W 4115
Programming Language and Translators
Proposal – Revised
Jahyun Kim (jk3111)

Pixelish (Pixel + -ish)

Pixelish, Pixel + -ish from English, is a simplified image processing language which specifically deals with pixels of image. Pixelish provides a built-in image data type that can be used to easily manipulate the pixels of an image to realize various effects such as turning into grayscale, creating a filter effect similar to that of Instagram, vignetting, and warping of shapes.

Goal

The primary goal of Pixelish is to deliver a simplified, user-friendly image processing experience to users. Many languages support image processing, but there is a need for special libraries and extensive coding. Instead, Pixelish reduces the complexity and provides basic, yet enough features to specifically support the image transformation through the pixel manipulation. There are two main features supported by this language.

First, with the help of Pixelish, users can change brightness, contrast, rgb level, and hs(v) level of images without any special programs such as Photoshop. This is done by the built-in fields of the image data type such as red, blue, green. Users can easily play around with these fields to come up with customized color adjustment that can be done on the entire image.

Another feature is the pixel manipulation. Users can access each pixel of the image to alter the color intensity, rendering, and position of the pixel. With this feature, users can generate vignetting effect, or partially blurred image, or distorted shapes. The change in position of a pixel means the swapping of the pixels from different locations. If this relation is not specified, then the original position of the pixel will be filled with white by default, and the selected pixel will replace the pixel in the desired location.

Syntax

There must be a single main function where the program happens. A program is either just the main function where everything is written, or is a series of function definitions followed by the main function to execute them.

Data Types

image: 2-D image, similar to Mat in OpenCV (2-D array)
int, float, string, array

Comments

/* */: Only a multi-line comment is supported.

Block

{ }

function: for functions such as fun in OCaml

semi-colon (;): for ending lines
imread(image_file_name): for reading the image file
imout(image, image_file_name): for printing out the image under the given name

Example

Example 1: Color adjustment on the entire image.

```
function main () {

   image example;

    /* Read the image file */
   example = imread(“./image1.jpg”);

   /* Increase the Red level by 50% */
   example.red = example.red*1.5;

   /*
     * Increase the Blue level by 10.
     * If the resulting value is greater than 256, the error message will appear.
     */
   example.blue = example.blue + 10;

   /* Print the resulting image. */
  imout(example, “./image1_copy.jpg”);
}
```

Example 2: Pixel manipulation.

```
function yoda (image in) {
   /* Changes the even position pixels (both height and width) to green
     * and swaps the pixels in positions ending with 3 with the ones in positions
     * ending with 9 (both height and width).
     */

   int h = in.height;
   int w = in.width;

   /* Change the even position pixels to green */
   /* yoda green: 009900 (hexadecimal) */
   for (int i = 0; i < w; i =  i + 2) {
      for (int j = 0; j < h; j = j + 2) {
         in.(i, j) = “009900”;
      }
   }
```

```
    /* Position Swapping */
    for (int i = 3; i  < w; i = i + 10) {
        for (int j = 3; j < h; j = j + 10) {
            pixel temp = in.(i + 6, j + 6);
            in.(i + 6, j + 6) = in.(i, j);
            in.(i, j) = temp;
        }
    }
}

function main () {
    image example;

    /* Read the image file */
    example = imread(“./image2.jpg”);

    /* Apply “yoda” function */
    yoda(example);

    /* Print the resulting image */
    imout(example, “./image2_copy.jpg”);
}
```

References

1. http://mashable.com/2013/10/20/photoshop-instagram-filters/
2. docs.opencv.org
3. http://www.processing.org/tutorials/pixels/