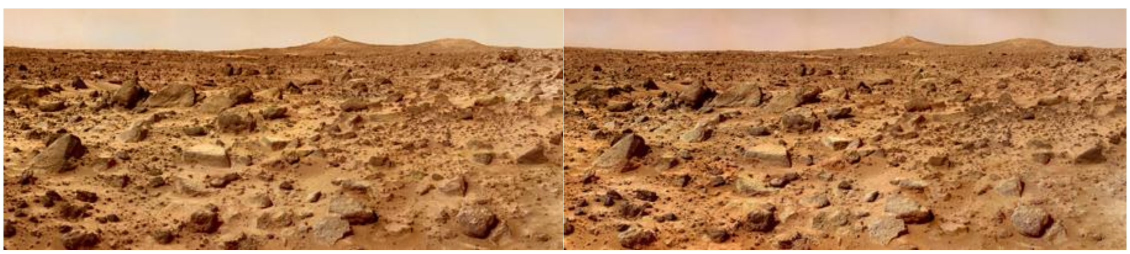
Zeta Surgical Programming Challenge

This programming challenge will test your ability to integrate various external libraries together into your own program to suit your needs. Please perform this in C++. You may use any IDE of your choice, as well as online resources as needed to supplement your knowledge. If any online resources are used, please provide citations.

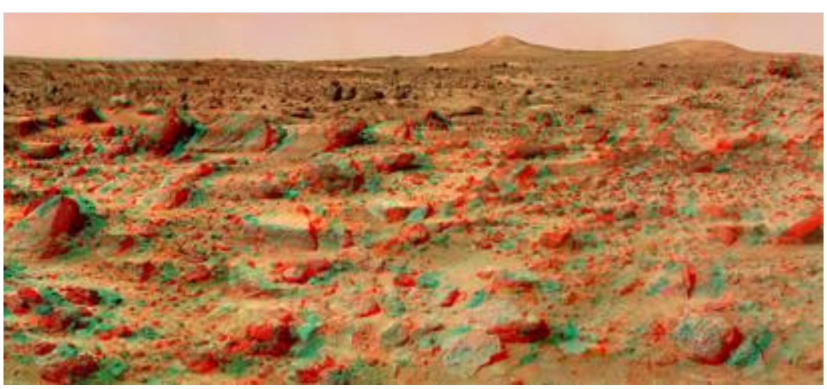
I understand that you are busy with other commitments and may not be able to complete the whole assignment. Please do your best and prepare a presentation that covers however much you were able to accomplish in the given timeframe. The purpose of this challenge is to analyze the way in which you approach, process, and solve problems, not necessarily to get the correct answer. I'm looking forward to seeing what you produce!

Background

A stereo pair is a set of images that contains two views of the same scene displayed side-by-side. Below is an example of a stereo pair taken from the [Mars Pathfinder mission](#).



When a viewer is fed one image into their left eye, and the other image into their right eye, they are able to get a perception of 3D depth of the scene. Many methods can be used to feed each eye a separate image, but one of the most common ones is the use of red/cyan 3D viewing glasses. Each separate image is tinted and then combined together into an image known as an analygraph, as in the image below.



Task

Your task is to create a program with multiple features that allow a user to manipulate, create, and deconstruct analygraphs. You may use the following C++ libraries:

- Eigen: matrix math
- OpenCV: image loading, management
- FreeImage: image loading
- OpenMP: multithreading

While these libraries are provided as suggestions, you do not necessarily need to use any or all of them in your program. You may also use C++ standard library code to perform any of the necessary components of the task.

Program features:

- Given an image stereo pair (either as one file, side-by-side combined image as seen here or as two separate image files), convert the input into a single-picture 3D analygraph. This feature should allow the user to select between various methods for coloring the final image. See the “Analygraph Generation Functions” section on the next page for some of these methods.
- Attempt to optimize the runtime and/or space complexity of your program. You may use multithreading in order to do so.

You do not need to do any image matching or warping. Assume that each pixel position is preserved in the left image, right image, and analygraph. Assume each image has the same resolution.

Please prepare the following for submission:

- Git repository with final code
- A 15 to 20 minute presentation covering:
    - Development environment setup, libraries used
    - Code overview, program structure
    - Optimization process and findings
    - Program demo
- The presentation format is at your discretion. What’s important is that you present your ideas clearly and effectively.

These requirements are meant to be open-ended, allowing you to design your program how you see best fit.

Analygraph Generation Functions

The below formula calculates the analygraph's RGB values (R_a, G_a, B_a) from the values of the original left and right images. (Source 1, Source 2, Source 3)

$$\begin{pmatrix} R_a \\ G_a \\ B_a \end{pmatrix} = M_l \begin{pmatrix} R_l \\ G_l \\ B_l \end{pmatrix} + M_r \begin{pmatrix} R_r \\ G_r \\ B_r \end{pmatrix}$$

Where M_l and M_r are matrices designed to optimize the image on various parameters. This computation is done for each pixel in the image. Below are some matrix pairs obtained from the above sources.

True analygraph: $M_l = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, M_r = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.299 & 0.587 & 0.114 \end{pmatrix}$

Gray analygraph: $M_l = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, M_r = \begin{pmatrix} 0 & 0 & 0 \\ 0.299 & 0.587 & 0.114 \\ 0.299 & 0.587 & 0.114 \end{pmatrix}$

Color analygraph: $M_l = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, M_r = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Half-Color analygraph: $M_l = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, M_r = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

3DTV-optimized analygraph: $M_l = \begin{pmatrix} 0 & 0.7 & 0.3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, M_r = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

DuBois analygraph: $M_l = \begin{pmatrix} 0.437 & 0.449 & 0.164 \\ -0.062 & -0.062 & -0.024 \\ -0.048 & -0.050 & -0.017 \end{pmatrix}, M_r = \begin{pmatrix} -0.011 & -0.032 & -0.007 \\ 0.377 & 0.761 & 0.009 \\ -0.026 & -0.093 & 1.234 \end{pmatrix}$

Roscolux analygraph: $M_l = \begin{pmatrix} 0.3185 & 0.0769 & 0.0109 \\ 0.1501 & 0.0767 & 0.0056 \\ 0.0007 & 0.0020 & 0.0156 \end{pmatrix}, M_r = \begin{pmatrix} 0.0174 & 0.0484 & 0.1402 \\ 0.0184 & 0.1807 & 0.0458 \\ 0.0286 & 0.0991 & 0.7662 \end{pmatrix}$