

Procesamiento de Imágenes, Características del color.

Camilo Andrés Nemocón Farfán

Resumen— Este informe da a conocer el funcionamiento del software de procesamiento de imágenes utilizado en el proyecto de investigación “Diseño de nuevas propuestas cromáticas y patrones sobre prendas de uso militar”, cuyas directoras del proyecto son Amparo Quijano, Zamira Kadamani y Jorge Gutiérrez.

Palabras Claves— Procesamiento de imágenes, Processing, Patrones, Paleta cromática, Camuflaje.

El procesamiento de imágenes tiene como función principal, analizar las imágenes de forma digital para así adquirir las características de los diferentes elementos que la componen como son el tono, la iluminación, el tinte, la claridad, el contraste, la exposición, la saturación y demás detalles que componen la imagen.

El objetivo de este proceso es poder adquirir propiamente las características de tono, saturación y brillo, y los componentes de color rojo (R), verde (G) y azul (B), de las imágenes procesadas para así poder generar una escala estadística de los valores resultantes y generar una paleta cromática con pantones, basados en la imagen fuente y así mismo producir un camuflado para esa imagen.

El primer paso para la metodología de procesamiento es tomar fotos a una resolución de 3872 X 2592 pixeles, estas fotos tienen como requerimiento, estar enfocadas hacia un ambiente visual homogéneo pero con diversidad vegetal para así obtener características cromáticas de una fuente natural y generar un camuflado coherente a la imagen base.

A partir de esta imagen se utiliza un proceso digital implementado en Processing, el cual es un entorno de desarrollo de programación usado como herramienta para la producción de aplicaciones multimedia e interactivas; en este caso utilizamos este entorno de desarrollo para realizar un procesamiento de los pixeles de la imagen, por lo cual el programa ajusta la resolución de la imagen a 320 X 240 pixeles y procede a hacer una segmentación de la imagen unificando los colores de los pixeles con aspectos cromáticos similares y produciendo una cuadrícula de 40X30, la cual relaciona los colores de alrededor en un solo cuadrado dándome un total de 1.200 cuadrados con todas las características de la imagen.

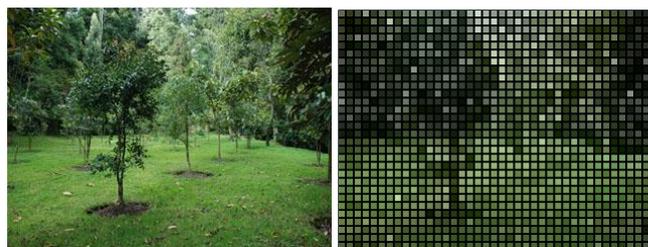


Fig 1. Imagen fuente e imagen segmentada en cuadrículas.

En este punto del procesamiento de la imagen, el programa recorre cada uno de los cuadrados generados y recolecta en una base de datos los valores de la imagen y a partir del algoritmo basado en la ecuación matemática gradiente se determina los colores y bordes de cada una de las formas que componen la imagen.

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Ecuación 1. Gradiente de una imagen $f(x,y)$ en (x,y) .

Por otro lado se genera la implementación de las funciones propias de las librerías de Processing, como es `hue()`, `saturation()` y `brightness()` para obtener los valores de cada uno de los puntos de la cuadrícula y que estos sean guardados en un arreglo de datos generando un reporte posterior con las características de la imagen.

```
//tono
numTono[numPixel] = hue(col);
enteroTono[numPixel] = int(numTono[numPixel] * 10);
numeroTono[numPixel] = int(numTono[numPixel] * 1);
ultCifraTono[numPixel] = enteroTono[numPixel] % 10;
if(ultCifraTono[numPixel] >= 5)
{
  numeroTono[numPixel] = numeroTono[numPixel] + 1;
}

//saturacion
numSat[numPixel] = saturation(col);
enteroSat[numPixel] = int(numSat[numPixel] * 10);
numeroSat[numPixel] = int(numSat[numPixel] * 1);
ultCifraSat[numPixel] = enteroSat[numPixel] % 10;
if(ultCifraSat[numPixel] >= 5)
{
  numeroSat[numPixel] = numeroSat[numPixel] + 1;
}

//brillo
numBrillo[numPixel] = brightness(col);
enteroBrillo[numPixel] = int(numBrillo[numPixel] * 10);
numeroBrillo[numPixel] = int(numBrillo[numPixel] * 1);
ultCifraBrillo[numPixel] = enteroBrillo[numPixel] % 10;
if(ultCifraBrillo[numPixel] >= 5)
{
  numeroBrillo[numPixel] = numeroBrillo[numPixel] + 1;
}
```

Algoritmo 1. Implementaciones de las funciones de saturación, brillo y tono sobre la imagen.

Mediante la estructura implementada para el análisis de la cuadrícula, se pasa a la investigación cromática de los componentes del color, definidos a nivel de rojo (R), verde (G) y azul (B), donde a partir de la teoría de color, estos componentes tienen un valor que corresponde al rango de 0 a 255 y la composición entre estos me determinan un color que también puede ser representado bajo una identificación única expresada en términos de pantones.

Se escala el programa implementando la adición de los componentes de color a cada cuadro de la imagen segmentada, para así determinar y guardar los valores de rojo, verde y azul, y adicionalmente traduciendo estos valores en términos de pantones.

```
for(int i=0 ;i <im.width; i=i+1)
{
    for(int j=0 ;j <im.height; j=j+1)
    {
        numPixel = (j*im.width)+i;
        col=im.pixels[numPixel];
        fill (red(col),green(col),blue(col),alpha(col));
        rect(i*8,j*8,6,6);
        rojo[numPixel]=red(col);
        verde[numPixel]=green(col);
        azul[numPixel]=blue(col);
        panton[numPixel]=hex(col, 6);
    }
}
```

Algoritmo 2. Implementaciones de las funciones R,G,B,Hex sobre la imagen.

Con el procesamiento completo de la imagen se obtiene un reporte con 8 valores por cada segmento analizado de la cuadrícula en el siguiente orden, Número de pixel, valor de rojo, valor de verde, valor de azul, saturación, tono, brillo, y su identificación única en términos de pantone. Estos datos son analizados y evaluados estadísticamente, a partir de los valores promedios y máximos que posee la imagen para así determinar cuál es la unificación del color que me genera un camuflado de acuerdo a un resultado estadístico basado en la descomposición del color a nivel algorítmico y matemático.

NUM PIXEL	R	G	B	tono	saturacion	brillo
1	97.0	106.0	77.0	79	27	42
2	51.0	62.0	32.0	82	48	24
3	70.0	81.0	51.0	82	37	32
4	142.0	157.0	126.0	89	20	62
5	49.0	65.0	36.0	93	45	25
6	99.0	120.0	89.0	101	26	47
7	71.0	94.0	65.0	108	31	37
8	50.0	77.0	46.0	112	40	30
9	103.0	129.0	100.0	114	22	51
10	124.0	153.0	123.0	118	20	60
11	110.0	139.0	109.0	118	22	55
12	101.0	130.0	100.0	118	23	51
13	81.0	110.0	82.0	122	26	43
14	51.0	80.0	52.0	122	36	31
15	124.0	150.0	123.0	118	18	59

Fig 2. Reporte generado mostrando los valores de 15 segmentos de la imagen procesada.

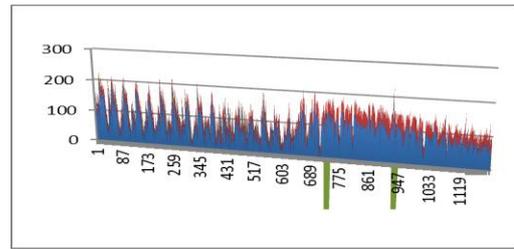


Fig 3. Gráfica de promedios y máximos rgb Vrs. Cuadrados Segmentados.

Finalmente el análisis de promedios y máximos determinan 12 valores, los cuales son 3 valores promedios a nivel de RGB (rojo, verde, azul), otros 3 valores promedio a nivel HSB (tono, saturación, brillo). Por otro lado se establecen 3 valores máximos a nivel de RGB (rojo, verde, azul) y los últimos 3 valores promedio a nivel HSB (tono, saturación, brillo), para así tener finalmente la composición de 4 colores formando una paleta tetra cromática, generada a partir del procesamiento de una imagen fuente, con lo cual, estos colores resultantes son pantones que se camuflaran sobre esta imagen.

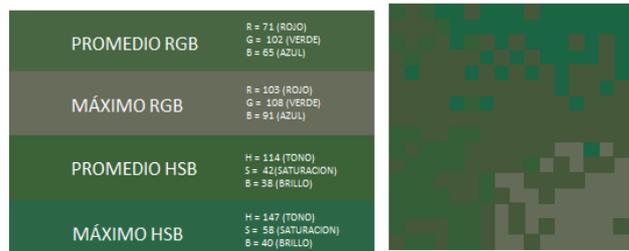


Fig 4. Paleta Tetra Cromática.

BIBLIOGRAFÍA

- [1] Visualizing Data. Ben Fry. Published 2007, O’Reilly.
- [2] Programming Interactivity. Joshua Noble. Published 2009, O’Reilly.
- [3] Interaction design beyond human-computer interaction. Jhon Wiley. Published 2007, Sharp Rogers.

REFERENCIAS

Ecuacion1.
<http://gavab.escet.urjc.es/wiki/download/vc/Tema4/temaSegmentacion.pdf>