Brief Notes on the Cool Image Effects

In this note, we provide an 'under–the–hood' view of the cool effects shown in the application alongside. We give the mathematical formulae, and brief explanations as necessary.

There are two types of algorithms yielding these cool effects:

1. **Scalable**: Algorithms giving the same result irrespective of image size (different sizes of the same image). Applying these algorithms on an image of size 600×450 yield the same results as on the corresponding image of size 1800×1350, within limits of numerical accuracy – in other words, one appears as a scaled version of the other. This is analogous to WYSIWYG. These are based on point operations on the pixels.
2. **Non-scalable**: Algorithms giving different results on different sizes of the same image. These are not WYSIWYG algorithms. Stated otherwise, one does *not* appear as a scaled version of the other. These are based on area/neighbourhood operations on the pixels.

1 Non–Scalable Effects

As mentioned earlier, these effects apply differently on images of different sizes.

1.1 Edges and Outline Effects

1. Convert the image to grayscale.
2. Apply the filter $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ to get the corresponding gaussian image. Don't worry about normalizing the image now; we'll do it later.
3. Compute the gradient image $\nabla I = \begin{bmatrix} I_x & I_y \end{bmatrix}'$ of this gaussian. This gradient image has two components, and these are $I_x = \dfrac{\partial I}{\partial x}$ and $I_y = \dfrac{\partial I}{\partial y}$; one way of computing these is via Sobel filters $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ respectively.
4. Compute the θ and $|\nabla I| = \text{magnitude}$ images. The θ image is computed as $\theta = \arctan\left(\dfrac{I_x}{I_y}\right)$ (can also use the reciprocal of our $\arctan$ argument), and $|\nabla I| = \sqrt{I_x^2 + I_y^2}$.
5. The last step is to determine the colour of each pixel, depending upon the angle θ and the intensity (0 – 255) of that pixel depending upon its $|\nabla I|$ value. For this, we need to compute the maximum and minimum magnitude values, and scale that to the range $0 - 255$. Further, colours needs to be assigned such that a vertical edge is yellow; horizontal is blue; and the other two inclined edges are red and green in colour.