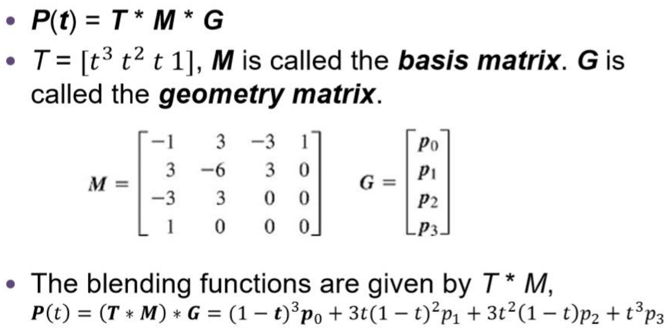
Multimedia Assignment4

Q1. Bézier curve

※Implementation

I use the formula in slide to create matrix M, vector T and vector P .



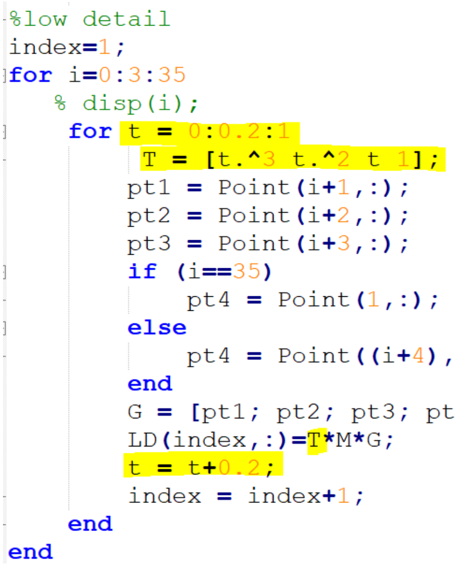
- ***P***(***t***) = ***T*** * ***M*** * ***G***
- *T* = $[t^3\ t^2\ t\ 1]$, ***M*** is called the ***basis matrix***. ***G*** is called the ***geometry matrix***.

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

- The blending functions are given by *T* * *M*,

$$\boldsymbol{P}(t) = (\boldsymbol{T} * \boldsymbol{M}) * \boldsymbol{G} = (1-\boldsymbol{t})^3\boldsymbol{p}_0 + 3t(1-t)^2p_1 + 3t^2(1-t)p_2 + t^3p_3$$

According to different level of detail, to change vector T element t.



```
%low detail
index=1;
for i=0:3:35
   % disp(i);
    for t = 0:0.2:1
        T = [t.^3 t.^2 t 1];
        pt1 = Point(i+1,:);
        pt2 = Point(i+2,:);
        pt3 = Point(i+3,:);
        if (i==35)
            pt4 = Point(1,:);
        else
            pt4 = Point((i+4),:);
        end
        G = [pt1; pt2; pt3; pt4;];
        LD(index,:)=T*M*G;
        t = t+0.2;
        index = index+1;
    end
end
```

I use two for loops to divide 37 points by 4(it will overlap in the first number, which is 1~4,4~7,7~10,10~13…..)so that we can get 12 parts of numbers.

Notice that if the point is the last, just link it to the first so that we can get a circle.

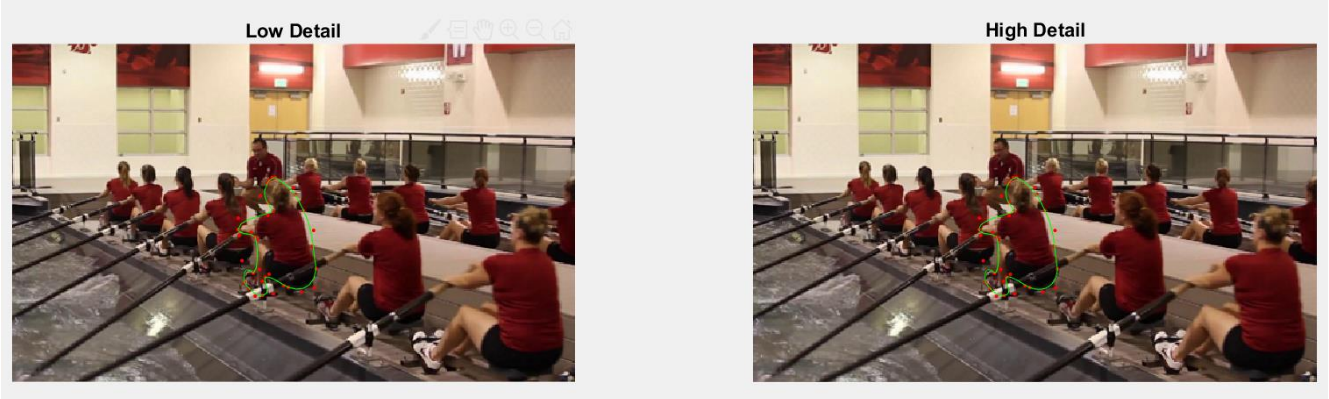
So I write: if i==35 pt4 = Point(1,:); else pt4 = Point((i+4),:);

As for scale picture by 4, I use function `img_4 = imresize(img, 4, 'nearest');` so that I don't have to implement by myself.

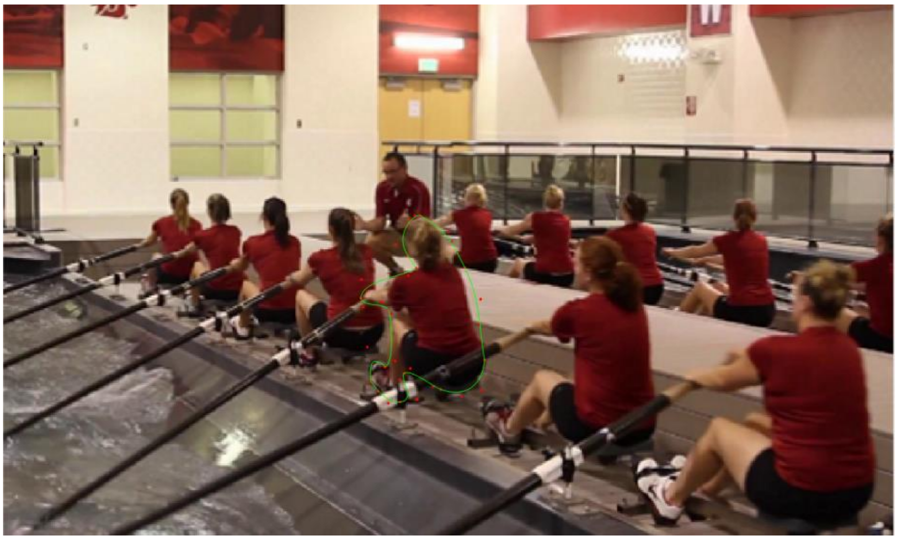
Also, multiply points by 4, which is done by TA `Point = Point .* 4;`

Then , do everything as same as question a will solve this question.

※Result display & Discussion



Scaled processing:



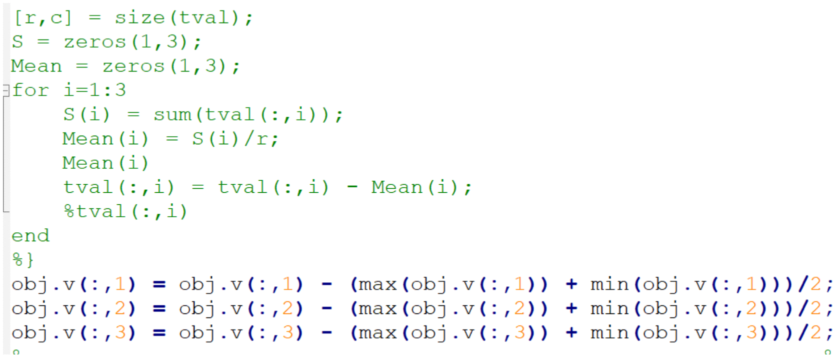
I can not tell which picture is low detail and which is not since they look so alike. However, it should be more smooth when using high detail setting. I think that if two of the setting are much more different, maybe it can be seperated easier. As for second question, I found out that the picture is much more jagged than unscaled picture. However, the curve is the same since we also scaled it for 4 times. There would not be any position differences comparing to unscaled one. So size are different , curve still the same (just bigger).

Q2. 3D Models

※Implementation

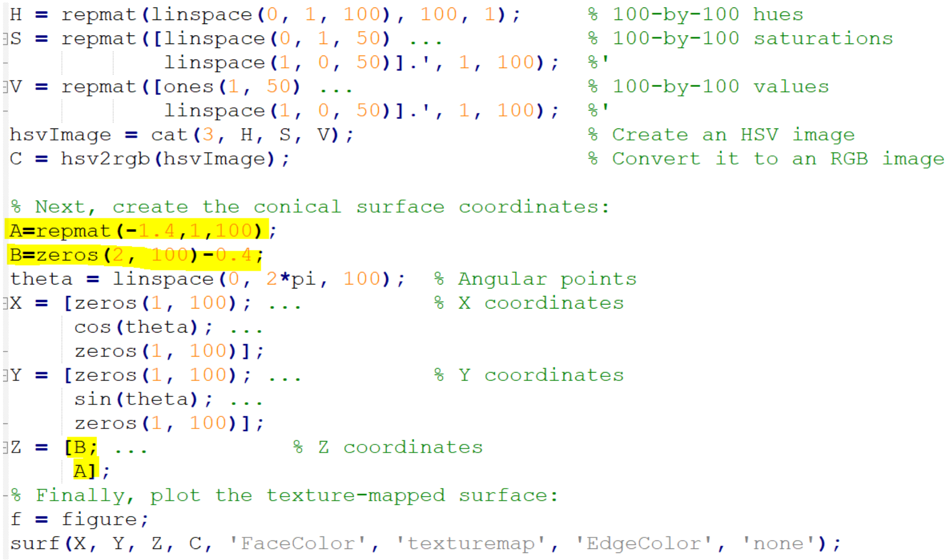
This question is more complicated, here is my implementation:

In the beginning, I try to shift the center index , but I just stupidly used for loops to do it. Reminded by classmate , I change to a more clever and fast way:



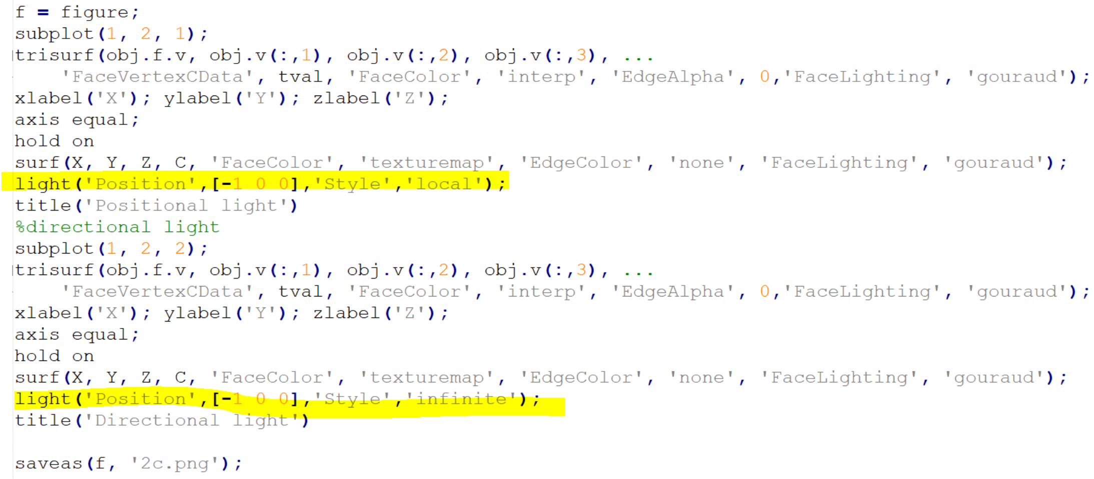
```
[r,c] = size(tval);
S = zeros(1,3);
Mean = zeros(1,3);
for i=1:3
    S(i) = sum(tval(:,i));
    Mean(i) = S(i)/r;
    Mean(i)
    tval(:,i) = tval(:,i) - Mean(i);
    %tval(:,i)
end
%}
obj.v(:,1) = obj.v(:,1) - (max(obj.v(:,1)) + min(obj.v(:,1)))/2;
obj.v(:,2) = obj.v(:,2) - (max(obj.v(:,2)) + min(obj.v(:,2)))/2;
obj.v(:,3) = obj.v(:,3) - (max(obj.v(:,3)) + min(obj.v(:,3)))/2;
```

As for HSV , just change Z index and other scale such as hieght, radius and so on.



```
H = repmat(linspace(0, 1, 100), 100, 1);     % 100-by-100 hues
S = repmat([linspace(0, 1, 50) ...           % 100-by-100 saturations
            linspace(1, 0, 50)].', 1, 100);  %'
V = repmat([ones(1, 50) ...                  % 100-by-100 values
            linspace(1, 0, 50)].', 1, 100);  %'
hsvImage = cat(3, H, S, V);                  % Create an HSV image
C = hsv2rgb(hsvImage);                       % Convert it to an RGB image

% Next, create the conical surface coordinates:
A=repmat(-1.4,1,100);
B=zeros(2, 100)-0.4;
theta = linspace(0, 2*pi, 100);  % Angular points
X = [zeros(1, 100); ...          % X coordinates
     cos(theta); ...
     zeros(1, 100)];
Y = [zeros(1, 100); ...          % Y coordinates
     sin(theta); ...
     zeros(1, 100)];
Z = [B; ...             % Z coordinates
     A];
% Finally, plot the texture-mapped surface:
f = figure;
surf(X, Y, Z, C, 'FaceColor', 'texturemap', 'EdgeColor', 'none');
```

As for changing different lighting , just change this line.

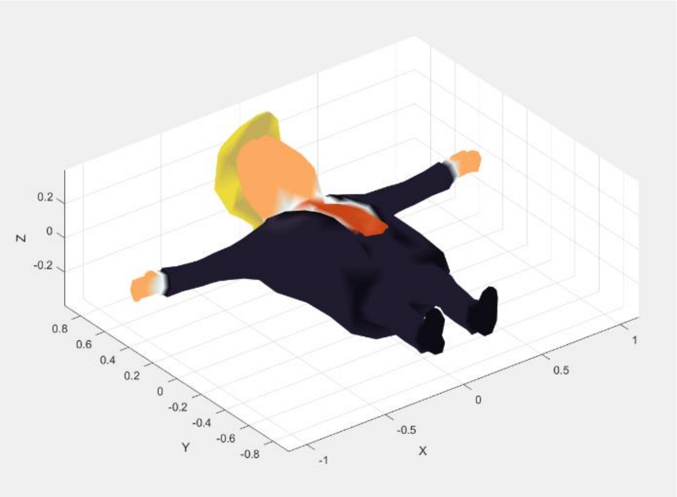


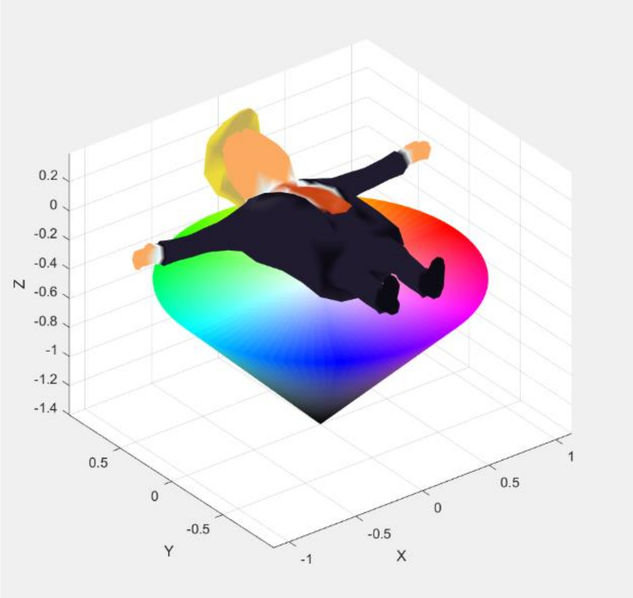
```
f = figure;
subplot(1, 2, 1);
trisurf(obj.f.v, obj.v(:,1), obj.v(:,2), obj.v(:,3), ...
    'FaceVertexCData', tval, 'FaceColor', 'interp', 'EdgeAlpha', 0,'FaceLighting', 'gouraud');
xlabel('X'); ylabel('Y'); zlabel('Z');
axis equal;
hold on
surf(X, Y, Z, C, 'FaceColor', 'texturemap', 'EdgeColor', 'none', 'FaceLighting', 'gouraud');
light('Position',[-1 0 0],'Style','local');
title('Positional light')
%directional light
subplot(1, 2, 2);
trisurf(obj.f.v, obj.v(:,1), obj.v(:,2), obj.v(:,3), ...
    'FaceVertexCData', tval, 'FaceColor', 'interp', 'EdgeAlpha', 0,'FaceLighting', 'gouraud');
xlabel('X'); ylabel('Y'); zlabel('Z');
axis equal;
hold on
surf(X, Y, Z, C, 'FaceColor', 'texturemap', 'EdgeColor', 'none', 'FaceLighting', 'gouraud');
light('Position',[-1 0 0],'Style','infinite');
title('Directional light')

saveas(f, '2c.png');
```

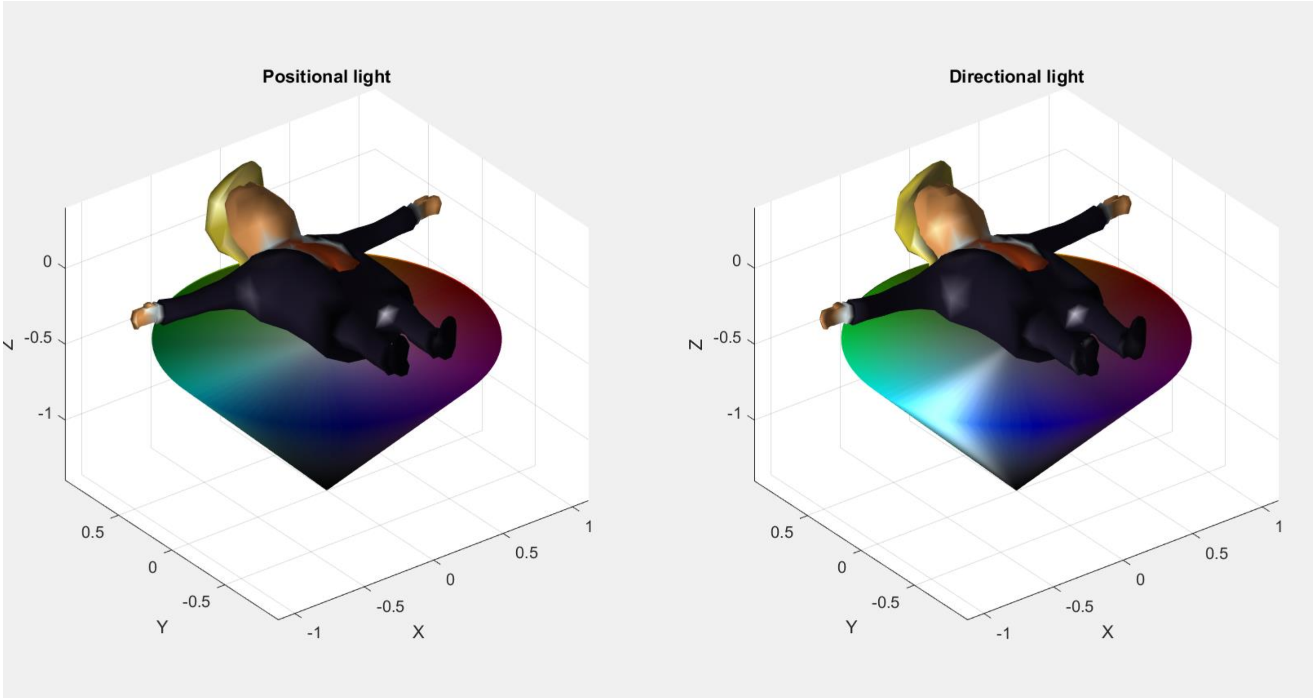
Finally, to change different strength of ambient , diffuse, specular ,just modify their scale. Very simple!

```
first1= surf(X, Y, Z, C, 'FaceColor', 'texturemap', 'Edg
light('Position',[0 0 1], 'Style', 'infinite');
lighting phong;
first.AmbientStrength = 1;
first.DiffuseStrength = 0;
first.SpecularStrength = 0;
first1.AmbientStrength = 1;
first1.DiffuseStrength = 0;
first1.SpecularStrength = 0;
```

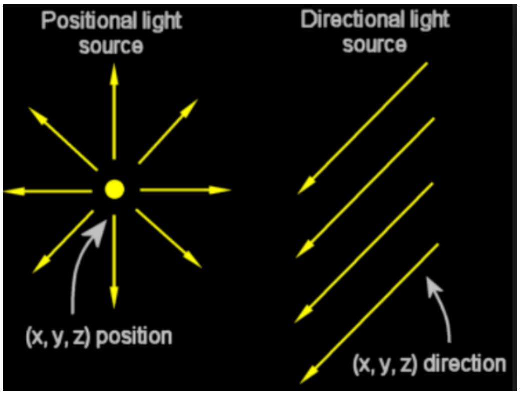
※Result display & Discussion





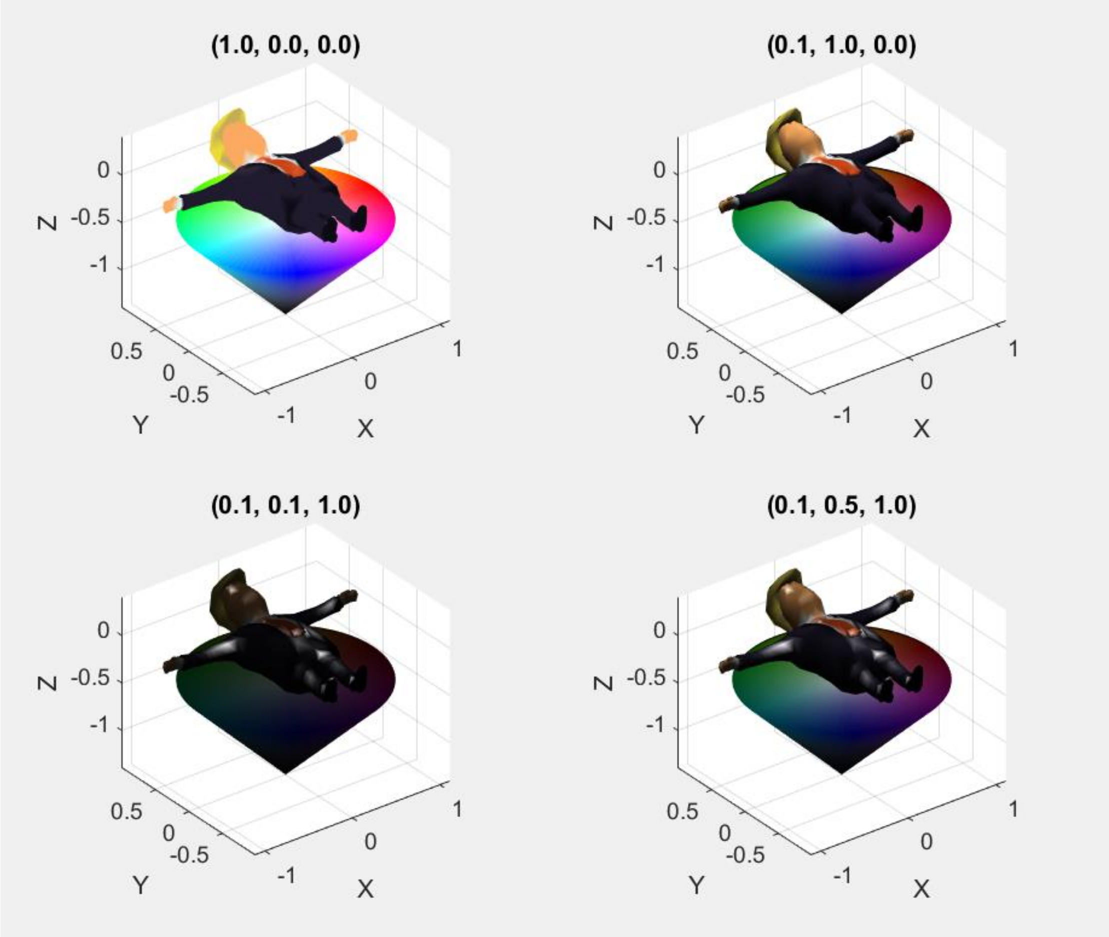


Use one picture to demonstrate the difference between position light and directional light.



If use position light, it will get darker since the lighting source is small , many places will not get the light. Since my light source is set on (-1,0,0) , so you can see the left part of HSV cone is lighter than other parts in directional light.

Since positional light source is "local" it can't be the same lightness as directional light source .



Ambient 是指環境光

So we can conclude that if ambient is 1, environment light will be very bright, you can see (1,0,0), the trump is in clean fresh color, even if other two factors are 0. But if ambient is 0.1 , it will become darker so that other effect will influence the object more noticeable .

Diffuse 是指物體在光源下的實際表現

Comparing (0.1,0.1,1.0) and (0.1,0.5,1.0) ,we can find out that in same ambient and specular , when diffuse is 0.1 , it is almost completely dark. In this case, ambient is very little so not so much light can reflect.

Specular 是當入射光對物體產生全反射的情形

For picture (0.1,1.0,0.0) ,since specular is zero , there's no "light spot" on trump's head. Since no light will reflect. So I conclude that specular influence the existence of reflection spot.

Conclusion: If ambient not big enough , object will be darker however you change diffuse strength or specular strength.
Diffuse just do reflection work , so it can't as powerful as ambient.
If specular is big enough, it will have light spot on object, the surface of the object will have metal like luster.

Reference:
HSV
https://stackoverflow.com/questions/3339692/modeling-hsv-color-space-in-matlab?fbclid=IwAR27Sv3H6m2u2oYCHT3OEjoW2j3Slgc-LJOp_FNNKST8ow3KVMMRyw6s70A
saving files
https://www.mathworks.com/matlabcentral/answers/264639-saving-a-figure-with-several-subplots-on-to-a-file
light strength
https://www.csie.ntu.edu.tw/~r89004/hive/perpixel/page_2.html