EGR 115 Final Project Proposal – SVG Inverter

Daniel E. Janusch

November 12, 2024

1 Input/Output Examples

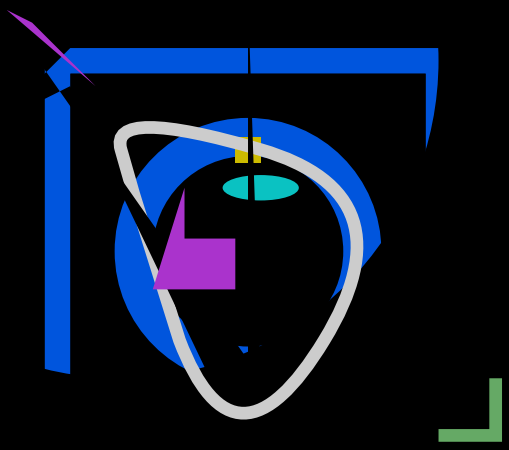

Figure 1: Ex 1. Original SVG

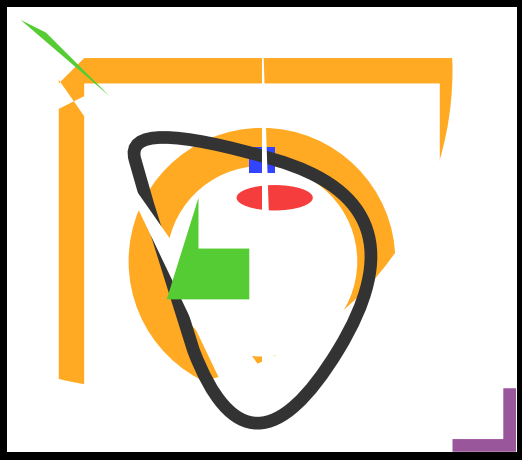

Figure 2: Ex 1. Inverted SVG

Figure 3: Ex 2. Original SVG

Figure 4: Ex 2. Inverted SVG

2 Program Overview

2.1 Program Psueocode

```
print the help text if requested

if no file was given, throw an error

read in the file content to a variable

test if each <rect> has the same dimensions as the SVG
add a background <rect> to the SVG if there wasnt one already

invert all the stroke colors
invert all the fill colors

find all the IDs of images referenced in <use> tags within the <mask> tag
find all the inline images within <mask> tags

for each <image> in the SVG
    if the image is defined or referenced in a <mask>
        continue

    invert the image using ImageMagick and update the SVG

trim and output the content, either to a file or to stdout.
```

2.2 MATLAB Capabilities Used

- ☑ conditionals (e.g. `if`, `switch`)
- ☑ loops (e.g. `for`, `while`)
- ☑ matrix and array operations* (e.g. `.*`, `.^`, `\`)
- ☑ string and character arrays and operations
- ☑ data I/O (e.g. file or stdin/stdout)
- ☐ visualization (plots)
- ☑ boolean operations (e.g. `||`, `&&`)
- ☑ built-in functions (e.g. `struct`, `extractBetween`, `regexpi`)

*only if the additions are made, discussed on the next page

3 Possible Project Additions

I could use some kind of continuous map like a vector function or matrix transformation instead of just inverting the colors. I could also rasterize the inverted image and analyze and plot the color distribution, which would mark off the visualization aspect of MATLAB. Using a color transformation matrix wouldn't allow for all linear transformations, so this will be used instead:

$$C_2 = \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix} + \begin{bmatrix} a \\ b \\ c \end{bmatrix} \odot \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} r_0 + aR \\ g_0 + bG \\ b_0 + cB \end{bmatrix} \text{ given } \begin{bmatrix} R \\ G \\ B \end{bmatrix} \text{ as the input.}$$

and either $C_{\text{final}} = \text{clamp}(C_2, [0 \;\; 255])$ or $C_{\text{final}} = C_2 \bmod 256$

$\vec{u} \odot \vec{v}$ is the Hadamard Product of $\vec{u}$ and $\vec{v}$ (`u .* v`).

4 Appendix

Example 1 original:

```svg
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" width="800" height="700" viewBox="0 0 40 35">
    <rect width="40" height="35" fill="#000"/>
    <rect width="30" height="30" x="5" y="5" stroke-width="2" fill-opacity="0" stroke="#05d"/>
    <circle cx="20" cy="20" r="9" stroke-width="3" fill-opacity="0" stroke="#05d"/>
    <circle cx="10" cy="5" r="34" stroke-width="18" fill-opacity="0" stroke="#000"/>
    <rect x="19" y="11" width="2" height="2" fill="rgb(204, 187, 0)"/>
    <ellipse cx="21" cy="15" rx="3" ry="1" fill="hsl(180, 90%, 40%)"/>
    <path d="M 11 15.5 T 10 12, 21 12, 26 27, 14.6 27 Z" stroke="#ccc" fill="none"/>
    <polygon points="0 0 10 0 0 10 6 7 17 30 20 30 20 0 21 30" fill="#000"/>
    <path d="M 1 1 l 2 1 5 5 z M 15 15 v 4 h 4 v 4 h -6.5" fill="#a3c"/>
    <polygon points="40,35  40,30  39,30  39,34  35,34  35,35" fill="#65a965"/>
</svg>
```

Example 1 final:

```svg
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" width="800" height="700" viewBox="0 0 40 35">
    <rect width="40" height="35" fill="#fff"/>
    <rect width="30" height="30" x="5" y="5" stroke-width="2" fill-opacity="0" stroke="#fa2"/>
    <circle cx="20" cy="20" r="9" stroke-width="3" fill-opacity="0" stroke="#fa2"/>
    <circle cx="10" cy="5" r="34" stroke-width="18" fill-opacity="0" stroke="#fff"/>
    <rect x="19" y="11" width="2" height="2" fill="rgb(51, 68, 255)"/>
    <ellipse cx="21" cy="15" rx="3" ry="1" fill="hsl(0, 90%, 60%)"/>
    <path d="M 11 15.5 T 10 12, 21 12, 26 27, 14.6 27 Z" stroke="#333" fill="none"/>
    <polygon points="0 0 10 0 0 10 6 7 17 30 20 30 20 0 21 30" fill="#fff"/>
    <path d="M 1 1 l 2 1 5 5 z M 15 15 v 4 h 4 v 4 h -6.5" fill="#5c3"/>
    <polygon points="40,35  40,30  39,30  39,34  35,34  35,35" fill="#9a569a"/>
</svg>
```

Example 2 is derived from `https://upload.wikimedia.org/wikipedia/commons/7/7b/Adobe_Systems_logo_and_wordmark.svg`