Assignment 1 – Raster images and basic processing

UPR Mayagüez - COMP 4046 - Computer Graphics

Abstract: Implement image processing and visualization using the 2D HTML5 canvas.

Guidelines for assignment hand in:
- Program should be using the 2D HTML5 canvas with Javascript, unless explicitly.
- One ZIP file contains all code and data necessary to run your assignment. It is organized as follows
- A main index.html HTML file is present at top level. It provides links to all results with a short explanation.
- If several results are asked in an exercise, provide a way to run each of them individually (using a button callback or independent HTML pages). File organization (filenames, flat or hierarchical subfolders…) is up to you, but should be clear and meaningful.
- The example should run automatically by simply following the links, when the code is run in the Firefox browser through a webserver as seen in class. If any additional action is required to run the results, they must be listed in the main page.
- Provide a screenshot or PDF of each requested result with meaningful filenames.
- Hand in by email to remi.megret@upr.edu using the following object:
[COMP4046 Assg1] First_name FAMILY_NAME

1. Image processing and GUI [40]

Objective : Process raster images using simple pixelwise operations and add interactivity.

Setup a page with at least two canvases. Draw the original input image in the first one, process its data using pixelwise operations, and put the content of the output images in the other canvases.

Question 1: Decomposition into individual components:
Keep only the R component, setting the G and B components to zero.

Question 2: Thresholding of grey-scale image
Convert the image to grey-level by averaging the R, G, B components.
Threshold the grey-level image at grey level threshold=120.

Question 3: Linear correction
Apply a linear correction on the pixel values R, G, B:
- Increase the contrast of the image by a scale=1.2, leaving intensity value center=128 unchanged
- then lighten the image by offset=+30 intensity levels.

Question 4: Interactivity - Parameters
Add controls to modify interactively offset, scale, center and threshold.

Question 5: Interactivity - Mouse
Add mouse handling that enables the user to select a rectangular zone (x,y,w,h). The output image should be a copy of the corresponding region from the input image. Make sure the output canvas is resized according to this content.

Bonus*: Display the selected rectangle interactively in the canvas without introducing artifacts,
Bonus**: Compute and display the image histogram.

2. Colored disks [30]

Objective : Create graphics by direct pixel manipulation and using 2d canvas operations.

Question 1: Colored disk

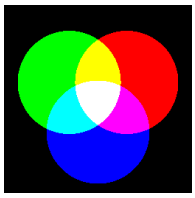
a) Draw a colored disk in the canvas *by modifying each pixel one by one in the RGBA array* obtained by getImageData() (rasterization of the disk)

- Define the coordinates x0, y0 of the disk center and a RGB color.
- For each pixel (x,y), compute the distance to (x0,y0) and modify the pixel R, G, B values if this distance is less than the radius.
- Draw the image using putImageData()

b) Compare your result with the result obtained by using the higher level 2d canvas function arc().

Question 2: Additive synthesis

Extend the code of Q1a so that now 3 disks are shown (see example image), and the image is generated using additive synthesis of color pixel by pixel.



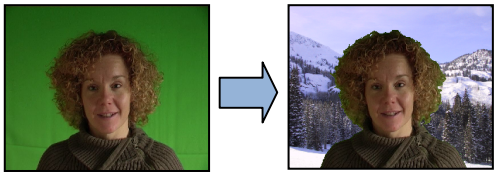
Question 3: Animation

- Animate the display of Q2a to show the 3 disks rotating around the image center by using requestAnimFrame()

Bonus*: Tune the rotation speed to be as close as possible to one full rotation / second.

3. Chroma-key [30]

Objective : Implement the chroma-key technique to replace the green background by another image.



Question 1: RGB segmentation

a) Load image « shelley4.jpg »[1].
Display separately the 3 channels R/G/B.

b) Find intervals of the type $[R_{min},R_{max}]$ for each channel that contains all pixels from the background. Explain how you found these values.

c) Use previous intervals to create an alpha mask where value=0 for the background and value=255 for the foreground. Display this mask as a black/white image.

Question 2: Alpha-compositing

Replace the background pixels by the content of another image (« montagne.png »[2]) using the alpha composition method and the mask computed at Q1. (see example image)

Bonus***: Apply the approach to a video with green background using the HTML5 <video> tag.

[1] Source : http://www.bfvs.fsnet.co.uk/chromakey/chromakey.htm
[2] Source : http://commons.wikimedia.org/wiki/Image:Usa_Utah_Brighton_Peaks.jpg