

MEJORA DE LA APARIENCIA VISUAL DE CÓDIGOS DE BARRAS BIDIMENSIONALES, QR-CODES, APLICANDO LA TÉCNICA DE TONOS MEDIOS

IMPROVEMENT OF THE AESTHETIC APPEARANCE OF BIDIMENSIONAL BARCODES - (QR-CODES) BY APPLYING HALFTONE TECHNIQUES



AUTOR

OFELIA PATRICIA VILLARREAL DULCEY
Magister (c) en Ingeniería Electrónica
*Universidad Industrial de Santander
Investigadora, Grupo de Investigación
en Control, Electrónica, Modelado y
Simulación – CEMOS
ofelia.villarreal@correo.uis.edu.co
COLOMBIA

AUTOR

RODOLFO VILLAMIZAR MEJÍA
Doctor en Tecnologías de la Información
*Universidad Industrial de Santander
Docente Investigador, Grupo de
Investigación en Control, Electrónica,
Modelado y Simulación – CEMOS
rovillam@uis.edu.co
COLOMBIA

*INSTITUCIÓN

Universidad Industrial de Santander
UIS
Universidad Pública
Carrera 9, calle 27, ciudad universitaria
Bucaramanga
vacadem@uis.edu.co
COLOMBIA

INFORMACIÓN DE LA INVESTIGACIÓN O DEL PROYECTO: Este trabajo es uno de los productos desarrollados en el proyecto de maestría customized QR code by integrating chromatic images based on halftone techniques realizado por la ingeniera Ofelia P. Villarreal Dulcey. El proyecto dio inicio en el mes de septiembre del año 2012 y durante el mes de junio del año 2013 se obtuvo los resultados de la primera versión. El proyecto se desarrollo en el grupo de investigación en Control, Electrónica, Modelado y Simulación, CEMOS de la Universidad Industrial de Santander.

RECEPCIÓN: Agosto 1 de 2013

ACEPTACIÓN: Diciembre 12 de 2013

TEMÁTICA: Ingeniería Eléctrica, Electrónica, Telecomunicaciones y Telemática: Procesamiento Digital de Imágenes

TIPO DE ARTÍCULO: Artículo de Investigación Científica e Innovación

Forma de citar: Villarreal Dulcey, O. P., & Villamizar Mejía, R. (2014). Mejora de la apariencia visual de códigos de barras bidimensionales, QR-CODES, aplicando la técnica de tonos medios. En R, Llamosa Villalba (Ed.). Revista Gerencia Tecnológica Informática, 13(35), 33-45. ISSN 1657-8236.

RESUMEN ANALÍTICO

Los códigos de barras son imágenes binarias que contienen datos codificados. Su versión bidimensional más reciente conocida como código de rápida respuesta, QR (*Quick Response*) ofrece mayor ventaja respecto a otros códigos de esta clase en términos de capacidad de almacenamiento, tipos de datos a codificar y rapidez en decodificación. Recientemente, los códigos QR han recibido especial atención como alternativa publicitaria debido a que es posible decodificar la información mediante el uso de teléfonos inteligentes. Esta característica impulsó el uso de códigos QR como un medio rápido, sencillo y económico de conectar los usuarios de telefonía móvil con servicios, información y negocios en línea. Debido a su naturaleza binaria el código QR no es visualmente atractivo al usuario, ni aporta información directa sobre el tipo de contenido codificado en el mismo. Por tanto en este trabajo presenta una alternativa para mejorar la apariencia visual de los códigos QR, sin perder la calidad de la información o modificar la estructura del mismo. La propuesta consiste en técnicas de tonos medios para incrustar una imagen visualmente definida en el código QR, así como explorar las características de los algoritmos de decodificación de los mismos. El algoritmo toma ventaja del proceso de decodificación basado en la detección de la luminancia. Los valores de luminancia de la imagen a ser incrustada son modificados de acuerdo con la imagen del código QR. El resultado final de los algoritmos desarrollados es la incrustación de la imagen dentro del código QR. La distribución de píxeles a ser modificados es realizada aplicando máscaras de ruido azul. Los códigos QR generados son compatibles con lectores de códigos de barras bidimensionales estándar. Resultados experimentales muestran una mejora significativa de la apariencia visual de los códigos QR con tasas de decodificación similares a las de la versión dicromática original.

PALABRAS CLAVES: Procesamiento digital de imágenes, Ruido azul, Medios tonos, Códigos QR, Librería zxing.

ANALYTICAL SUMMARY

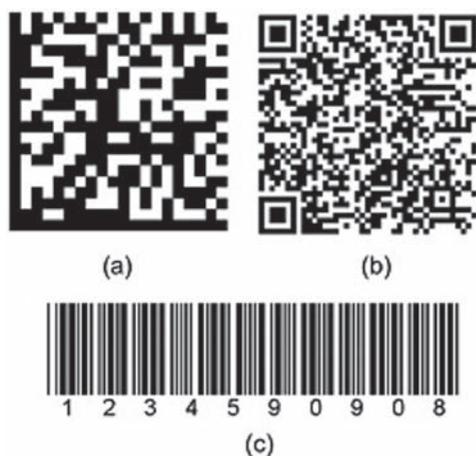
Barcodes are binary images containing encoded data. The most recent two-dimensional version known as quick response code, QR (*Quick Response*) offers greater advantage than another similar codes in terms of storage capacity, data rates and faster decoding process. Recently, QR codes have received special attention as an alternative for advertising due to the possibility of decoding the information contained in the QR code, by using smartphones. This feature prompted the use of QR codes as a quick, simple and economical way to connect online mobile users with services, information and businesses. However, due to binary nature of QR code, it is not visually pleasing for users, and visual explicit information is not presented. Thus, this paper presents an alternative approach in order to improve the visual appearance of QR codes by no losing information or modifying its structure. The proposed method exploits both halftone techniques, in order to embed a visually defined image into a QR code, and also the characteristics of the decoding algorithms. This algorithm takes advantage by using the decoding process based on the luminance detection. The luminance values of image to be embedded are modified according to the QR code image. The final result of the proposed algorithm is a standard QR code, visually improved with embedded image. The distribution of pixels to be modified is performed by using blue noise masks. The generated QR codes are compatible with standard two-dimensional barcode readers. Experimental results show a significant improvement in the visual appearance of QR codes with decoding rates similar to those achieved with the original dichromatic version.

KEYWORDS: Digital Image Processing, Blue noise, Halftones, QR codes, Zxing library.

INTRODUCCIÓN.

Los códigos de barras han sido adoptados de manera global debido a la necesidad de agilizar procesos en la industria relacionados con actividades de inventario e identificación, así como la disminución de la probabilidad de errores relacionados con digitación. En esta reciente década el código de barras tradicional ha evolucionado a versiones bidimensionales de las cuales existen diversas clases como son los códigos *PDF147*, *MAXICODE*, *DATAMATRIX*, y códigos *QR* [1-4]. En la figura 1 se puede distinguir códigos unidimensionales de los códigos bidimensionales. Particularmente, se ha destacado el código de barras de rápida respuesta o código *QR* (*Quick Response*), como la evolución de otros códigos bidimensionales, específicamente de los códigos *PDF147* el cual permite una alta capacidad de almacenamiento de información y el código *MAXI* que maneja una alta velocidad en la decodificación. Además de estas características de almacenamiento y velocidad, otra característica de interés en los códigos *QR* es la eficiencia en la codificación de caracteres japoneses conocidos como caracteres *Kanji* o *Kana* [7]. La inserción de estos caracteres ha contribuido a la utilización masiva de los códigos *QR* en la industria japonesa bajo el estándar *JIS* (*Japanese Industrial Standards*) – *XO510*, en 1999. Finalmente, en el año 2000 los códigos *QR* se convirtieron en el estándar internacional *ISO/IEC18004* [4].

FIGURA 1. Códigos de barras. a) Código de barras bidimensional tipo *DATAMATRIX* b) Código de barras bidimensional tipo código *QR* c) Código de barras unidimensional tipo code 39.



Fuente: a) [3]. b) [5]. c) [6].

La difusión de los códigos *QR* es debida en gran parte al auge de los teléfonos inteligentes "*smarthphones*", con los cuales es posible adquirir la imagen del código *QR* mediante el uso de la cámara integrada en el celular

y posteriormente realizar procesos de decodificación de la información de manera rápida, así como, acceder a páginas web o descargar contenido en línea. Entre las principales ventajas que ofrecen los códigos *QR* se encuentran su fácil uso, alta capacidad de almacenamiento y velocidad en la decodificación [8]. En los códigos *QR*, la velocidad en la decodificación es alcanzada usando patrones especiales para localización y alineamiento. Estos patrones permiten detectar de forma rápida la orientación correcta de la imagen y las distorsiones geométricas introducidas debido a superficies irregulares o ángulos por la posición del teléfono respecto a la imagen. Los patrones de localización se encuentran ubicados en tres de las cuatro esquinas del código *QR* [4], como es indicado en la figura 3. Estos y otros patrones usados para alineamiento son contemplados en el estándar y descritos en secciones posteriores.

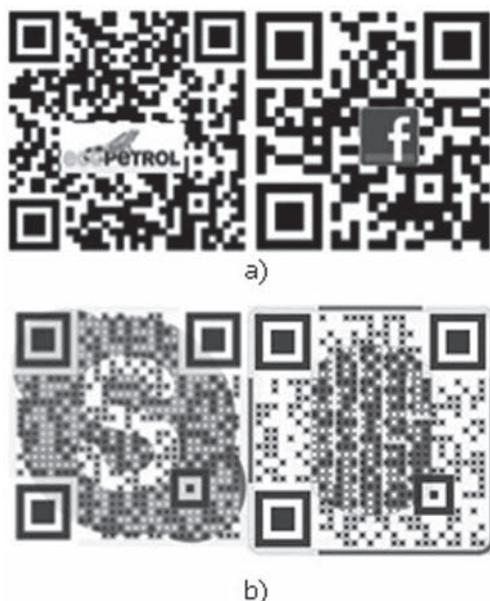
Como se indica en la figura 3, los códigos *QR* están conformados por un conjunto de módulos cuadrados blancos y negros. Esta imagen binaria no permite identificar el tipo de información codificada, por tanto los usuarios no pueden discriminar visualmente si la información codificada en el código *QR* es de su interés. Numerosas modificaciones se han realizado para incorporar imágenes en los códigos *QR* de forma que los usuarios puedan relacionar fácilmente el código *QR* con un producto o servicio [9], [5]. Algunas modificaciones están basadas en la eliminación de los módulos de información redundante del código *QR* usada para corrección de errores. El espacio adquirido al realizar la eliminación de los módulos es usado para insertar imágenes con las que el usuario puede identificar el tipo de contenido en el código *QR*, como se muestra en la figura 2.a [10].

La técnica da una solución al problema de identificar el tipo de información en código *QR*, sin embargo, con esta solución se sacrifica la robustez del código. Al eliminar la información redundante, no es posible asegurar la decodificación de la información. Otra aproximación se basa en la manipulación de los valores de luminancia, como se muestra en la figura 2.b. En donde el valor del módulo en el código *QR* es identificable por el decodificador siempre que se mantenga una pequeña porción en el centro del módulo blanca o negra dependiendo del valor asignado en el código *QR*; el resto de píxeles en el módulo conserva el color de la imagen [5].

Este procedimiento no utiliza la información redundante del código *QR*, por lo que preserva la robustez en la decodificación. En este trabajo se busca mejorar la apariencia estética de los códigos *QR*, para ello se introduce una nueva técnica para grabar una imagen en un código *QR* y con ello mejorar no solo la

aparición visual del código QR sino también facilitar la identificación de la información contenida en el mismo. La nueva técnica desarrollada utiliza la manipulación de los valores de luminancia de la imagen a grabar, así como también máscaras que obedecen a una distribución de ruido azul. Los niveles de luminancia así como las restricciones son determinados por el proceso de binarización que se aplica en la decodificación. Por otro lado el desarrollo de las máscaras está basado en la teoría de medios tonos usada principalmente en técnicas usadas en la industria de la impresión.

FIGURA 2. Códigos QR con imágenes incrustadas. En la figura a) se ha eliminado una pequeña área de módulos blancos y negros para incrustar la imagen del logo. En la figura b) se ha manipulado el tamaño del módulo de la imagen y la luminancia.



Fuente: Figura a) Autores. b) La imagen es obtenida usando el algoritmo en [5].

El artículo está organizado como sigue: En la sección uno se realiza una descripción de la estructura de los códigos QR así como sus características principales. En la sección dos se describen las características de la binarización usada en la decodificación. En la sección tres se describe las características de la máscara de ruido azul y se determinan los valores de luminancia óptimos para alcanzar la decodificación de los datos a pesar de las modificaciones realizadas. La sección cuatro, presenta el procedimiento a realizar para grabar las imágenes en los códigos QR. La sección cinco presentan los resultados obtenidos aplicando el procedimiento descrito en la sección cuatro. En las secciones restantes se proveen las conclusiones y referencias asociadas al documento.

1. ESTRUCTURA Y CARACTERÍSTICAS DE LOS CÓDIGOS QR.

Los códigos QR están diseñados para agilizar el proceso de identificación y decodificación, para ello se basa en el uso de patrones especiales [4]. Estos patrones pueden ser clasificados en dos grupos, el primero llamado patrones de funcionamiento, está conformado por patrones de localización, alineamiento, sincronización y una zona de reserva. Al segundo grupo, pertenece la zona dedicada a la codificación de los datos en los módulos blancos y negros. Los códigos QR pueden variar en tamaño y versión, bajo el estándar se encuentran cuarenta versiones diferentes de códigos QR que van desde un tamaño de 21 x 21 módulos hasta un tamaño de 177 x 177 módulos [4]. Los patrones de funcionamiento se conservan fijos a pesar de los cambios en tamaño o versión para favorecer la detección y decodificación del código QR de forma rápida.

1.1 PATRONES DE FUNCIONAMIENTO.

Los códigos QR son diseñados para operar con dispositivos de escaneo de alta velocidad. Alcanzar una identificación y decodificación a alta velocidad es posible gracias al uso de los patrones de funcionamiento. Estos patrones son usados para ubicar, rotar, y alinear el código QR, así como también delimitar la región de codificación. Los patrones de funcionamiento son fijos y de la precisión en la identificación de estos depende la decodificación de la información. Estos patrones se dividen en patrones de localización, alineamiento y sincronización.

FIGURA 3. Características de los códigos QR. a) La zona de reserva rodea el símbolo e introduce espacio entre este y otros objetos alrededor. b) Patrón de localización, permite ubicar rápidamente la posición correcta del código QR. c) Patrón de alineamiento, ayuda a corregir distorsiones. d) Patrón de sincronización, ayuda a detectar con precisión la cuadrícula en el código QR. e) Módulo de información, representan la información codificada, grupos de 8 módulos conforman 1 byte de información.



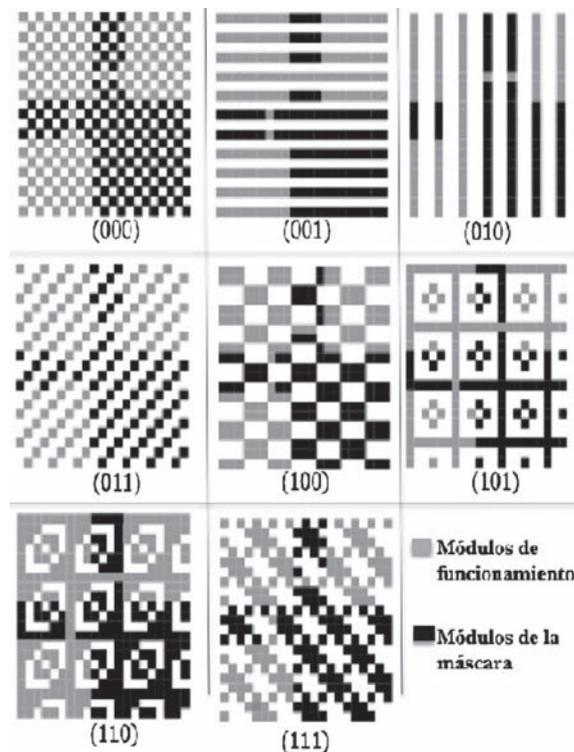
Los patrones de localización están ubicados en tres de las cuatro esquinas del código QR, visualmente se pueden identificar como tres cuadrados de gran tamaño tienen una longitud de 1:1:3:1:1 pixeles de información, como se observa en la figura 3.b. Estos patrones son usados para facilitar la identificación de la posición correcta del código QR para decodificación. Por tanto, la posición de estos patrones es fija dentro del código QR y la configuración de sus módulos es irrepetible, disminuyendo con estas restricciones la probabilidad de error debida a duplicidad de los patrones. Los patrones de alineamiento se encuentran ubicados a lo largo de la región de codificación, según la versión del código QR se puede encontrar uno o muchos de estos patrones. En el caso de distorsión de la imagen se utilizan para reasignar los módulos de información en su posición original. También en estos casos es necesario identificar las coordenadas de cada fila y columna de la matriz, para ello el código QR cuenta con patrones de sincronización, estos patrones están formados por módulos blancos y negros ubicados de forma alterna en una fila y una columna a lo largo del código QR, como se observa en la figura 3.d.

1.2 REGIÓN DE CODIFICACIÓN.

La región de codificación, ubicada al interior del código QR, está delimitada por los patrones de localización. Esta región es destinada para contener la información a codificar. Además de los datos, en esta región se codifica la información correspondiente a la versión y formato, así como la información redundante utilizada en la corrección de errores. De acuerdo al estándar la información a codificar es transformada en un conjunto de bits los cuales son divididos en palabras de código, manteniendo un tamaño de 8 bits. Estas palabras de código ó módulos de información, son representados en el código QR como cuadrados blancos y negros, como se muestra en la figura 3.c. Una vez divididas en el tamaño especificado anteriormente, las palabras de código son organizadas en forma secuencial en la región de codificación. La configuración de los módulos de información en el código QR deben evitar la réplica de los patrones de funcionamiento, el estándar propone alterar el orden del código mediante el uso de máscaras. Las máscaras son aplicadas únicamente a la región de codificación. Los módulos en esta región son transformados aplicando la operación XOR entre cada una de las máscaras y el código QR. El estándar propone un total de ocho máscaras, como se observa en la figura 4.

Las máscaras son generadas cada una siguiendo cierta condición, donde se define como negro cualquier módulo para el cual la condición se cumple. En la tabla 1. se definen las condiciones para cada una de las máscaras.

FIGURA 4. Tipos de máscaras propuestos por el estándar.



Fuente: [4]

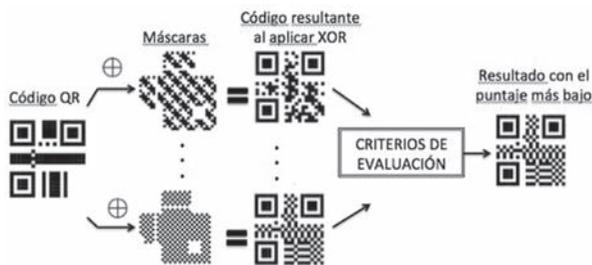
Los patrones resultantes son evaluados asignando puntaje al patrón resultante cada vez que se encuentre algunos de los siguientes criterios: i) Módulos del mismo color en filas/columnas. ii) Bloque de módulos del mismo color. iii) Módulos alternados siguiendo la configuración de los patrones de localización con radio 1:1:3:1:1. iv) La proporción de los módulos negros en todo el código QR es mayor a 50%. Finalmente, el patrón resultante con menor puntaje es seleccionado, como se muestra en la figura 5.

Tabla 1. Condiciones para generar la máscara.

Referencia de la Máscara	Condición
000	$(i,j) \bmod 2 = 0$
001	$i \bmod 2 = 0$
010	$j \bmod 3 = 0$
011	$(i + j) \bmod 3 = 0$
100	$((i \bmod 2) + (i \bmod 3)) \bmod 2 = 0$
101	$(i,j) \bmod 2 + (i,j) \bmod 3 = 0$
110	$((i,j) \bmod 2 + (i,j) \bmod 3) \bmod 2 = 0$
111	$((i,j) \bmod 3 + (i + j) \bmod 2) \bmod 3 = 0$

Fuente: [4]

FIGURA 5. Pasos para aplicar las máscaras al código QR.



1.3 CAPACIDAD DE ALMACENAMIENTO Y CORRECCIÓN DE ERRORES.

La robustez de los códigos de barras se ha concentrado principalmente en favorecer el incremento en la capacidad de almacenamiento de información, así como en una mayor resistencia a errores en la decodificación, normalmente generados durante la adquisición de la imagen. Los códigos QR tienen la capacidad de almacenar miles de dígitos, sin embargo, esta capacidad se ve limitada por el tipo de caracteres, versión, formato y nivel de corrección de errores. Un código QR configurado a su máxima capacidad de almacenamiento puede contener hasta 7089 caracteres de tipo numérico, hasta 4296 datos de tipo alfanumérico, hasta 2953 datos tipo byte y hasta 1817 caracteres Japoneses tipo *Kanji* o *Kana*. En cuanto a la versión los códigos QR están disponibles en cuarenta versiones, donde cada versión determina el número de módulos por símbolo, por ejemplo, la versión uno corresponde a un código QR de tamaño 21 x 21 módulos, mientras que la versión cuarenta corresponde a un código QR de tamaño 177 x 177 módulos. Por otro lado, los códigos QR utilizan códigos Reed-Solomon, para proveer corrección de errores durante la decodificación de los datos. En la mayoría de los casos al adquirir la imagen del código QR para la decodificación, se presenta ruido, así como distorsión y en algunos casos daño parcial de la imagen. Los códigos QR manejan cuatro niveles de corrección de errores, L, M, Q y H, donde cada nivel permite recobrar respectivamente el 7%, 15%, 20% y 30%, de información perdida o dañada en un código QR. La cantidad de información redundante en el código QR es determinada por el nivel de corrección de errores escogido.

2. DECODIFICACIÓN.

El procedimiento para la decodificación de los datos se divide en tres etapas, binarización, detección y decodificación de los datos. El enfoque de este trabajo se basa en mejorar la apariencia visual del código QR mediante la incrustación de una imagen en el código

QR. Para este fin se realiza la manipulación de los valores de luminancia de la imagen a incrustar con respecto a los valores de luminancia que se percibirían el decodificador. Por tanto, el proceso de decodificación no es modificado. En este sentido los parámetros y técnicas usadas en la etapa de binarización son de particular interés. Al conocer el procedimiento aplicado en la etapa de binarización, es posible optimizar los valores de luminancia y la distribución de píxeles en la imagen de código QR previo a ser adquirida por la etapa de decodificación. En la etapa de binarización la imagen adquirida en escala de grises es segmentada en píxeles blancos y negros. Una vez binarizada la imagen es posible identificar los módulos de información que deben ser extraídos, corregidos y decodificados. La salida de un proceso de umbralización está dada por,

$$Q(i, j) = \begin{cases} 1 & \text{si } Y_{i,j} > t_{i,j} \\ 0 & \text{si } Y_{i,j} \leq t_{i,j} \end{cases} \quad (1)$$

Donde Y es la imagen en escala de grises adquirida por la cámara, $t_{i,j}$ es el umbral asignado a i, j y $Q(i, j)$ es la salida binaria. El proceso de binarización tiene un papel fundamental en la decodificación, dado que una mala interpretación de los módulos conlleva a que la información no pueda ser correctamente decodificada. Esta tarea es particularmente compleja cuando hay variaciones en las condiciones de iluminación o exceso de ruido. La binarización puede ser realizada aplicando diversos métodos, el estándar sugiere el uso de umbralización global, sin embargo; la mayoría de aplicaciones emplean métodos de binarización mucho más eficientes. [20, 13]

2.1 UMBRALES GLOBALES.

En una umbralización global, el valor del umbral es determinado usando todos los píxeles de la imagen. Entre los métodos implementados con éxito anteriormente en la binarización de un código QR se encuentra, por ejemplo, el método aplicado por Otsu en [11], el cual es uno de los más efectivos cuando el histograma de la imagen es aproximadamente bimodal. El método sugerido por Otsu tiene una complejidad relativamente baja, lo que es positivo en términos de velocidad, sin embargo, el éxito de este método se basa en que la iluminación al tiempo de la captura de la imagen sea uniforme, lo cual no es garantizable en la mayoría de los casos, puesto que las imágenes no son adquiridas en ambientes controlados. Por otra parte, métodos sencillos para umbralización como son la media y la mediana obtienen resultados ligeramente peores a los obtenidos con el método de Otsu, pero son resultados aceptables siempre y cuando el contraste de la imagen sea alto.

2.2 UMBRALES LOCALES.

El cálculo de umbrales usando el método local, se realiza utilizando pequeñas regiones de la imagen, por lo cual esta estrategia es mucho más adecuada cuando hay variaciones en la iluminación. *Niblack* define un método de umbralización local en [12]. Donde los umbrales son definidos como,

$$t_{i,j} = \bar{Y}_{B_{i,j}} + k \sqrt{\text{var}(Y_{B_{i,j}})} \quad (2)$$

Donde $B_{(i,j)}$ es un recuadro centrado alrededor del pixel (i, j) y $\bar{Y}_{B_{i,j}}$ es igual a la media de $Y_{B_{i,j}}$ y $\text{var}()$ es una función que calcula la varianza.

Otro método conocido para umbralización local es el desarrollado por *Sauvola* en [13]. Los umbrales son definidos como,

$$t_{i,j} = \bar{Y}_{B_{i,j}} + \left[1 + k \left(1 - \frac{\sqrt{\text{var}(Y_{B_{i,j}})}}{R} \right) \right] \quad (3)$$

Otros métodos aplican funciones como la media, la mediana, gris medio o métodos como el desarrollado por *Yang* en [14] utilizan umbrales adaptativos en los cuales el tamaño de la ventana cambia basado en características locales.

2.3 PROCESO DE BINARIZACIÓN.

La librería *zxing* es uno de los proyectos de código abierto mejor conocidos para generar y decodificar códigos QR [15]. En este trabajo se toma como base el proceso de binarización aplicado por la librería *zxing* para determinar los parámetros óptimos de luminancia que deben ser usados en la imagen del código QR. La optimización de los parámetros de luminancia se realiza a fin de asegurar una correcta binarización en el lado del decodificador. El proceso de binarización usado en la librería *zxing* utiliza los valores de luminancia promedio de los bloques en una ventana. La imagen capturada es dividida en bloques $B_{m,n}$ de tamaño 8 x 8 pixeles. A su vez estos bloques son agrupados en ventanas de tamaño 5 x 5 bloques. La ventana se desplaza por la imagen de manera que la luminancia promedio de la ventana es calculada para cada bloque, con la siguiente ecuación,

$$u_{m,n} = \frac{1}{25 \times 64} \sum_{p=m-2}^{p=m+2} \sum_{q=n-2}^{q=n+2} \sum_{(k,l) \in B_{p,q}} Y[k,l] \quad (4)$$

Esta técnica resulta computacionalmente menos costosa, sin embargo, de menor rendimiento en comparación con los métodos de *Niblack* o *Sauvola*.

3. TÉCNICA APLICADA AL MEJORAMIENTO ESTÉTICO DE LOS CÓDIGOS QR.

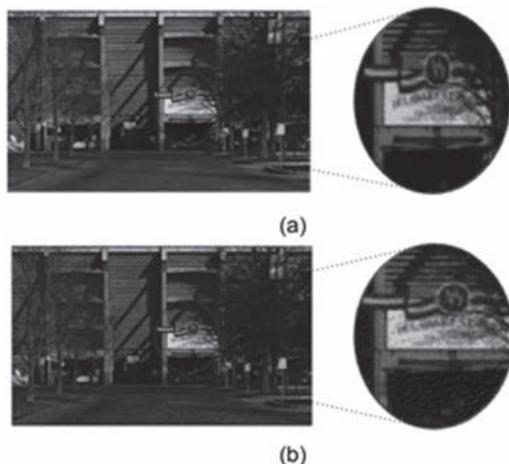
La técnica propuesta requiere de tres entradas, i) una máscara de ruido azul, usada para definir los pixeles cuya luminancia se debe modificar. ii) La imagen del código QR con la información codificada. iii) La imagen con la cual se desea caracterizar el código QR. A continuación se describen las etapas del algoritmo implementado.

3.1 SELECCIÓN DE PÍXELES.

La técnica de tonos medios ha sido ampliamente estudiada y usada en la industria de la impresión digital [21]. Principalmente, consiste en convertir un tono continuo de una imagen a un patrón binario de puntos cuya distribución aprovecha las características de no-linearidad del sistema de visión humano para dar la sensación de tono continuo, como se muestra en la figura 6. El efecto visual que produce la distribución de puntos equidistantes ha sido ampliamente estudiado en el campo de medios tonos digitales y es aplicado en este trabajo para generar la ilusión de suavidad en las transiciones entre la imagen a incrustar y el código QR. Si los pixeles a modificar en la imagen son aleatorios pero están uniformemente distribuidos espacialmente, las transiciones bruscas entre la imagen a incrustar y el código QR son minimizadas, esto es posible, debido a que este tipo de patrones concentran la mayoría de su energía en las frecuencias altas donde el sistema visual humano es menos sensible. Esta clase de distribución es conocida como ruido azul, y es de gran importancia en el desarrollo de técnicas digitales de tonos medios usadas en impresión [17, 22]. Diversos algoritmos han sido propuestos para generar estos patrones de ruido azul, entre estos se encuentran, difusión de errores, máscaras de ruido azul y búsqueda binaria directa [16, 19, 23].

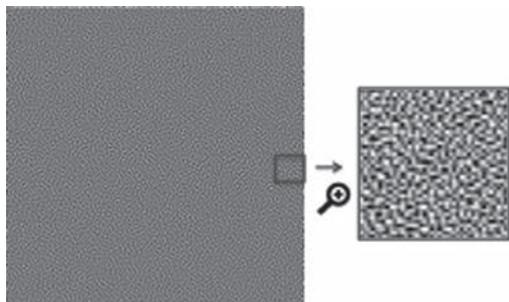
Las máscaras de ruido azul son preferidas a otras técnicas debido a que los patrones pueden ser computados en paralelo con la imagen de interés, lo cual incrementa la velocidad del proceso en comparación con otros algoritmos como difusión de errores que al ser de tipo recursivo incrementan la complejidad del proceso cuando se aplican a imágenes de gran tamaño.

FIGURA 6. a) Imagen Original. b) Imagen a medio tono aplicando difusión de errores.



El diseño de máscaras de ruido azul puede ser desarrollado por diversos algoritmos. En la literatura se encuentran por ejemplo, los algoritmos de búsqueda binaria directa, máscaras *green-noise*, cluster o grupo [16, 17, 18, 19].

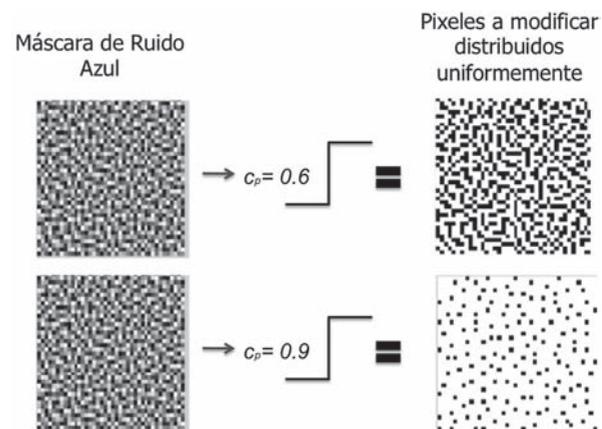
FIGURA 7. Máscara con distribución de ruido azul.



El diseño de la máscara es realizado previamente al proceso de incrustación de la imagen por lo que la complejidad del algoritmo no afecta la velocidad del proceso de incrustación de la imagen en el código QR. En la figura 7, se observa un ejemplo de la máscara de ruido azul utilizada en este trabajo. Para obtener esta máscara se implementó en la herramienta de simulación Matlab el algoritmo para generación de máscara de ruido azul propuesto en [23].

Como paso inicial del proceso se debe seleccionar una cantidad de pixeles de la imagen para modificar su valor de luminancia. Estos pixeles se distribuyen siguiendo la máscara de ruido azul diseñada previamente, la cual utiliza un umbral fijo igual a la concentración de pixeles que se deben modificar c_p , el patrón binario resultante es denotado por H_{c_p} , como se muestra en la figura 8.

FIGURA 8. La cantidad de pixeles de la imagen a ser modificados c_p . Los pixeles son distribuidos de manera uniforme en la imagen siguiendo el patrón dispuesto por la máscara de ruido azul.



En cuanto a los valores de la concentración de pixeles, se tiene que, un valor c_p cercano a uno $c_p \approx 1$, hace que la imagen resultante se asemeje al código QR, mientras que con un valor cercano a cero $c_p \approx 0$ la imagen resultante se asemeja a la imagen original, como se muestra en la figura 12. El valor óptimo de c_p es obtenido en el proceso de optimización y al alcanzar el valor óptimo, la imagen resultante presenta transiciones suaves que permiten distinguir la imagen original de la imagen del código QR, como se observa en la figura 9.

3.2 AJUSTE DE LOS VALORES DE LUMINANCIA EN LA IMAGEN ORIGINAL.

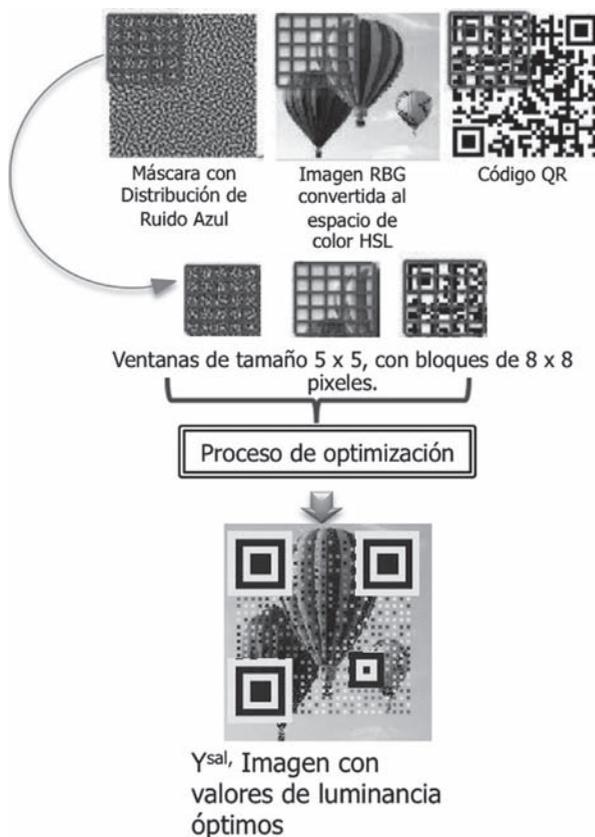
Para alcanzar la decodificación de un código QR, idealmente, solo los pixeles en la parte central de los módulos en el código son importantes. Sin embargo, debido a diversos factores como la distorsión, ángulos de inclinación inducidos y error a la hora de capturar la imagen, los pixeles restantes en el módulo juegan un papel importante.

Una vez establecida la proporción y distribución de los pixeles en la imagen a ser modificados, se procede a modificar los valores de luminancia de acuerdo a uno de los cuatro posibles niveles descritos en la ecuación (5).

Los valores de luminancia en la imagen original son cambiados de acuerdo a la clase de módulo (blanco o negro) en esa posición en el código QR. Sin embargo, no todos los pixeles han de ser modificados; para estos pixeles los valores de luminancia no son alterados, los valores de luminancia son asignados según el caso, como se define a continuación:

$$Y^{sal}[i,j] = \begin{cases} \beta & \text{if } Mk_{i,j} = 0, Q_{i,j} = 1, H_{cp} = 1 \\ \rho & \text{if } Mk_{i,j} = 0, Q_{i,j} = 0, H_{cp} = 1 \\ \beta_c & \text{if } Mk_{i,j} = 1, Q_{i,j} = 1 \\ \rho_c & \text{if } Mk_{i,j} = 1, Q_{i,j} = 0 \\ Y_{i,j}^{sal} & \text{otherwise} \end{cases} \quad (5)$$

FIGURA 9. Selección del área (ventanas) local para el cálculo de la luminancia y distribución óptima de píxeles.



Donde $\beta, \rho, \beta_c, \rho_c$ son los cuatro posibles niveles a los cuales la luminancia debe ser modificada según el criterio dado por $Q_{i,j}$ y donde $Y_{i,j}^{sal}$ es el valor de luminancia resultante de modificar el píxel en la posición (i, j) ; $Q_{i,j}$ es un píxel del código QR (valor 1 ó 0) en la posición (i, j) . Cabe destacar que esta modificación es definida solo sobre los valores de luminancia de la imagen, no en todo el espacio de color RGB. En este sentido, la información de color de cada píxel se conserva, y los valores de luminancia son modificados en otro espacio de color. En este trabajo se ha escogido el espacio de color HSL (*Hue, saturation, luminance*) debido a la baja complejidad computacional que emplea en comparación con otros espacios de color. De esta manera, cada píxel a ser modificado debe ser primero transformado del espacio RGB al espacio HSL, $[H, S, L] = T[R, G, B]$.

Una vez el píxel esté en el espacio HSL, se modifica el componente de luminancia manteniendo fijos los valores de saturación (S) y sombra (H) hasta alcanzar el valor de luminancia deseado. La definición de luminancia usada en este trabajo está dada por,

$$Y = 0.2989 R + 0.5870 G + 0.1140 B \quad (6)$$

Al fijar los valores de sombra (H) y saturación (S), se puede escribir R, G, B como la función de L dada por,

$$Y = f(L), \quad (7)$$

Donde la relación entre Y y el componente de luminancia en el espacio HSL es una función lineal, monotónica que se puede expresar como,

$$Y = f(L) = v^T T^{-1}(H, S, L), \quad (8)$$

Donde,

$$v = [0.2989 \ 0.5870 \ 0.1140]^T$$

$$Y = v^T (R, G, B)$$

$$[H, S, L] = T[R, G, B]$$

$$L = \frac{\min(R, G, B) + \max(R, G, B)}{2}$$

y f abarca el rango completo de $Y \in [0,1]$. El valor óptimo de L es obtenido de la luminancia objetivo $Y_{i,j}^{sal}$ como,

$$L^* = \underset{L}{\operatorname{argmin}} |f(L) - Y^{sal}| \quad (9)$$

Los nuevos componentes RGB para los píxeles con luminancia óptima, son obtenidos así

$$(R^*, G^*, B^*) = T(H, S, L^*) \quad (10)$$

3.3 CÁLCULO DE VALORES ÓPTIMOS DE LUMINANCIA.

Los niveles de luminancia óptimos y la cantidad de píxeles a modificar c_p son valores que se han calculado basados en la minimización de la probabilidad de error en la etapa de binarización. Para el cálculo de la probabilidad de error se debe tener en cuenta que la imagen obtenida con el algoritmo es adquirida usando la cámara integrada de un teléfono inteligente. Una vez adquirida, la imagen será binarizada según el umbral $u_{i,j}$ definido en la ecuación (4). La imagen binaria es obtenida al comparar el valor de luminancia en una posición (i, j) con el valor del umbral $u_{i,j}$.

$$Q_{i,j} = \begin{cases} 1 & \text{si } Y_{i,j} > u_{i,j} \\ 0 & \text{si } Y_{i,j} < u_{i,j} \end{cases} \quad (11)$$

El proceso de mejora de la apariencia visual del código QR puede introducir distorsiones o ruido incrementando la probabilidad de errores en la etapa de binarización. Esta probabilidad puede ser reducida optimizando los valores de luminancia, siendo consiente de que esta optimización implica un intercambio entre la robustez en la etapa de decodificación y la calidad visual de la imagen. Luego entonces, se tiene que la probabilidad de error en la etapa de binarización P_{be} está dada por,

$$P_{be(i,j)} = P(\text{definir } Q_{i,j} = 1 | Q_{i,j} = 0)P(Q_{i,j} = 0) + P(\text{definir } Q_{i,j} = 0 | Q_{i,j} = 1)P(Q_{i,j} = 1) \quad (12)$$

Si se escribe esta probabilidad en la ecuación (12) en términos de luminancia se puede decir que,

$$P(\text{definir } Q_{i,j} = 1 | Q_{i,j} = 0) = P(Y_{i,j}^{sal} > u_{i,j} | Q_{i,j} = 0)$$

$$P(\text{definir } Q_{i,j} = 0 | Q_{i,j} = 1) = P(Y_{i,j}^{sal} < u_{i,j} | Q_{i,j} = 1)$$

Del mismo modo se puede expresar estas probabilidades en función de los pixeles a ser modificados, como sigue,

$$\begin{aligned} P(Y_{i,j}^{sal} > u_{i,j} | Q_{i,j} = 0) &= P(\beta > u_{i,j})c_p \\ + P(Y_{i,j} > u_{i,j})(1-c_p)P(Y_{i,j}^{sal} < u_{i,j} | Q_{i,j} = 1) & \quad (13) \\ = P(\rho < u_{i,j})c_p + P(Y_{i,j} < u_{i,j})(1-c_p) \end{aligned}$$

Reemplazando las ecuaciones en (13) en la definición de probabilidad de fallo p_f se tiene que,

$$\begin{aligned} P_{be} &= [P(Q_{i,j} = 0)(P(u_{i,j} < \beta) - P(Q_{i,j} = 1)P(u_{i,j} < (\rho)))]c_p \\ + (1-c_p)[P(Q_{i,j} = 0) - P(Q_{i,j} = 1)]P(u_{i,j} < Y_{i,j}) & \quad (14) \\ + P(Q_{i,j} = 1). \end{aligned}$$

Otra probabilidad que se debe tener en cuenta es la que resulta de asumir el muestreo pixeles en el centro del módulo. Entonces la probabilidad de detectar erróneamente un pixel en el centro del módulo esta dada por,

$$\begin{aligned} P_{de} &= P(\text{decidir } Q_c = 1 | Q_c = 0)P(Q_c = 0) + \\ P(\text{decidir } Q_c = 0 | Q_c = 1)P(Q_c = 1) & \quad (15) \end{aligned}$$

En términos del umbral

$$P(\text{decidir } Q_c = 1 | Q_c = 0) = P(\rho_c > u)$$

$$P(\text{decidir } Q_c = 0 | Q_c = 1) = P(\beta_c < u)$$

La probabilidad de error global está entonces dada por

$$P_e = P_{be} p_m + P_{de} (1-p_m) \quad (16)$$

Donde p_m es la probabilidad de ser muestreado.

3.4 VALIDACIÓN DE LA CALIDAD VISUAL.

Para medir la calidad visual de los resultados obtenidos en este trabajo, se empleó el índice de similitud estructural (*Structural Similarity Index-SSIM*) propuesto por [24]. Este índice calcula la calidad de la imagen basado en la degradación de la información estructural en esta, teniendo en cuenta que el sistema visual humano está altamente adaptado para extraer información estructural del campo de visión. El SSIM se calcula entre la imagen original y la imagen resultante.

Finalmente, Los valores óptimos son aquellos que minimizan los niveles de luminancia sujetos a la probabilidad de error,

$$\begin{aligned} (\beta^*, \rho^*, \beta_c^*, \rho_c^*, c_p^*) &= \underset{\beta, \rho, \beta_c, \rho_c, c_p}{\operatorname{argmin}} J(\beta, \rho, \beta_c, \rho_c, c_p) \\ \text{subject to } P_e &< P_{\max} \end{aligned} \quad (17)$$

4. RESULTADOS.

El algoritmo descrito en la sección 3 es aplicado a dos experimentos descritos a continuación. La verificación de la decodificación de las imágenes se realizó utilizando la aplicación móvil desarrollada por la librería Zxing, llamada *Barcode Scanner* versión 4.4.1. Se utilizó para la lectura de los códigos QR los equipos móviles Samsung versión Galaxy modelo GT-S5830M y la versión 5S de *iphone*.

Experimento 1

Para la implementación de los algoritmos descritos en este trabajo se utiliza la herramienta de simulación MATLAB, en un computador MAC OSX, versión 10.8.2, con procesador *Intel core i7*, 32GB de memoria RAM. En esta primera prueba se aplicó el algoritmo desarrollado a tres imágenes a color, de logos e imágenes reconocidas, como son: i) imagen del escudo de la bandera de Colombia. ii) Logo de la empresa Skype. iii) Imagen de globos aerostáticos. En este experimento se compara la calidad de la imagen obtenida por el algoritmo propuesto con el enfoque desarrollado por la empresa *Visualed*, así como también, la efectividad en la decodificación de las imágenes obtenidas. La comparación se realiza solo con este método debido a que no hay otro en el

mercado que se enfoque en el mejoramiento visual de los códigos QR a través de incrustación de imágenes, sin utilizar la información redundante de código. La calidad de la imagen se compara mediante el uso del algoritmo desarrollado en [24] y mencionado en la sección anterior. Las imágenes resultantes se pueden observar en la Tabla 10.

La verificación de la decodificación de las imágenes se realizó tomando 10 muestras para cada imagen a niveles de intensidad luminosa ambiente. Se encontró que las imágenes resultantes son decodificables en todos los casos, pero la velocidad en la decodificación es directamente proporcional a la iluminación del ambiente y a una buena resolución en la impresión.

Tabla 2. Comparación del método propuesto y el método desarrollado por la empresa *Visualed*.

Imagen Original	Imagen Resultante	Método propuesto		Otro método	
		Algoritmo 1		Algoritmo 2	
		Decodificación %	Similitud %	Decodificación %	Similitud %
		100%	99,36%	-	-
		-	-	100%	61,47%
		100%	99,53%	-	-
		-	-	100%	57,47%
		100%	99,51%	-	-
		-	-	100%	62,30%

Tabla 3. Medición del tiempo de decodificación en segundos.

Imagen	Método	Barcode Scanner					Tiempo Promedio	QR scanner					
		Equipo Samsung						Equipo Iphone 5s					
Bandera	propuesto	1,3	1,8	2,4	2,3	2,2	2,0	1,7	1,9	1,2	1,1	1,2	1,42
	Visualed	2,1	4,2	2,4	2,0	3,6	2,86	1,5	1,4	2,6	1,3	2,3	1,82
Skype	Propuesto	2,0	3,6	1,9	2,2	1,5	2,24	3,6	2,5	2,8	2,7	2,9	2,9
	Visualed	1,8	1,5	2,0	1,5	2,2	1,8	2,0	2,2	3,4	3,2	2,1	2,58
Globos	Propuesto	1,7	0,5	1,6	2,0	2,0	1,56	3,7	3,8	2,6	3,2	2,8	3,22
	Visualed	1,8	4,58	1,5	1,4	3,3	2,51	5,1	2,7	3,7	4,5	4,1	4,02

Visualmente la imagen incrustada mantiene los detalles más importantes de la imagen original, por lo que el índice de similaridad es alto. En comparación con la imagen tipo logo "skype" se observa que para el trabajo propuesto las imágenes con mayor número de colores y variedad de detalles alcanzan una mayor definición

Experimento 2

En el segundo experimento se mide cuan rápido se reconoce el código QR (en redes de computadores esta medida se conoce como latencia) y la confiabilidad, puesto que no importa que tan rápido la imagen sea escaneada si genera información incorrecta en la decodificación. Para realizar el experimento se usaron dos aplicaciones distintas para lectura de códigos QR, i) *QR scanner*, aplicación desarrollada por ShopSavvy.Inc. ii) *Barcode Scanner*, aplicación desarrollada por el grupo Zxing, creadores de la librería Zxing usado en este trabajo. El tiempo alcanzado para la decodificación de los datos es bastante rápido, no sobrepasando los 4 segundos para ninguno de los dos algoritmos comparados. Las características de la cámara influyen notablemente en el tiempo de adquisición de la imagen, debido a las características de enfoque superiores. La luminancia es otro factor que afecta la rapidez con que se identifica la imagen, dificultando los procesos de umbralización que debe realizar la aplicación para alcanzar la decodificación. Los resultados son almacenados en la Tabla 11. En promedio, el tiempo de decodificación es bueno para ambas soluciones, sin embargo, como se aprecia en la tabla los tiempos de decodificación alcanzados por la aplicación propuesta son menores.

5. CONCLUSIONES.

El algoritmo propuesto mejora significativamente la apariencia visual de los códigos QR, sin afectar la capacidad de corrección de errores del código

y sin realizar modificaciones en los algoritmos de decodificación.

A partir del algoritmo de incrustación de imágenes en códigos QR propuesto, permitirá que el usuario reconozca de antemano la fuente o propietario de la información codificada al incluir imágenes alusivas al contenido. Por otro lado, al no requerirse modificar el decodificador, el algoritmo propuesto mantiene compatibilidad con diversos lectores de códigos QR disponibles en el mercado.

Se observa que el valor de luminancia óptimo es esencial para mantener un equilibrio entre la mejora visual del código QR y la correcta decodificación de la información en el mismo.

6. REFERENCIAS.

- [1] International Organization for Standardization. (2006). PDF417 bar code symbology specification. Geneva, Switzerland. ISO/IEC 15438:2006(E).
- [2] International Organization for Standardization. (2000). Maxicode. Geneva, Switzerland. ISO/IEC 16023:2000.
- [3] Kaywa. (2005). Datamatrix generator. Zuerich, Switzerland. Recuperado (2013, julio 5) de <http://datamatrix.kaywa.com/>
- [4] International Organization for Standardization. (2006). QR Code 2005 bar code symbology specification. Geneva, Switzerland. ISO/IEC 18004:2006
- [5] Visualed. (2012). QR Code generator. USA. Recuperado (2013, julio 20) de <http://www.visualead.com/>.
- [6] Barcode. (2012). Barcode generator. Baltimore, USA. Recuperado 20 julio,2013) de <http://www.barcoding.com/upc/>

- [7] Soon, T. J. (2008). Qr Code. En D. W. Kian Beng (Ed.). *Synthesis Journal* 2008, p.59 – 78. ISSN 0219-4767.
- [8] Young-Gon, K & Moon-Seog, J. (2011). A design of user authentication system using QR code identifying method. En Ko, F. (Ed.). *6th IEEE International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, Nov. 29 a Dic 01 de 2011, (p.31-35). Seogwipo, South Korea. ISBN 978-1-4577-0472-7.
- [9] Samretwit, D & Wakahara, T. (2011). Measurement of reading characteristics of multiplexed image in QR code. En Xhafa, F, Barolli, L & Köppen, M. (Eds.). *3th IEEE International Conference of Intelligent Networking and Collaborative Systems (INCoS)*, del 30 de noviembre al 02 de diciembre de 2011. (p.552-557). Fukuoka, Japan. ISBN 978-1-4577-1908-0.
- [10] Cox, R. (2012). *Qart codes*. Cambridge, MA, USA. Recuperado (2013, agosto 4). <http://research.swtch.com/qart>
- [11] Otsu, N. (1979). A threshold selection method from gray-level histograms. En Sage, A.P. (Ed.). *IEEE Transactions on System, Man and Cybernetics*, 9 (1), 62-66. ISSN: 0018-9472
- [12] Niblack, W. (1986). *An introduction to digital image processing*. Hemel Hempstead: Prentice Hall.
- [13] Sauvola, J. & Pietikainen M. (2000). Adaptive document image binarization. En Ching, S.Y. (Ed.). *Pattern recognition*, 33(2), 225-236. ISSN 0031-3203
- [14] Huijuan, Y. , Kot, A. , & Jiang, X. (2012). Binarization of Low-Quality Barcode Images Captured by Mobile Phones Using Local Window of Adaptive Location and Size. En Pappas, T. N. (Ed.). *IEEE Transactions on Image processing*, 21(1), 418-425. ISSN 1057-7149
- [15] Owen, S. (2012). *Zxing, multi-format 1d/2d barcode image processing library with clients for android, java and c++, (version 2.0)*. [Programa de computador]. United Kingdom
- [16] Analoui, M. & Allebach, J. (1992). Model-based halftoning using direct binary search. En Rogowitz, B.E. (Ed.) *Imaging: Science and Technology*.). *SPIE Proceedings in Human vision, Visual Processing and Digital Display III*, 1666, 96-108. Doi 10.1117/12.135959
- [17] Lau, D., Arce, G. And Galleagher, N. (1998). Green-noise digital halftoning. En Werner, B. (Ed.). *International conference on Image Processing ICIP'98*, del 4 al 7 de octubre de 1998. (p.2424-2444). Chicago, IL, USA. ISBN 0-8186-8821-1
- [18] Ulichney, R. (1993). Void-and-cluster method for dither array generation. En Allebach, J & Rogowitz, B.E. (Ed.). *SPIE/IS&T symposium on Electronic Imaging: Science and Technology*, 1913, 332-343. Doi 10.1117/12.152707
- [19] Lau, D., Ulichney, R. & Arce, G. (2003). Blue and green noise halftoning models. *Signal processing magazine, IEEE*, 20 (4), 28-38.
- [20] Zou Jiejing, Liu, Junfei & Li, Peng. (2010). Research on binarization of QR code image. En Jiao, F. (Ed.). *International Conference on Multimedia Technology (ICMT)*, del 29 al 31 de octubre de 2010. (p.1-4). Ningbo, China. ISBN 978-1-4244-7871-2
- [21] Lau, D. & Arce, G. (2008). *Modern digital halftoning*. CRC Press.
- [22] Bacca, J., Lau, D. & Arce, G. (2006). A new method for multitone using gray level separation. En Barnes, C. (Ed.). *International conference on Image processing del 8 al 11 de octubre de 2006*, (p. 1505-1508). Atlanta, GA, USA. ISBN 1-4244-0480-0
- [23] Garateguy, G., Lau, D. & Arce, G. (2010). Voronoi tessellated halftone masks. En Law, B. (Ed.). *17th IEEE International conference on Image Processing (ICIP)* del 26 al 29 de septiembre de 2010, (p. 529-532). Hong Kong, China. ISBN 978-1-4244-7992-4
- [24] Wan, Z., Bovick, A., Sheikh, H., Simoncelli, E. (2004). *Image Quality Assessment: From Error Measurement to Structural Similarity*. *Image Processing IEEE Transactions on vol. 13 n°4* pp 600-612.