Synthetic Lighting for Photography

CS 452 - Computer Graphics
Assignment 1

When a scene is lit from various light sources, they can be added together to get the total illumination in the scene. This principle is called super-position and can be used to post-process photographs of scenes in which the contribution of each light source is known. In this assignment we will be using this principle to create interesting visual affects by changing the color of light sources in a scene and by using negative light sources.

A little background on images:

- Digital images are essentially large matrices that contain the color values of each pixel displayed on a screen.
- Grayscale images are 2D matrices.
- Color images have 3 layers (R, G, B).
- Each cell of the matrix represents a pixel.
- Each color in a pixel is quantized to a set of values, e.g., from 0-255.

In this program you will learn

- Concept of R,G,B colors
- Image manipulation in code

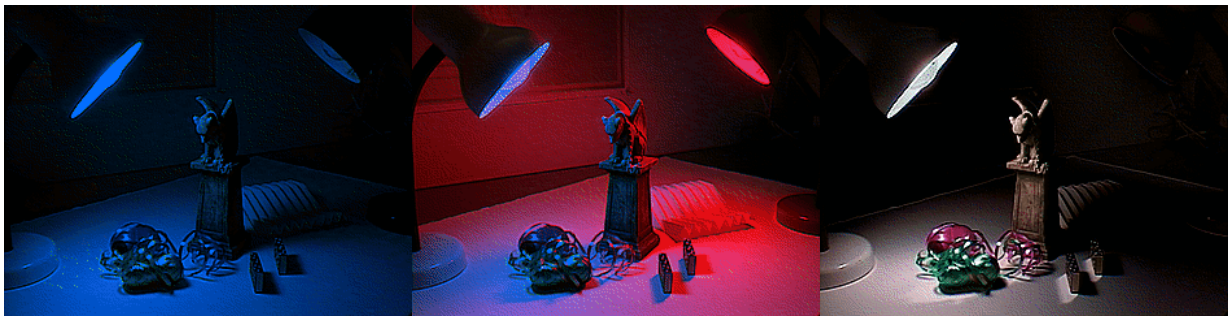
Multiple images are taken with different lights, without moving the camera

See: Graphica Obscura – http://www.graficaobscura.com/synth/index.html
Also see Magnetic - http://magnetiq.com/pages/multiple-synthetic-lights/

As an input data you can either use the images uploaded on LMS or you can use the images available on the above websites. You can also capture you own images.

Interesting manipulations can now be done…

Subtract ambient to find contribution of each light source
Change color of lights by modifying the contribution of red, green and blue components
Generate negative lights!



Another interesting thing you can do is morphing. Assume your input is two images (lets say the right lamp contribution and left lamp contribution). Now using the equation of weighted average, generate a sequence of images that have the intermediate lighting:

$$I_t = wI_a + (1\text{-}w)I_b$$

Where I_t is one image in a sequence, I_a and I_b are the input images and w is the weight that is varied between 0 and 1 at equal intervals. For example, if w is varied at intervals of 0.1, then the sequence of 10 images is:

$I_{t01} = 0.0 * I_a + 1.0*I_b$
$I_{t02} = 0.1 * I_a + 0.9*I_b$
…
$I_{t10} = 1.0 * I_a + 0.0*I_b$

Watch the video CgAssign1.wmv from LMS to see the kind of visualization you can create.

You will have to…

You have been provided with a skeleton code in which you are required to implement the Initialize, Render and Destroy functions in the morphing.cpp file in the VC++ 2010 project uploaded on LMS. Your output could be an interactive animation in which the image morphs as the mouse moves in the window or it could be a video/sequence of images generated. Mouse coordinates are available to you as parameters to the Render function. You can implement the project using other tools such as MSYS/MinGW but you may not get support from the TAs and instructors.

Get creative with your morphing! You can choose to implement different types of light superimpositions, and even apply customized color filters. You must implement at least two different types of morphings. You may implement a third for bonus points.

For this assignment, you will be using the FreeImage library to read and manipulate images. FreeImage is a small, convenient library for image I/O and manipulations. A pdf file containing documentation for FreeImage is uploaded on LMS.

Submission Details

- DUE: Monday Sep 08th, 11:55 p.m.
- Submission will be done on LMS.
- You must submit only your morphing.cpp file from your VC++ 2010 project. DO NOT SUBMIT THE ENTIRE PROJECT!. In case you have developed your code using MSYS/MINGW, submit all the files you have modified along with instruction on how to compile the code
- Submit at least 5 snapshots of various images that you have obtained