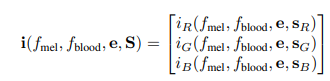
**Implementation Details**

In both the papers “A Biophysical 3D Morphable Model of Face Appearance” ([1](https://openaccess.thecvf.com/content_ICCV_2017_workshops/papers/w16/Alotaibi_A_Biophysical_3D_ICCV_2017_paper.pdf)) and “Decomposing Multispectral Face Images into Diffuse and Specular Shading and Biophysical Parameters” ([2](https://ieeexplore.ieee.org/abstract/document/8803369/)), the biophysical model for skin colouration has been presented.

In 1, the model has been fitted using 3d face meshes and diffuse albedo maps of the face (which they captured using s lightstage). In 2, the fitting has been done using scene spectral radiance of the images. These images were acquired in an earlier experiment (mentioned [here](https://web.stanford.edu/group/scien/cgi-bin/scien/jfsite/Papers/ImageCapture/2013_HyperspectralImagingDatabase.pdf)) using hyperspectral sensors. After fitting, both papers used the parameters in the following equation to reconstruct the image (get the raw colour value):



where i is the colour value, fmel , fblood are the biophysical parameters, e is the spectral power distribution of the illuminant and S is the spectral sensitivity of the camera used. Since we only need an estimate of the parameters, may not have the additional information (like 3d face mesh, diffuse albedo map, spectral radiance information of each pixel) and also aren’t concerned with reconstruction of the image, I decided to use the above equation for fitting the model and getting the estimates of fmel and fblood.

I have added the fitting.m, estimation.m and reconstruct.m to the already developed [Skin Model code](https://github.com/ssma502/SkinModel) (by one of the authors of the above papers).

In fitting.m, I have defined the function fitting in which the inputs are the parameters of fmel and fblood, model (which is a struct having all the different fixed parameters incorporated), T\_RAW2XYZ (a matrix that is used to convert the raw colour value to RGB space) and x (which is a matrix of S and e). The function takes these inputs and calculates the colour value of a single pixel (in RGB). The e (spectral power distribution of the illuminant) has been taken as illuminant A of CIE space and S is the mean camera spectral sensitivity from [this](https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.364.4764&rep=rep1&type=pdf). These values of e and S are standard values used in the papers and knowing these values for our particular setup would lead to more accurate results.

In estimation.m, I have imported all the required variables first. Then the image is read and a cropped portion of the image is taken for estimation. This is because the model is only supposed to work for skin pixels and will give erroneous values for the other regions of the image. Ideally, using face detection and/or skin segmentation on the image should give us the required image. After this a least square curve fitting is done on each of the pixels of the image one by one using the lsqcurvefit function. The image pixels are input to the function along with the fitting function and the estimates of fmel and fblood are found and stored in matrices. Here I want to mention that

1. I used this method for fitting since this was used in 2 (although for a slightly different equation)
2. I programmed it for running one pixel at a time (instead of maybe passing a whole image matrix and getting estimates) because of the statement from 2: “we solve the optimisation problem independently at each pixel”

Because of going one pixel at a time, the estimation.m takes a bit long to run. To somewhat reduce the time, I used parfor to parallelise the outer for loop. Maybe we can come up with some other methods to speed up the execution as well.

The reconstruct.m is just using the estimated parameter values to get the pixel values again. I wrote it simply to check whether the estimated parameters give back the actual pixel values. It is just for checking and doesn’t have any other use.