**CSC 545/645 Computer Speech, Music and Images**

**Exercise No. 10b, Week 13, Due April 18, 2021**

**Chromakey**

**Goals**

Fly a helicopter on the moon

**Background**

You may have heard that it’s impossible to fly a helicopter on the moon because of the lack of atmosphere. You are going to prove your physics teacher wrong, using chroma key (or greenscreen) technology.

**Procedure**

Write a Processing program to insert a background in a green screen video. Start with Ex10b\_VideoChromakey, downloaded from Blackboard. The data folder holds an image of the moon’s surface and a video of a helicopter flying against a green background. Replace the green background in the video with the image of the moon’s surface. The moon’s image has to be resized to the same size as the movie frames (already done); the alternative is to resize each frame of the movie (you could try that). Write the code for addImage() to insert the background over the green screen.

Also, add some hotkey controls: ‘g’ toggles whether to replace the green or not; ‘p’ toggles pause; ‘r’ reverses the direction of play (use the global variable *playSpeed)*. Note that, after making these changes, you have a rudimentary video player—when *replaceGreen* is *false,* you can play any video, pause it, and reverse the play. One thing I have noticed is that when the video reaches the end—whether it’s playing forward or backward, I have to pause it before reversing in order to get the video to display frames again.

The helicopter video was downloaded from YouTube. You may want to find other green screen videos and replace their green background with other background images.

**Chromakey algorithm**

The chromakey algorithm is simple: scan each pixel of the foreground image. If the pixel is the background color (usually green or blue), replace the foreground pixel with a pixel from the background image. The only real problem is that it can be tricky to determine if the pixel is the background color—shadows and shades of illumination can make the color vary. One possibility, when using a blue screen, is to compare the blue value against the sum of the red and green values; if the blue value is greater, then it’s a background pixel. That may not work for all images.

**Deliverables**

Submit your pde file on Blackboard by the due date.