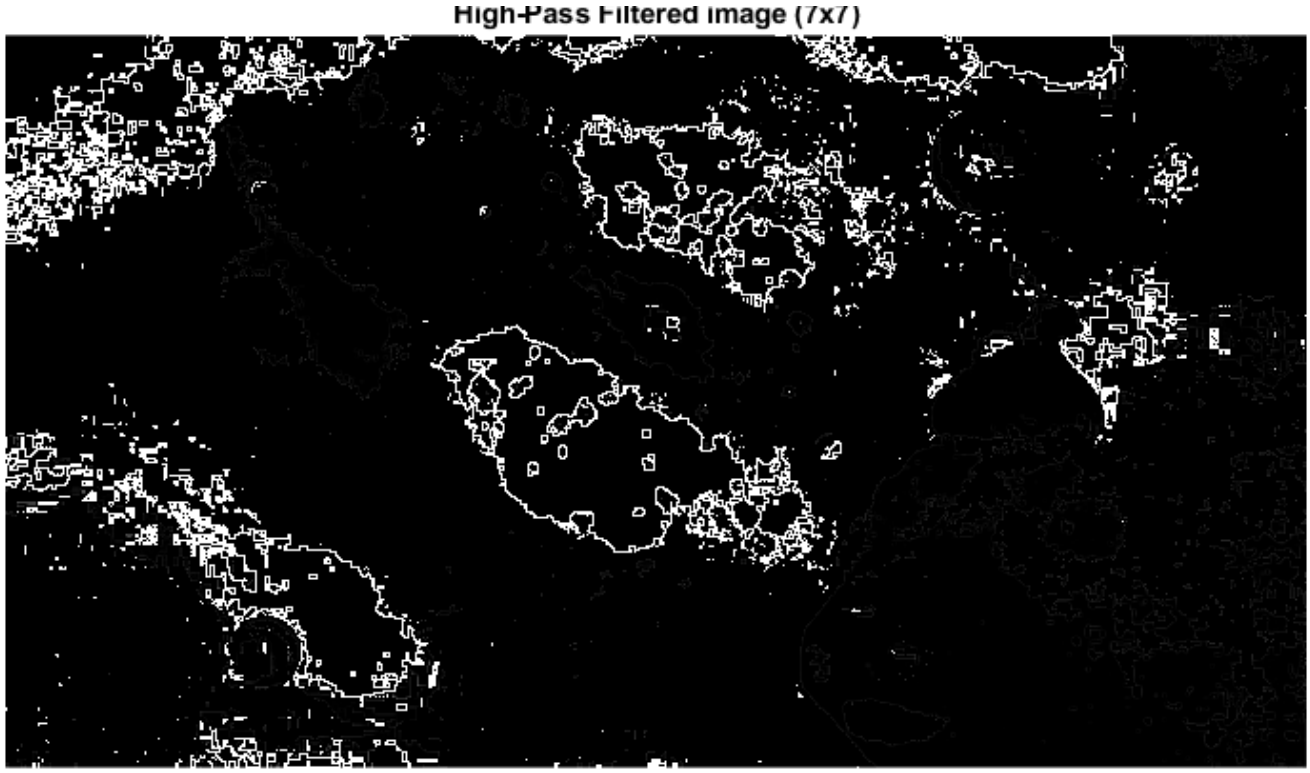

```
imshow(HSIsatFiltered), title('Low-Pass Filtered image (7x7)')
```



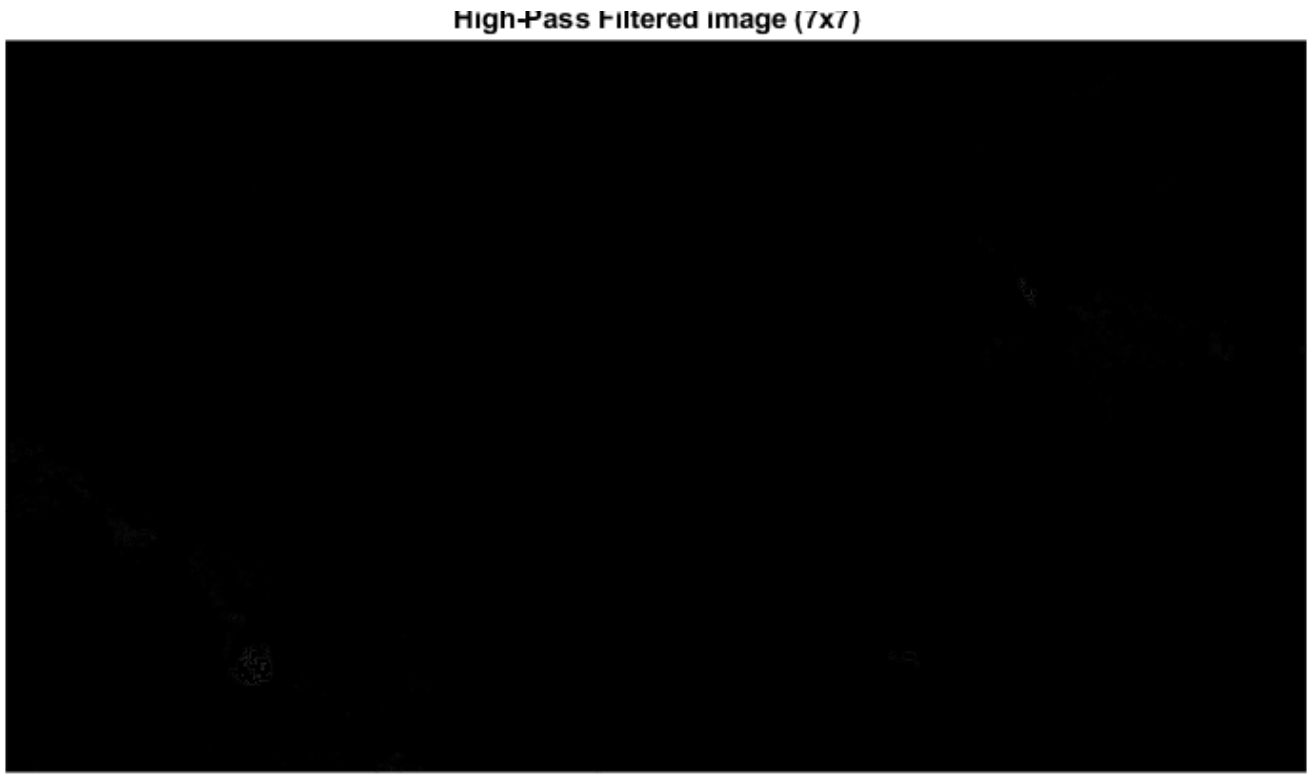

```
imshow(HSIintFiltered), title('Low-Pass Filtered image (7x7)')
```



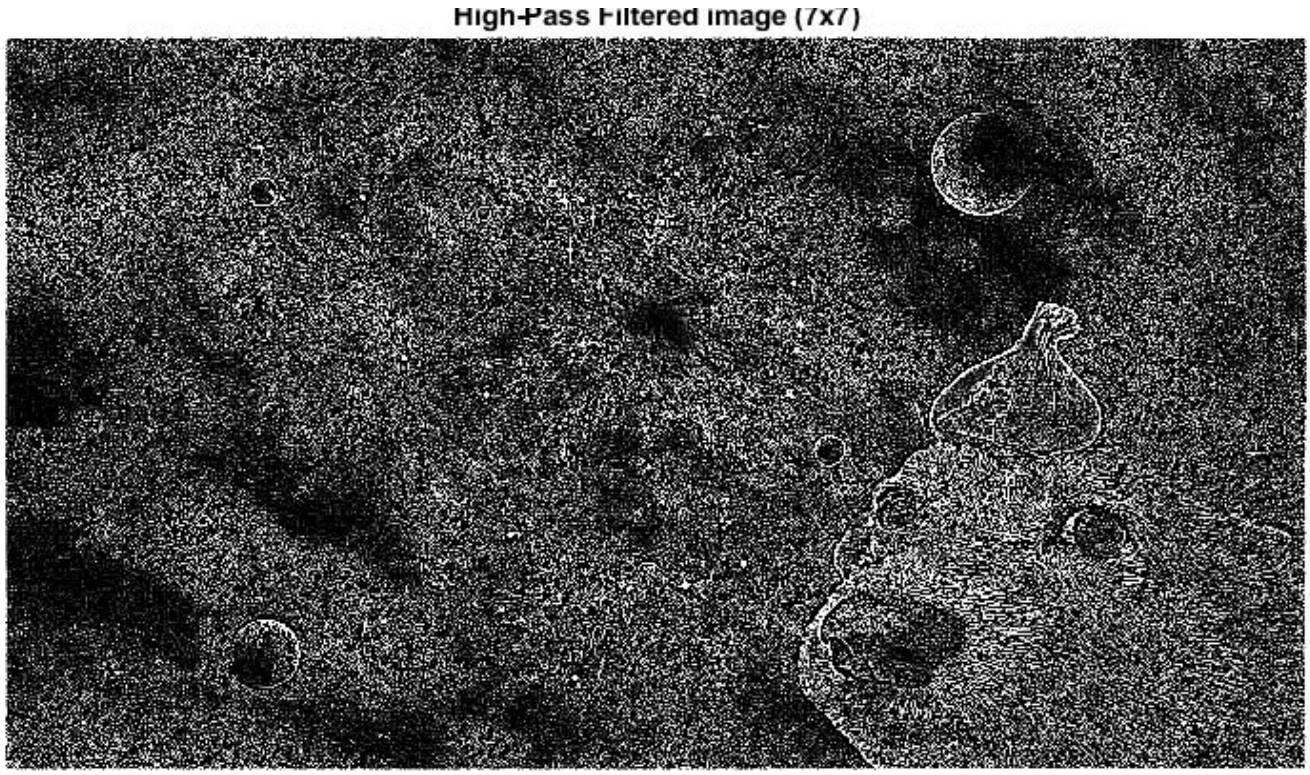
```
Highpass7x7 = [ -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, 48, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1;
                -1, -1, -1, -1, -1, -1, -1 ];
Kernel = Highpass7x7;

HSIhueFiltered = imfilter(HSIhue, Kernel, 'conv');
HSIsatFiltered = imfilter(HSIsat, Kernel, 'conv');
HSIintFiltered = imfilter(HSIint, Kernel, 'conv');

imshow(HSIhueFiltered), title('High-Pass Filtered image (7x7)')
```




```
imshow(HSIsatFiltered), title('High-Pass Filtered image (7x7)')
```



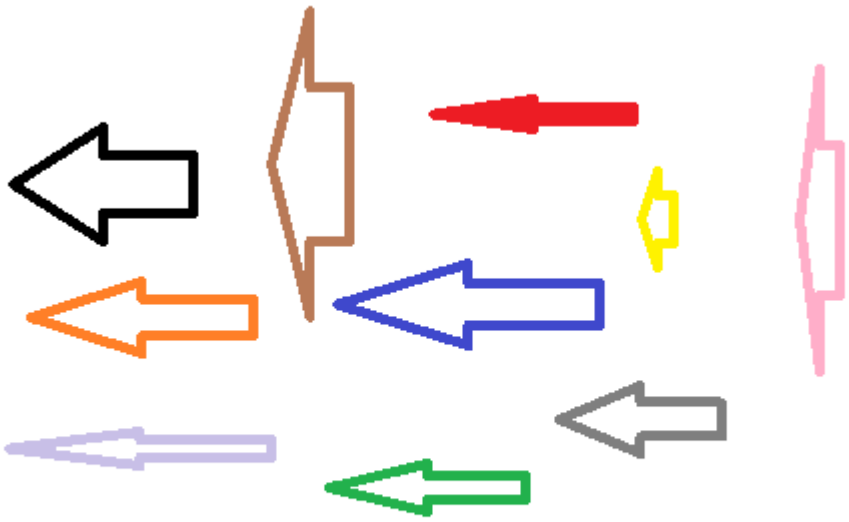

```
imshow(HSIintFiltered), title('High-Pass Filtered image (7x7)')
```



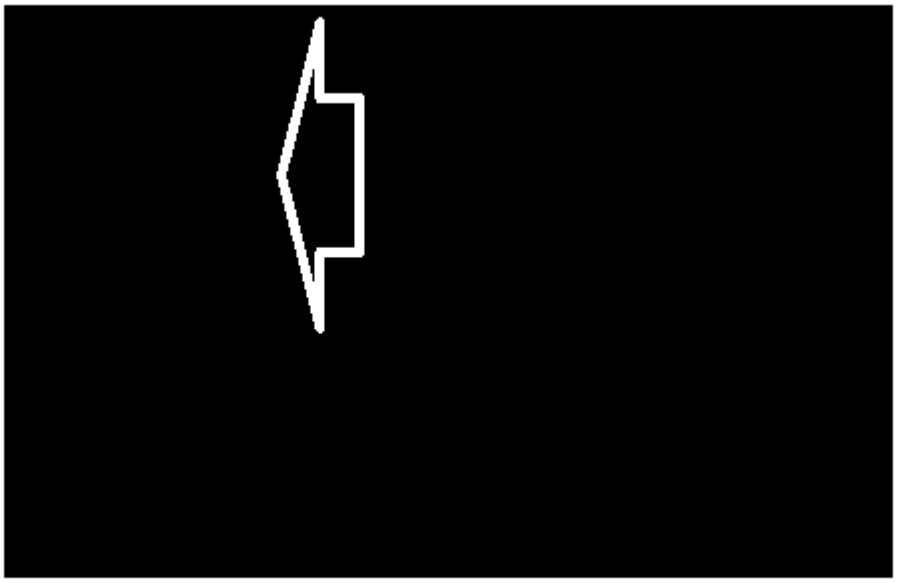
The results don't match up to what I expected them to be. I get the feeling my HSI values are being calculated incorrectly (especially saturation) but I don't know what to fix it. One application where HSI would be preferred over RGB is when the human percetion is taken into account. HSI also allows for easier manipulation of severy lighting-related effects like brightnes or saturation.

Part 5: Simple Color Detection

```
syn = imread("syn.png");
imshow(syn)
```



```


color = [185, 122, 87]; % brown arrow

imX = zeros(size(syn));
for i = 1:size(syn, 1)
    for j = 1:size(syn, 2)
        if (syn(i, j, :) ~= color)
            imX(i, j, :) = 0;
        else
            imX(i, j, :) = color;
        end
    end
end
imshow(imX)
```

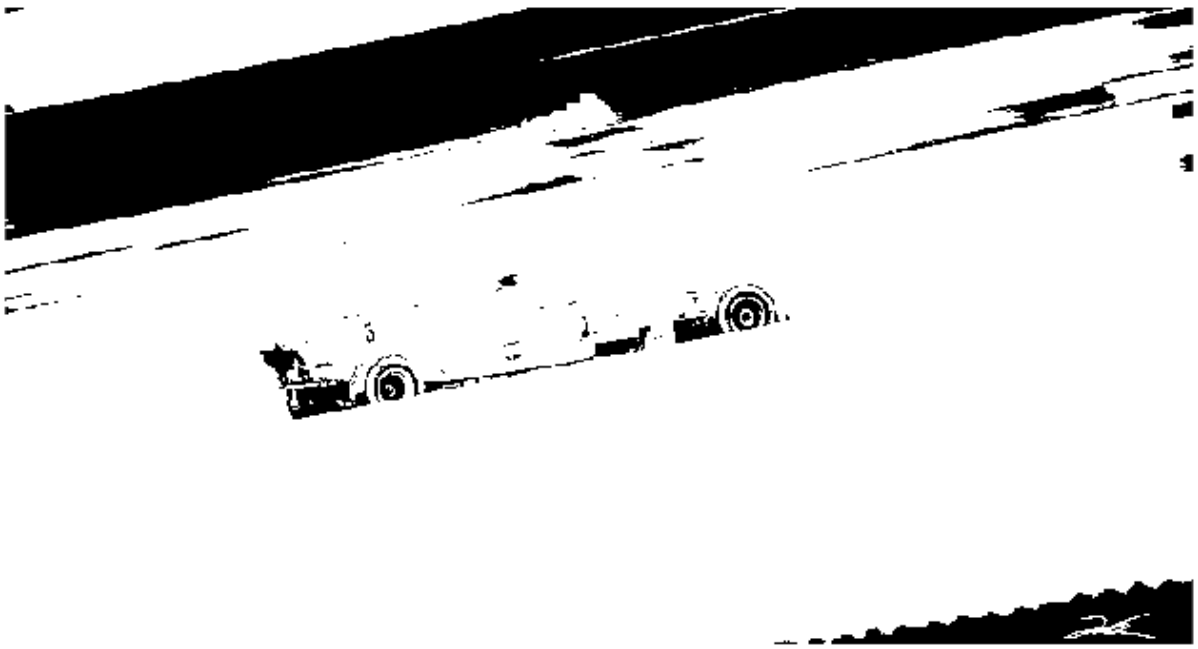


```
imX = zeros(size(im));
color = [185, 122, 87]; % brown arrow
for i = 1:size(im, 1)
    for j = 1:size(im, 2)
        if (im(i, j, :) ~= color)
            imX(i, j, :) = 0;
        else
            imX(i, j, :) = color;
        end
    end
end
imshow(imX)
```

My algorithm mostly works but has a weird bug where it doesn't display the color you tell it to. Through manual verification I can tell that is is functioning. My algorithm visits every pixel and checks it agasinst the desired RGB value. If the colors do not match, the pixel is turned black.

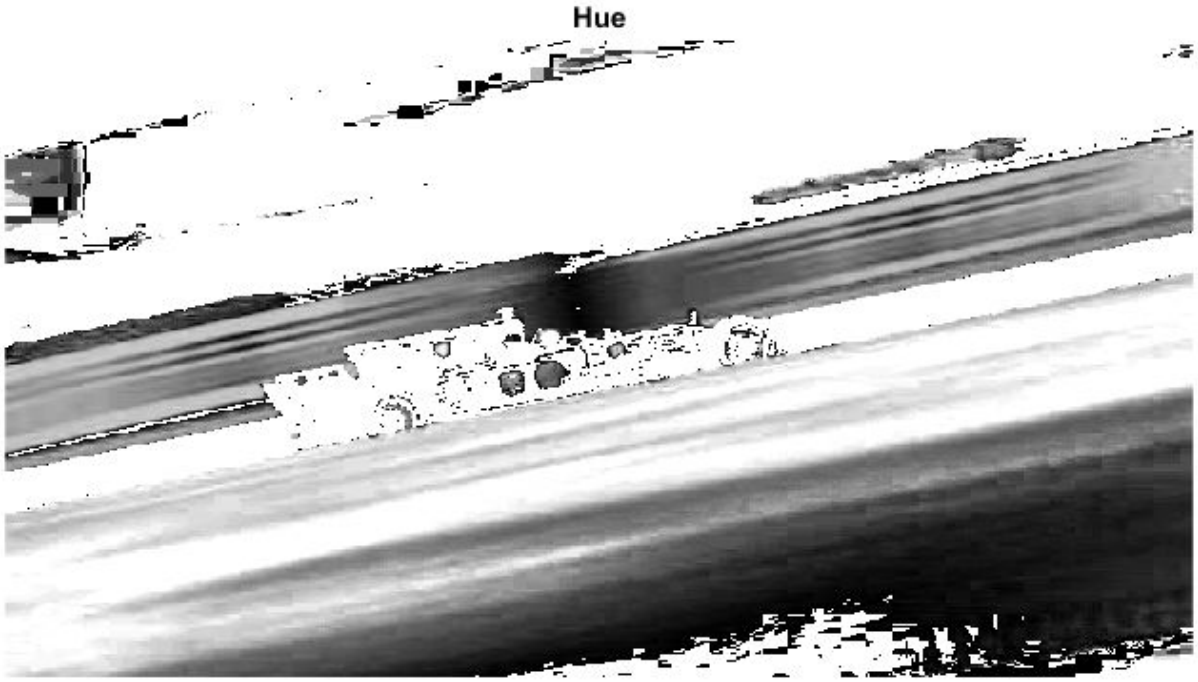
```
f1 = imread("f1.jpg");
imshow(f1)
```

```
imX = zeros(size(f1));
color = [248, 38, 57]; % exact
threshhold = 50;
for i = 1:size(f1, 1)
    for j = 1:size(f1, 2)
        if (((f1(i, j, :) - threshhold) .^ 2 ) <= 10)
            imX(i, j, :) = 0;
        else
            imX(i, j, :) = color;
        end
    end
end
imshow(imX)
```

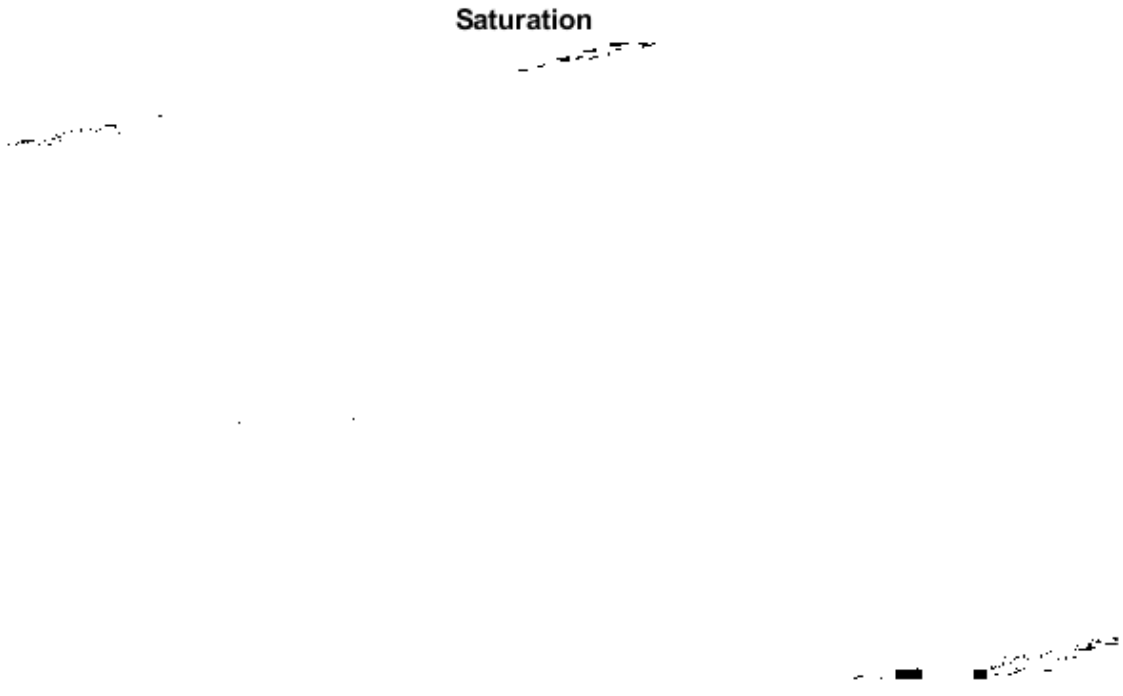


```
imRed = f1;
imRed(:, :, [2, 3]) = 0;
imRedGrey = f1(:, :, 1);

imGreen = im;
imGreen(:, :, [1, 3]) = 0;
imGreenGrey = f1(:, :, 2);

imBlue = im;
imBlue(:, :, [1, 2]) = 0;
imBlueGrey = f1(:, :, 3);

imRedGreyD = double(imRedGrey);
imGreenGreyD = double(imGreenGrey);
imBlueGreyD = double(imBlueGrey);

intensity = mean(f1, 3);

saturation = 1 - (3 ./ (imRedGrey + imGreenGrey + imBlueGrey)) .* min(f1, [], 3);

hue = acos((0.5 * ((imRedGreyD - imGreenGreyD) + (imRedGreyD - imBlueGreyD))) ./ (sqrt((imRedGr
hue(imBlueGreyD > imGreenGreyD) = 360 - hue(imBlueGreyD > imGreenGreyD);

HSI = zeros(size(f1));
HSI(:, :, 1) = (hue);
HSI(:, :, 2) = (saturation);
```

```
HSI(:, :, 3) = uint8(intensity);

imshow(HSI(:, :, 1)), title('Hue');
```




```
imshow(HSI(:, :, 2)), title('Saturation');
```



```
imshow(HSI(:, :, 3)), title('Intensity');
```