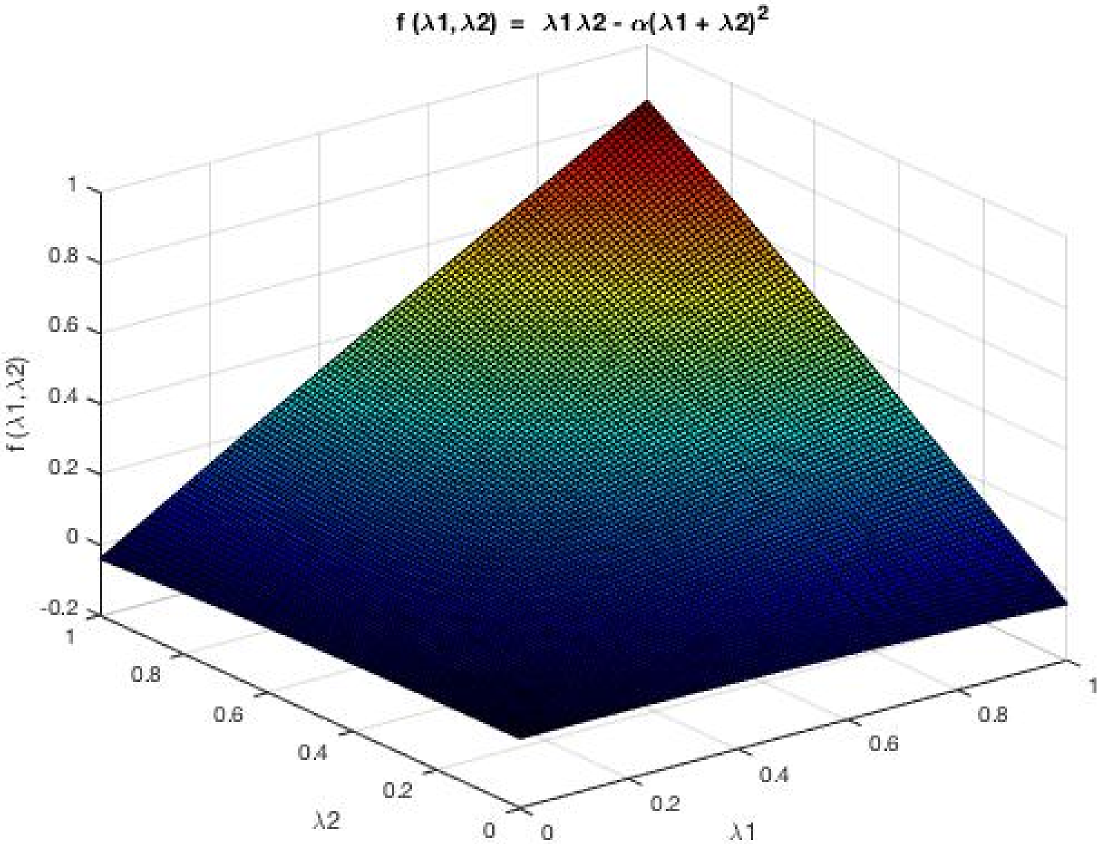
Computer Vision *Name: Abhay Doke, UID: 29552668*

HW05: Corner detection

Hand in via moodle at: `https://moodle.umass.edu/course/view.php?id=33024`. Remember that only PDF submissions are accepted. We encourage using LaTeX to produce your writeups. See `hw00.tex` for an example of how to do so. You can make a `.pdf` out of the `.tex` by running "`pdflatex hw00.tex`".

1. The Harris corner detector computes the corner-score of a pixel as $f(\lambda_1, \lambda_2) = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$, where λ_1 and λ_2 are the eigenvalues of the matrix M discussed in the class, and α is a parameter typically set $\in [0.04, 0.06]$. In Matlab compute f for $\alpha = 0.04$ as a function of λ_1 and λ_2. The result should be a 2D image where the intensity of a pixel (i, j) corresponds to $f(\lambda_1 = t_i, \lambda_2 = t_j)$. Uniformly sample a set of values of $t_i \in [0, 1]$. You may find the `meshgrid()` function in Matlab helpful for this. Use `colormap jet` to display this image.



2. Come up with another function of λ_1 and λ_2 that might work as a corner score function. This function should be small when both values are close to zero, or when one value is significantly higher than the other. Plot this function next to the previous one.

My function of λ_1 and λ_2 that might work as a corner function :

$$\frac{2}{(\frac{1}{\lambda_1}) + (\frac{1}{\lambda_2})} + \alpha$$

As seen in the plot below, this function is small when both values are close to zero, or when one value is significantly higher than the other. Plot is very similar to the original function.