Congr. Int. en Ing. Electrónica. Mem. ELECTRO, Vol. 43 pp. 11 - 14, Oct 2021, Chihuahua, Chih. México
http:// electro.itchihuahua.edu.mx/revista
ISSN 1405-2172

CODE STORAGE

Villanueva Vallejo Raúl, Herrera Guerrero Ana, Reyes Ibarra Luis Alberto, Romero Alvarez Nelson Paul
Universidad Politécnica de Durango
Ingeniería en Redes y Telecomunicaciones
Carr. Durango-México Km 9.5 S/N, Loc. Dolores Hidalgo, C.P. 34300, Durango, Dgo.
618 4564260
raul.villanueva@unipolidgo.edu.mx, annaherrera280@gmail.com, luis.reyes@unipolidgo.edu.mx,
nelson.romero@unipolidgo.edu.mx

RESUMEN.

Se presenta un módulo de información basado en 3 tonalidades de colores que permiten que el módulo sea capaz de almacenar gran cantidad de información en comparación con los módulos actuales. La implementación del módulo se realiza mediante un sistema de cómputo el cual está diseñado en Matlab®, el sistema trabaja a partir de una interfaz amigable, además de un algoritmo diseñado para codificar los datos almacenados. La lectura del módulo funciona gracias a la identificación de colores y numero de figuras con base en los cuales se crea un archivo .txt en donde es almacenada la información.

Palabras Clave: Almacenamiento de información, algoritmo, sistema de cómputo.

ABSTRACT.

An information module based on 3 shades of colors that allow the module to be able to store large amounts of information to compare the current modules is presented. The module implementation is performed by a computer system which is designed in MatLab®, the system works from a friendly interface and an algorithm designed to encode the stored data. Reading module works by identifying colors and number of figures on the basis of which a .txt file where the information is stored is created.

Keywords: Information storage, algorithm, computer system.

1. INTRODUCCIÓN

A medida que se va desarrollando los avances en la tecnología, todo aquello que rodea a la sociedad de igual manera va evolucionando, poco a poco las cosas más comunes van entrando en un entorno tecnológico y más práctico para la vida del ser humano. En el entorno es muy importante resaltar las necesidades de la humanidad en la subsistencia diaria, en este caso se toman en cuenta acciones tan sencillas como lo es acceder a información o realizar cierta acción mediante la captura de datos en un esquema y al transformarlos proporcionan una salida. La tecnología no detiene su evolución, por lo tanto, surge la necesidad de ir creando a la par nuevos sistemas que permitan cubrir las necesidades que van surgiendo conforme al paso del tiempo. Años atrás, era imposible imaginar un disco duro que fuera capaz de almacenar 1 TB de información, y esto fue posible gracias a la necesidad que surgió de crear dispositivos con más capacidad de almacenamiento, de igual manera dentro de los próximos años, los códigos QR o los códigos de barras no proporcionarán la memoria de almacenamiento necesaria para la época.

2. DESARROLLO

2.1. Descripción del sistema

En la actualidad existen módulos de información cuya función consiste en almacenar información, los más comunes en la actualidad son los códigos QR, ya que a diferencia de los códigos de barras pueden almacenar hasta 4,296 caracteres alfanuméricos utilizando una gran cantidad de pixeles, contando además con un sistema de corrección de errores [1]. Los códigos QR corresponden a códigos bidimensionales, fácilmente identificables por tres cuadros ubicados en las esquinas superiores e inferior izquierda [2]; los códigos de barras representan la clave de un registro de una base de datos donde realmente la información es almacenada. La cantidad que almacenan ambos es poca en relación con la cantidad de pixeles que involucran, por lo cual se crea un nuevo módulo de información que pueda almacenar mayor cantidad de información con menor cantidad de pixeles.

Con base en lo anterior, el departamento de Ingeniería en Redes y Telecomunicaciones de la Universidad Politécnica de Durango propone el desarrollo de un código de almacenamiento por colores RGB, que sea capaz de almacenar hasta el doble de información que los códigos actuales, así como el desarrollo del algoritmo que, a través de un sistema de cómputo, sea capaz de interpretar el código.

2.2 Desarrollo del sistema de descifrado del código

Para la captura de fotos se necesita una cámara de resolución media-alta para captar los colores detalladamente y de esta manera el sistema no tenga problemas para identificar los colores. Un ejemplo de ello se muestra en la figura 1.

Figura 1. Muestra del sistema

Como se apreció en la figura citada anteriormente, el sistema se conforma de diferentes secciones: visión de la cámara, configuraciones y, por último, sección de visualización del resultado.
Los botones que conforman la sección de configuración realizan las siguientes acciones:

- Combobox de resolución, el cual permite elegir la resolución de la cámara.
- Combobox de formato, permite seleccionar el formato en el que se tomará la fotografía; dicho formato será establecido en RGB.
- Botón de video apagado, permite activar o desactivar la cámara en tiempo real.
- Botón de tomar fotografía, permite capturar la imagen a través de la cámara.

2.3 Desarrollo del algoritmo.

El sistema desarrollado en MatLab® se basa principalmente en la identificación de colores y en el número total de figuras, esto, a partir del siguiente algoritmo:

```
fot=videoinput('winvideo',2,res);
    set(fot,'ReturnedColorSpace',efec);
    foto=getsnapshot(fot);
    imshow(foto);
    imagen = foto;
    imagenGris = rgb2gray(imagen);
    imR = double(imagen(:,:,1));
    imG = double(imagen(:,:,2));
    imB = double(imagen(:,:,3));
    imagenV = (imG-imR-imB);
    imagenR = (imR-imG-imB);
    imagenA = (imB-imR-imG);
    imagen_binariaV = imagenV > 50;
    imagen_binariaR = imagenR > 50;
    imagen_binariaA = imagenA > 50;
    imagen_binariaV_filtrada = medfilt2(imagen_binariaV);
    imagen_binariaR_filtrada = medfilt2(imagen_binariaR);
    imagen_binariaA_filtrada = medfilt2(imagen_binariaA);
    mascaraV = imagen_binariaV_filtrada;
    mascaraR = imagen_binariaR_filtrada;
    mascaraA = imagen_binariaA_filtrada;
    imshow(imagen)
```

La detección de bordes es la forma más utilizada en segmentación para encontrar discontinuidades en valores de intensidad, esto, en teoría, permite delimitar su tamaño y su región [3].
"En una imagen, un borde es una curva que sigue un camino de cambio rápido en la intensidad de la imagen. La detección de bordes se utiliza para identificar los bordes de una imagen" [4].

La segmentación fue utilizada una vez que fueron filtrados los colores con el fin de garantizarle al usuario que el código fue analizado correctamente; lo anterior se realiza a partir del código:

```
[l Ne]=bwlabel(mascaraV);
    propiedV=regionprops(l,'BoundingBox');
    hold on
    [l Ne]=bwlabel(mascaraR);
    propiedR=regionprops(l,'BoundingBox');
    hold on
    [l Ne]=bwlabel(mascaraA);
    propiedA=regionprops(l,'BoundingBox');
    hold on
    R = 0;
    V = 0;
    A = 0;
```

2.4 Diseño de la matriz

Para el diseño del código se optó por utilizar colores en la escala RGB y representarlos mediante cuadrados (Figura 2). Esto debido a que el sistema cuenta el número de objetos encontrados; la matriz no necesariamente debe de ser de un tamaño especifico.
"Cada color en el modelo RGB tiene como componentes primarias al rojo, verde y al azul, y se forma por la mezcla aditiva de éstos" [5].

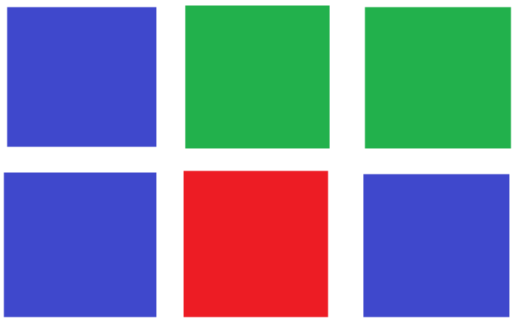

Figura 2. Ejemplo matriz de colores RGB

2.5 Diseño del algoritmo de almacenamiento

El almacenamiento del código se efectúa a través de:
for v=1:size(propiedV,1)

```
rectangle('Position',propiedV(v).BoundingBox,'EdgeColor','g','
LineWidth',2)
        V = V + 1;
    end
    hold off
    for r=1:size(propiedR,1)
rectangle('Position',propiedR(r).BoundingBox,'EdgeColor','g','L
ineWidth',2)
        R = R + 1;
    end
    hold off
    for a=1:size(propiedA,1)
rectangle('Position',propiedA(a).BoundingBox,'EdgeColor','g','
LineWidth',2)
        A = A + 1;
    end
    hold off
    T = A+V+R;
    fprintf('hay %d Objetos rojos\n',R)
    fprintf('hay %d Objetos verdes\n',V)
    fprintf('hay %d Objetos azules\n',A)
    fprintf('Total de objetos: %d \n',T)
    direc = 'infos/';
    direc = strcat(direc, num2str(R))
    direc = strcat(direc, num2str(V))
    direc = strcat(direc, num2str(A))
    direc = strcat(direc, num2str(T))
    direc = strcat(direc,'.txt')
    datos=importdata(direc);
    A = A * 1000;
    V = V * 100;
    R = R * 10;
    T = T;
    num = A+V+R+T;
    x = datos{1,1}
    set(handles.text5,'String',x)
```

El código mostrado permite contar el número de objetos encontrados y el número de objetos encontrados de cada color, por lo que, con base en estos datos, se genera un archivo de extensión .txt que contiene la información para su posterior relación con un archivo existente; posteriormente, el sistema muestra el contenido del archivo en pantalla como se muestra en la figura 3.

Figura 3. Identificación del código de almacenamiento

En la figura 3 se puede observar una matriz de 3x3 que contiene los siguientes datos:
Número de objetos azules encontrados: 3
Número de objetos rojos encontrados: 2
Número de objetos verdes encontrados: 1
Total, de objetos encontrados: 6

Con base en estos datos el sistema llama de la carpeta de almacenamiento el archivo llamado 3216.txt el cual, como se observa en la figura 4, contiene los siguientes datos:

Universidad Politécnica de Durango
Ingeniería en Telemática
R. Villanueva Vallejo
A.C. Herrera Guerrero

Cuando el sistema vuelva a detectar el mismo número de objetos volverá a mostrar el texto anterior en pantalla. La ventaja del algoritmo de descifrado es que no requiere alineamientos de posición [6] esto le permite al usuario colocar las figuras en la forma más conveniente posible, además de que la matriz puede ser de cualquier tamaño.

2.6 Resultados

El sistema está desarrollado en MatLab®, por lo tanto, este toma como primera opción para la captura de la imagen, la cámara web del equipo de cómputo, no obstante, debido a que los pixeles que manejan la mayoría de las cámaras web son insuficientes, es necesaria una cámara de mayor resolución.

```
fot=videoinput('winvideo',2,res);
```

La línea de código anterior permite seleccionar alguno de los adaptadores disponibles que tiene MatLab®, por lo general el winvideo 1 corresponde a la cámara web del equipo de cómputo, por lo que, se utiliza winvideo 2, relacionando la aplicación DroidCam para enlazar, mediante dirección IP, la cámara de un smartphone Android con la computadora.

Figura 4. Pantalla de DroidCam en Android

Con el fin de que los colores fueran captados de una manera más rápida por el sistema, se diseñó una caja blanca de madera con el propósito de aislar la imagen de la luz natural, en el interior de dicha caja fueron colocadas tiras de luces led color blancas para iluminar el código y facilitar la captura de las imágenes como se muestra en la figura 5.

Figura 5. Interior de la caja

Después de enlazar la cámara del teléfono móvil con el sistema, se procede a colocar la combinación de colores y número de figuras deseadas dentro de la caja como se muestra en la figura 6.

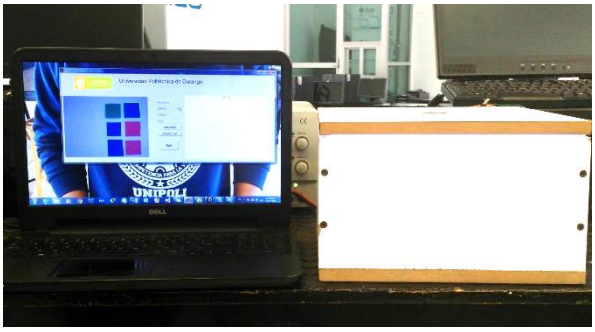

Figura 6. Captura de imagen

El sistema llama el archivo correspondiente con el número de figuras y colores detectados (ver figura 7).

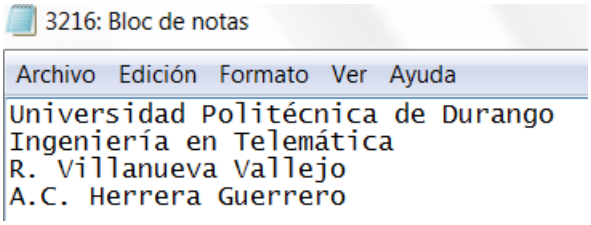

Figura 7. Archivo .txt

2.7 Referencias

[1] ionlitio.com (2007). El código QR. Disponible en: https://www.ionlitio.com/el-codigo-qr/

[2] González, J., et al. (2016). Códigos QR y sus aplicaciones en las ciencias de la salud. Recuperado de: https://www.redalyc.org/pdf/3776/377645765009.pdf

[3] De la Fuente, E., Trespaderne, F. (s.f.). Imágenes binarias. Recuperado de: http://www.librovision.eii.uva.es/pdf/cap4.pdf

[4] MathWorks. (2021). Detección de bordes. Disponible en: https://es.mathworks.com/help/images/edge-detection.html

[5] Alonso, M. (2009). Espacios de Color RGB, HSI y sus Generalizaciones a n-Dimensiones. Recuperado de: https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/362/1/AlonsoPeMA.pdf

[6] Sanz, A. (2013). Reconocimiento Automático de Formas. Recuperado de: http://diposit.ub.edu/dspace/bitstream/2445/48925/2/memoria.pdf