

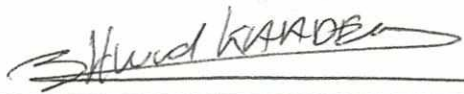
**Técnicas de reconocimiento
de imágenes para la
creación de fotomosaicos**

Universidad del Valle de Guatemala
Facultad de Ciencias y Humanidades

**Técnicas de reconocimiento
de imágenes para la
creación de fotomosaicos**

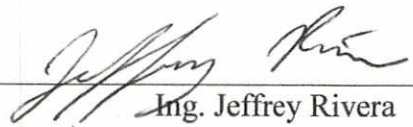
BIBLIOTECA
DE LA
UNIVERSIDAD DEL VALLE DE GUATEMALA

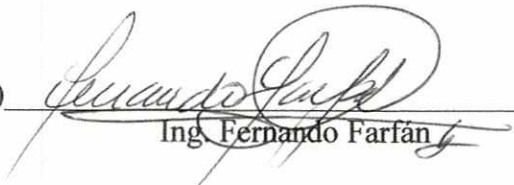
Trabajo de investigación presentado
por Gustavo Adolfo Lara Rodríguez
para optar al grado de Licenciado en Ciencias de la Computación
Guatemala
2003

(f) 
Ing. Jahannes Van der Kaaden

Tribunal:

(f) 
Ing. Luis Masaya

(f) 
Ing. Jeffrey Rivera

(f) 
Ing. Fernando Farfán

Fecha de aprobación: 6 de octubre de 2003

PREFACIO

Mi admiración por los fotomosaicos tiene ya una larga trayectoria. Para la promoción de la película protagonizada por Jim Carrey "The Truman Show" en 1998 se utilizaron carteles en los que se podía distinguir la cara de Jim Carrey construida con imágenes extraídas de la película. Comencé a investigar sobre cómo hacer este tipo de imágenes, pero en ese entonces existía un solo programa cuyo nombre no recuerdo, pero tenía la limitante de sólo poder utilizar 1,000 imágenes y los resultados eran desastrosos.

Pocos meses después tuve la oportunidad de leer una revista en donde se entrevistaba a Robert Silvers, creador de los fotomosaicos. Silvers indicó que su proceso requería de reconocimiento de color, forma y textura y que era necesario el uso de supercomputadoras para poder crear uno solo de estos fotomosaicos, como él los llamaba. Utilizando estas pistas busqué en el Internet sobre técnicas para reconocer estas características en imágenes o fotografías; y me topé con una serie de artículos dedicados al OCR (*Optical Character Recognition*, reconocimiento óptico de caracteres).

Las teorías detrás del OCR podían usarse como guía pero no solucionaban todos los problemas; estaban orientadas a determinar las líneas que conformaban una letra. Entonces decidí intentar con lo que primero se me ocurriera. Los primeros fotomosaicos tardaban aproximadamente 20 horas en su creación pero utilizaban 5,000 imágenes. La utilización de recursos en estos primeros intentos era tan grande que tres discos duros quedaron inservibles. Seguí investigando y haciendo pruebas y en este trabajo expongo los resultados.

CONTENIDO

PREFACIO	I
CONTENIDO	II
RESUMEN	IV
I. INTRODUCCIÓN	1
II. FOTOMOSAICOS.....	2
A. RECONOCIMIENTO DE IMÁGENES	2
1. <i>Reconocimiento de caracteres</i>	3
2. <i>Identificación de personas para investigaciones policíacas</i>	3
3. <i>Biometría</i>	3
B. HISTORIA DE LOS FOTOMOSAICOS	3
C. LOS FOTOMOSAICOS DE SILVERS	5
III. ANTECEDENTES.....	6
A. PERCEPCIÓN DE COLORES.....	6
B. IMÁGENES DIGITALES.....	6
1. <i>Representación de colores en imágenes digitales</i>	6
2. <i>Resolución de las imágenes digitales</i>	7
IV. EL PROCESO DE CREACIÓN DE FOTOMOSAICOS	8
A. NOTACIÓN Y VOCABULARIO.....	8
B. CREACIÓN DE FOTOMOSAICOS	8
1. <i>Entrada</i>	9
2. <i>Salida</i>	9
3. <i>Proceso</i>	9
a. <i>Algoritmo general</i>	9
4. <i>Reconocimiento de imágenes</i>	10
a. <i>Método lineal</i>	11
b. <i>Método cuadrático</i>	11
c. <i>Método Compuphase</i>	12
5. <i>Calidad de los fotomosaicos</i>	12
6. <i>Reduciendo el número de imágenes comparadas</i>	13
a. <i>Color promedio</i>	13
b. <i>Margen de selección</i>	14
7. <i>Alteración de colores</i>	15
8. <i>Reutilización de imágenes</i>	17
9. <i>Prioridad</i>	19
V. COMPRESIÓN JPEG.....	21
A. ACCESO A LA COLECCIÓN DE IMÁGENES	21
B. PROCESO DE COMPRESIÓN.....	21
1. <i>Preparación de la entrada inicial</i>	22
2. <i>Transformada Coseno Discreta (DCT)</i>	22
3. <i>Asignación de pesos</i>	23
4. <i>Dar formato</i>	24
C. JPEG Y LOS FOTOMOSAICOS.....	25
VI. OTRAS TÉCNICAS.....	26
A. SUPERPOSICIÓN DE IMÁGENES	26
B. REUTILIZACIÓN DE IMÁGENES POR RADIO DE REPETICIÓN	27

1.	<i>Tamaño de la colección de imágenes</i>	27
C.	DETERMINACIÓN DE CONTORNOS.....	29
D.	WAVELET	30
VII.	APLICACIÓN DE LAS TÉCNICAS	31
A.	SISTEMAS DISTRIBUIDOS	31
VIII.	CONCLUSIONES Y RECOMENDACIONES	32
APÉNDICE A	33
A.	GLOSARIO DE TÉRMINOS	33
BIBLIOGRAFÍA	35

RESUMEN

En el presente trabajo se analizan y comparan las técnicas utilizadas por los programas comerciales para la creación de fotomosaicos, además se incluyen algunas técnicas nuevas. La mayoría de técnicas son demostradas con ejemplos gráficos comparativos. Para dar soporte a lo expuesto se desarrolló una aplicación en la que es posible encontrar en detalle la ejecución de las técnicas, algoritmos y estructuras de datos descritas en este trabajo.

I. INTRODUCCIÓN

Con este trabajo se documentan las técnicas y algoritmos involucrados en la creación de fotomosaicos asistidos por computadora. Se presentan comparaciones gráficas de cada una de las técnicas y se incluye una aplicación demostrando la ejecución de las mismas.

La comparación de imágenes digitales es la parte esencial en un proceso para la creación de fotomosaicos. Esta comparación presenta una alta demanda de recursos computacionales por lo que se hace mucho énfasis en la optimización del proceso de comparación de imágenes.

La creación de fotomosaicos involucra aspectos de diversas ramas de la ciencia como lo son las funciones matemáticas, teorías del análisis de señales, cálculos estadísticos, optimización de procesos computacionales, y otros más. Los primeros dos capítulos sirven de introducción y forman la base científica para las discusiones en los capítulos siguientes.

II. FOTOMOSAICOS

Un fotomosaico es una colección de imágenes ordenadas de tal manera que presentan una imagen más grande. Esta imagen grande consta de un arreglo bidimensional o matriz de otras imágenes más pequeñas.

La creación de fotomosaicos involucra una alta demanda de los recursos computacionales. En el proceso se pierde parte del detalle de la imagen original como se muestra en la gráfica 1.

Gráfica 1. Ejemplos de fotomosaicos utilizando técnicas expuestas en este trabajo.



1a. Palacio Nacional de Guatemala. Imagen original.



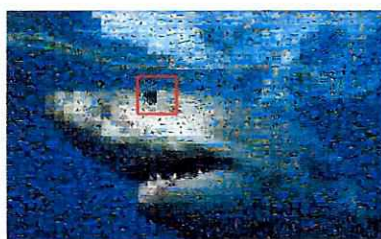
1b. Palacio Nacional de Guatemala. Fotomosaico.



1c. Palacio Nacional de Guatemala. Ampliación del área indicada en rojo en la imagen 1b.



1d. Tiburón. Imagen Original.



1e. Tiburón. Fotomosaico.



1f Tiburón. Ampliación del área indicada en rojo en la imagen 1e.

A. Reconocimiento de Imágenes

Los sistemas de computadoras son cada vez más potentes y menos costosos, lo que permite crear nuevas formas de arte que antes no eran posibles, y algunas otras formas de arte antiguas pueden ahora verse beneficiadas con novedosas técnicas asistidas por computadora.

El reconocimiento de imágenes ha evolucionado a medida que mejora la tecnología. Puede encontrarse en numerosos campos.

1. Reconocimiento de caracteres

El OCR (*Optical Character Recognition*) o reconocimiento óptico de caracteres, es un proceso por el cual en una imagen digital se reconocen los caracteres con la finalidad de poder editarla como texto. Este tipo de aplicaciones son utilizadas como complemento en escáneres y otros dispositivos de captura de imágenes digitales.

2. Identificación de personas para investigaciones policíacas.

Aunque las técnicas aun están en desarrollo en este campo, y aún no existe una aplicación totalmente confiable, es evidente la importancia del reconocimiento de imágenes para la identificación de personas en investigaciones policíacas. Muchas veces, en investigaciones de crímenes, un testigo puede describir con mucho detalle el rostro de un criminal. Un dibujante profesional convierte la descripción verbal del testigo en un dibujo sobre papel. El trabajo de la computadora consiste en buscar el rostro del criminal en una base de datos de imágenes. En las investigaciones policíacas también se utiliza la búsqueda de huellas dactilares en una base de datos.

3. Biometría

La biometría es el reconocimiento del cuerpo humano a través de ciertas características físicas, como el tamaño de los dedos de la mano, las huellas dactilares o los patrones en las retinas de los ojos.

Los sistemas de computadoras actuales permiten tener mejores niveles de seguridad utilizando la biometría. Por ejemplo, una persona puede tener acceso a un área restringida, por medio del reconocimiento de las características físicas de su mano en un dispositivo especial. Si en el proceso de validación se verifica que la persona tiene permiso para entrar al área, entonces le permitirá el acceso. Este tipo de sistemas se está volviendo cada vez más utilizado, desplazando los sistemas antiguos de identificación.

B. Historia de los fotomosaicos.

El artista estadounidense Chuck Close en 1970 dibujó dos imágenes sorprendentes, que parecían haber sido creadas con una sofisticada computadora. Las imágenes estaban hechas por patrones pero éstos habían sido dibujados con lápiz sin necesidad de ninguna computadora, como se muestra en la gráfica 2b.

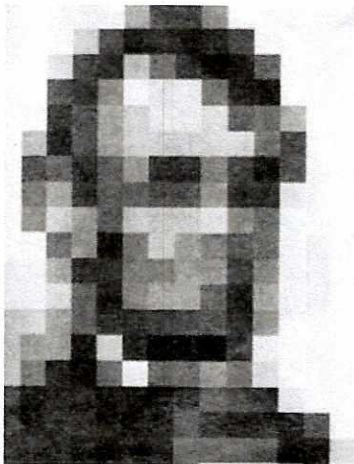
En 1973 un científico llamado Leon Harmon de Laboratorios Bell escribió un artículo en la revista *Scientific American* titulado "The Recognition of Faces" (El reconocimiento de caras). En este artículo se describían las condiciones mínimas para poder reconocer una cara humana. Como ejemplo presentó la cara de

Abraham Lincoln reducida a una matriz de 14 x 18. Cada cuadro de la matriz estaba representado por un único color sólido, como se aprecia en la Gráfica 2a.

La imagen de Abraham Lincoln que hizo Leon Harmon, inspiró al reconocido artista español Salvador Dalí en 1976. Dalí reemplazó los cuadros que mostraba la imagen de Harmon con verdaderas imágenes, como puede verse en la gráfica 2c.

En la década de los noventa, debido a los avances en los programas para la edición de imágenes por computadora, especialmente *Photoshop* de *Adobe*, muchos artistas encontraron una nueva forma de hacer arte. En la gráfica 2d puede apreciarse la obra de Dave McKean, un reconocido artista que creó unas imágenes que marcaban el inicio de una nueva forma de arte, como puede verse en la novela gráfica "Sandman" de *DC Comics*.

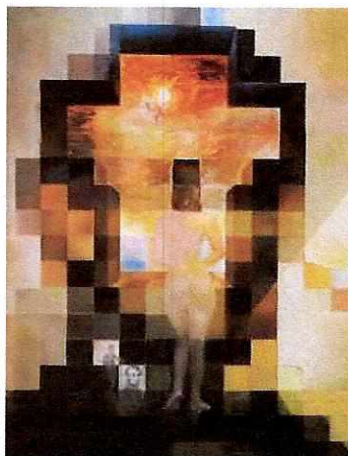
Gráfica 2. Imágenes de los orígenes de los fotomosaicos.



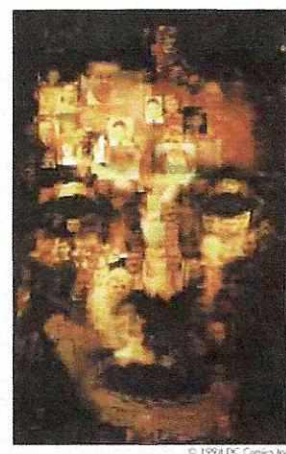
2a. Imagen extraída del artículo de Harmon.



2b. Autorretrato de Chuck Close.



2c. Obra de Salvador Dalí inspirada en el artículo de Harmon.



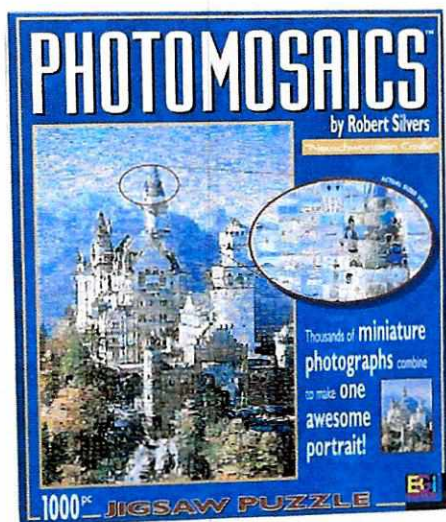
2d. Portada de novela Gráfica por Dave McKean.

C. Los fotomosaicos de Silvers

En 1996, Robert Silvers, aún siendo estudiante de *Massachusetts Institute of Technology* (MIT), desarrolló el primer programa específicamente para crear fotomosaicos. Luego, fundó la empresa *Runaway Technologies* la que ha creado diversos fotomosaicos comerciales como parte de varias campañas publicitarias, como se observa en las gráficas 3a y 3b. Expuso que el proceso involucraba análisis de color, forma y textura.

Después del trabajo de Robert Silvers han aparecido varios programas comerciales que crean fotomosaicos, algunos buenos otros no, algunos rápidos algunos lentos.

Gráfica 3. Ejemplos de fotomosaicos por Robert Silvers.



3a. Los rompecabezas forman parte de la línea de productos de Runaway Technologies.



3b. Publicidad para Jeep.

III. Antecedentes

A. Percepción de colores

El ojo tiene dos tipos de receptores: bastones y conos. Los bastones transmiten diferentes intensidades de gris y los conos le permiten al cerebro percibir la tonalidad de los colores. Existen tres tipos de conos, el primero es sensible a la luz roja, el segundo a la luz verde y el tercero a la luz azul. Cuando un cono es estimulado, el cerebro percibe el color correspondiente. Por ejemplo, cuando los conos verdes son estimulados, se percibe el color verde; cuando los conos rojos son estimulados, se percibe el color rojo. Si son estimulados simultáneamente ambos tipos de conos, los verdes y los rojos, se percibe el color amarillo.

El ojo no puede diferenciar entre un color amarillo espectral, y alguna combinación de rojo y verde. Lo mismo sucede con nuestra percepción de los colores cian, magenta, y los otros colores espectrales intermedios. Debido a esta respuesta fisiológica, el ojo puede percibir una amplia gama de tonalidades por medio de la variación de solo tres colores: rojo, verde y azul.

Cualquier color puede ser espectralmente analizado usando un prisma para determinar sus intensidades primarias de rojo, verde y azul.

B. Imágenes digitales

La digitalización de imágenes es el proceso por medio del cual se convierte una imagen en un formato interpretable por las computadoras. Una imagen digital se puede obtener utilizando cámaras fotográficas digitales, escáneres, etcétera.

En la década de los ochenta sólo era posible utilizar cuatro colores en las pantallas de las computadoras. Esto evolucionó rápidamente y actualmente se utilizan millones de colores.

Una imagen digital es un arreglo de dos dimensiones de píxeles (*picture element*) y un píxel es la unidad mínima que conforma una imagen digital.

1. Representación de colores en imágenes digitales.

Normalmente, los colores primarios son el rojo, el amarillo y el azul, y a partir de la mezcla de estos se crean los otros colores. Sin embargo, en las imágenes digitales los colores primarios son el rojo (*Red*), el verde (*Green*) y el azul (*Blue*) y se utilizan las intensidades de luz para obtener nuevos colores. Esto dio lugar al sistema RGB.

Una imagen digital está compuesta por una matriz bidimensional de elementos RGB. En imágenes digitales de color verdadero, se utilizan 8 *bits* (1 *byte*) para

representar la intensidad de cada componente o canal de color, y por ser tres componentes por color, se necesitan 24 *bits* (3 *bytes*) para formar un solo color, lo que es equivalente a un píxel. Entonces, una imagen de 10 x 10 píxeles utiliza 2400 *bits* o sea 300 *bytes*.

Siguiendo el esquema de 1 *byte* por cada componente o canal, y sabiendo que el valor máximo que puede ser representado por 1 *byte* es 255, la intensidad de una canal está en un rango de 0 a 255, por lo que un píxel estará compuesto por tres diferentes intensidades de R, G y B en un rango de 0 a 255.

El sistema RGB no es el único utilizado en las imágenes digitales. Se han creado diversos sistemas para representar los colores según la aplicación. El sistema RGB es nativo de las computadoras, pero no siempre es el más adecuado. Es un sistema aditivo porque el color blanco se logra mezclando los tres colores rojo, verde y azul, y el negro es la ausencia de color. Por el contrario, en el sistema CMY (*Cyan, Magenta, Yellow*) utilizado para los medios impresos, expresa las intensidades de los colores cian, magenta y amarillo, es un sistema sustractivo porque el color negro se logra mezclando los tres colores y el blanco es la ausencia de color.

En el sistema RGB el blanco es RGB (255, 255, 255) y el negro es RGB (0, 0, 0), en donde los valores entre paréntesis son las intensidades de cada uno de los canales RGB. RGB(255,0,0), RGB(0,255,0) y RGB(0,0,255) representan el rojo, el verde y el azul respectivamente. Al mezclar el color rojo con el azul se obtiene el color morado, RGB(255, 0, 255). Se puede crear todos los colores disponibles variando las intensidades de las componentes RGB.

Otro sistema es el YUV, que fue adoptado por *National Television System Committee* (NTSC) para la transmisión de señales de televisión a color. Su propósito es hacer uso de las características del ojo humano para maximizar la utilización de un ancho de banda definido. El sistema visual humano es más sensible a los cambios de intensidad del color que a los cambios de la tonalidad.

Existe una fórmula para convertir de un sistema de representación de colores a otro. En la conversión no se pierde la información, sino simplemente se representa de una forma distinta.

2. Resolución de las imágenes digitales

La resolución es un factor que se utiliza para convertir las dimensiones de una imagen física a píxeles de una imagen digital y viceversa. Si una fotografía es digitalizada a 300 dpi (*Dots Per Inch*) o puntos por pulgada, significa que por cada pulgada de la fotografía física original se van a obtener 300 píxeles en la imagen digitalizada. Si se imprime una imagen a 75 dpi, quiere decir que por cada 75 píxeles se imprimirá una pulgada sobre papel.

IV. El proceso de creación de fotomosaicos

A. Notación y vocabulario

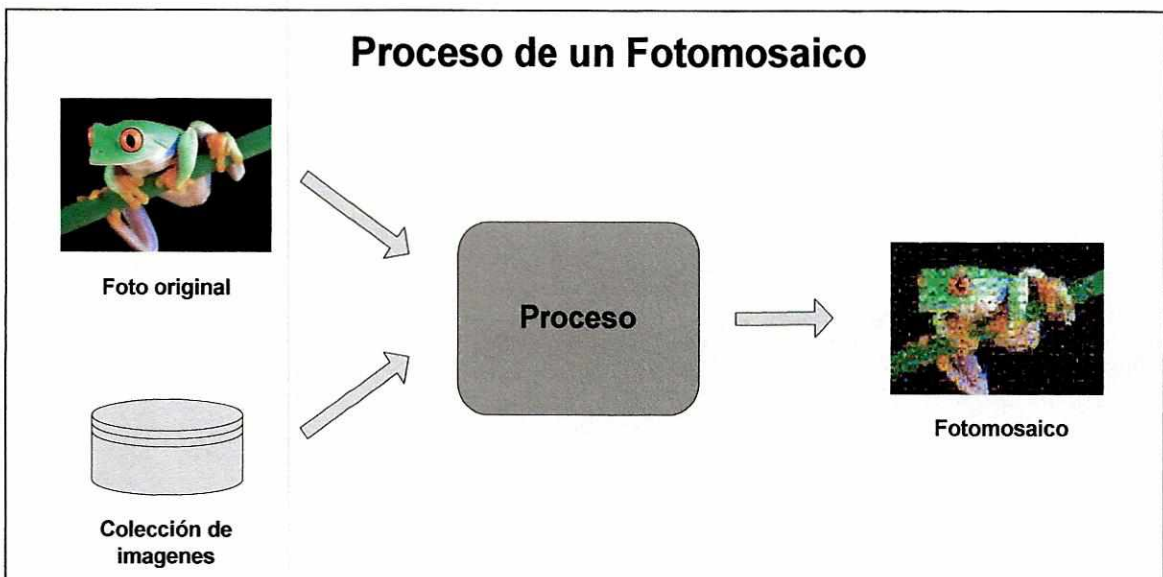
A continuación se encuentra una lista de la notación y el vocabulario utilizados.

Cuadro	Una división de la imagen original. La imagen original es dividida en partes iguales a las que se les llamaran cuadros.
$C(x, y)$	Cuadro localizado en x , y dentro de la matriz de cuadros que conforman la imagen original.
W y H	El ancho y el alto de un cuadro.
X y Y	Columna y fila.
$RGB(r, g, b)$	Un color en formato RGB con sus correspondientes componentes rojo, verde y azul. Ejemplo: $RGB(100, 25, 210)$
Imagen original	La imagen que será transformada en fotomosaico.
Colección de imágenes	El conjunto de imágenes que sirven de entrada al programa para reemplazar la imagen original.
Fotomosaico final o resultado	La imagen original convertida en fotomosaico.

B. Creación de fotomosaicos

La creación de un fotomosaico involucra métodos de alteración de imágenes y el reconocimiento de las mismas. En el Diagrama 1 se presenta el bosquejo del proceso de creación de fotomosaicos.

Diagrama 1.



1. Entrada

El proceso de creación de fotomosaicos necesita como mínimo:

- La imagen original que será convertida en un fotomosaico.
- Una colección de imágenes que se utilizarán para reemplazar la imagen original.

2. Salida

El proceso dará como resultado un fotomosaico en donde la imagen original es reemplazada completamente por una parte o toda la colección imágenes.

3. Proceso

a. Algoritmo general

El proceso para la creación de fotomosaicos parte de la siguiente idea: Reemplazar los segmentos de una imagen con otras imágenes de características similares. El proceso puede apreciarse en el cuadro del algoritmo 1, el cual puede resumirse en los siguientes pasos:

- Dividir la imagen original en cuadros.
- Por cada cuadro:
- Buscar en la colección de imágenes la imagen que más similitud tenga al cuadro original.
- Reemplazar el cuadro con la imagen encontrada.

Algoritmo 1. Algoritmo general para fotomosaicos.

```

crear_fotomosaico (C) : imagen
{
    c = columnas_de_cuadros_en_fuente
    f = filas_de_cuadros_en_fuente

    Para x = 1 .. c                                // por cada columna
    {
        Para y = 1 .. f                            // por cada fila
        {
            mejor_distancia = Infinito
            Para z = 1 .. imágenes_en_coleccion    // por cada imagen en la
                                                    // colección
            {
                distancia = diferencia ( C(x,y), coleccion(z) )
                Si distancia < mejor_distancia entonces
                {
                    mejor_distancia = distancia
                    mejor_imagen = coleccion(z)
                }
            }
            reemplazar ( C(x,y), mejor_imagen )
        }
    }
    resultado = C
}

```

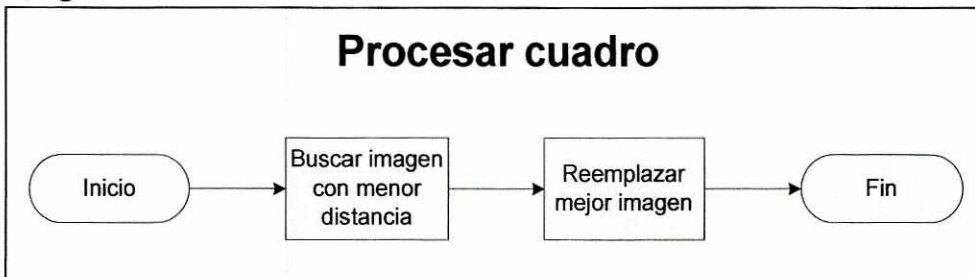
Según el Algoritmo 1, es necesario definir el número de columnas y filas que tendrá el fotomosaico, para determinar el número de cuadros. Si se quiere obtener por ejemplo un fotomosaico de 1280 por 1000 píxeles con cuadros de 128 por 100 píxeles cada uno, se debe dividir la imagen original en 100 cuadros distribuidos en 10 columnas y 10 filas.

En el Algoritmo 1, la función diferencia indica que tan diferentes son dos imágenes. Si el valor de la función diferencia es cero significa que las imágenes son idénticas. Cuanto más grande es el valor retornado por la función, las imágenes son más distintas entre si. La mejor imagen será aquella que dé el menor valor al ser comparada con el cuadro de la imagen original utilizando la función diferencia. Esta es la parte más importante del proceso, por lo que esta función debe ser especialmente eficiente. Es ejecutada por cada imagen de la colección de imágenes por cada cuadro en el fotomosaico.

La función reemplazar del Algoritmo 1, es la función que reemplaza un cuadro de la imagen original definida en el primer parámetro y lo sustituye con la imagen del segundo parámetro. En la aplicación esta función es simplemente copiar y pegar una imagen origen a una imagen destino.

En el Diagrama 2 se aprecia el procedimiento para procesar un cuadro del fotomosaico.

Diagrama 2.



4. Reconocimiento de imágenes

La función **diferencia** es la base del reconocimiento de imágenes; indica la distancia entre dos imágenes. Existen diversas estrategias para reconocimiento de imágenes digitales de acuerdo al tipo de aplicación y de los recursos del sistema.

Utilizando los conceptos derivados del OCR, el primer paso para comparar dos imágenes es vectorizar cada imagen y cada cuadro para luego, comparar las formas de los objetos resultantes. El proceso de vectorización consiste en definir imágenes utilizando la geometría y funciones matemáticas. Los algoritmos existentes para este proceso consumen una gran cantidad de recursos, y la metodología para reconocer la similitud entre estos objetos resulta muy compleja.

La manera más directa de comparar un cuadro la imagen original con una colección de imágenes, es comparar cada píxel del cuadro de la imagen original con su correspondiente píxel en la imagen de la colección imágenes, y acumular las distancia entre cada pareja de píxeles para determinar la distancia general entre las dos imágenes. Aunque ésta es una estrategia relativamente buena para comparar imágenes, la cantidad de comparaciones necesarias es muy grande. Por cada comparación debe calcularse la distancia entre los píxeles de las dos imágenes y por cada pareja de píxeles debe compararse cada uno de los tres canales RGB.

a. Método lineal

La distancia D entre dos píxeles está dada por:

$$D = (\Delta R) + (\Delta G) + (\Delta B)$$

Esta distancia es calculada por cada píxel y por cada canal de color en las imágenes comparadas.

Algoritmo 2. Método lineal.

```
diferencia_metodo_lineal (I1, I2) : entero
{
    diferencia = 0 // diferencia minima
    Para x = 1 .. I1.altura // por cada fila
    {
        Para y = 1 .. I1.ancho // por cada columna
        {
            distancia = Abs(I1(x, y) - I2(x, y)) // se compara cada pixel
            diferencia = diferencia + distancia
        }
    }
    resultado = diferencia
}
```

b. Método cuadrático

Se puede acentuar el efecto de la diferencia de cada píxel utilizando una diferencia cuadrática o distancia euclidiana.

$$D = \sqrt{(\Delta R^2) + (\Delta G^2) + (\Delta B^2)}$$

Como el objetivo es encontrar un valor significativo que nos indique la diferencia entre dos imágenes, podemos eliminar el cálculo de la raíz cuadrada debido al siguiente teorema matemático:

$$\text{Si } a^2 > b^2 \text{ entonces } a > b$$

Algoritmo 3. Método Cuadrático.

```

diferencia_metodo_cuadrático (I1, I2) : entero
{
    diferencia = 0 // diferencia mínima
    Para x = 1 .. I1.altura // por cada fila
    {
        Para y = 1 .. I1.ancho // por cada columna
        {
            distancia = I1(x, y) - I2(x,y) // se compara cada pixel
            diferencia = diferencia + distancia * distancia
        }
    }
    resultado = diferencia
}

```

Este requiere más procesamiento que el método lineal, pero es el más utilizado por la calidad de sus resultados. Puede acentuarse aun más la diferencia utilizando diferentes potencias en la fórmula, pero usualmente no es necesario.

En estos métodos presentados, se asume que las dos imágenes parámetro tienen las mismas dimensiones y que cada píxel está compuesto por tres canales de color RGB.

c. Método Compuphase

El grupo de investigación Compuphase en Holanda, determinó experimentalmente una fórmula que se aproxima mejor a nuestra percepción de colores. (<http://www.compuphase.com/cmetric.htm>)

$$D = \sqrt{(2 * \Delta R^2) + (4 * \Delta G^2) + (3 * \Delta B^2)}$$

Este método requiere mucho más capacidad de procesamiento que los dos anteriores y sus resultados no difieren mucho, por lo que debe utilizarse sólo si se requiere de mayor precisión en la selección de imágenes.

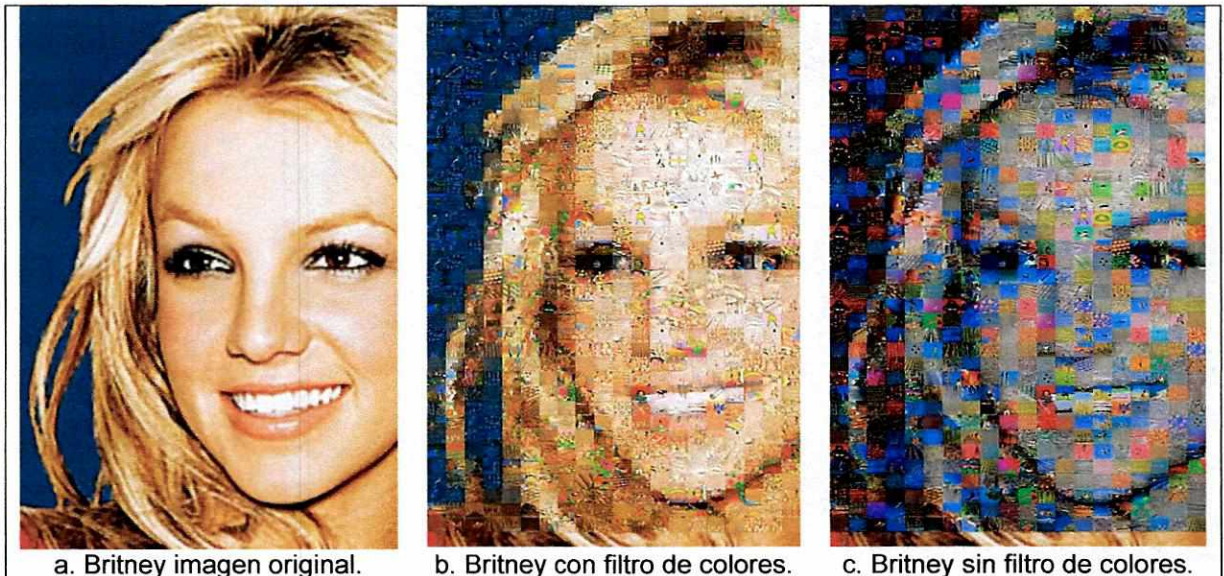
5. Calidad de los fotomosaicos

La mejor herramienta para determinar la calidad de una forma de arte visual es el sentido de la vista, sin embargo, éste es muy subjetivo. Es necesario definir un parámetro para poder comparar fotomosaicos de manera objetiva y así determinar si una técnica está siendo efectiva o no. Un parámetro puede ser el promedio de las diferencias encontradas en el proceso de cada cuadro de la imagen. Esta medida de comparación se puede usar únicamente para comparar los resultados de distintas técnicas sobre una misma imagen original.

En la Gráfica 3 se encuentran fotomosaicos creados con técnicas que aún no han sido introducidas pero sirven de ejemplo para explicar la calidad de los fotomosaicos. El fotomosaico (b) se creó utilizando filtro de colores y alteración de

las imágenes, reportó una diferencia acumulada mayor al mosaico del literal (c) pero evidentemente produce un mejor efecto por tener un color más parecido a la imagen original. El fotomosaico (c) fue creado sin tomar en cuenta la información de color, reportó la diferencia acumulada más pequeña ya que se hizo énfasis en la detección de contornos

Gráfica 3. Comparación de mosaicos.



6. Reduciendo el número de imágenes comparadas

Por cada cuadro se recorre toda la colección de imágenes. A medida que la colección aumenta, el tiempo necesario para encontrar la mejor imagen para ese cuadro también aumenta. Es necesario clasificar las imágenes según sus características de color.

a. Color promedio

Al alejarnos de una imagen, se llega a una distancia en donde la imagen se ve como un punto de un solo color. Se puede reemplazar un cuadro de la imagen con una imagen de un color sólido, como se aprecia en la Gráfica 4, y como menciona Harmon en su artículo. Este color se calcula con el promedio de los colores en la imagen. Para poder usar el color promedio en la colección de imágenes es necesario calcularlo previo a la ejecución del proceso de creación de fotomosaicos.

Algoritmo 4. Color promedio.

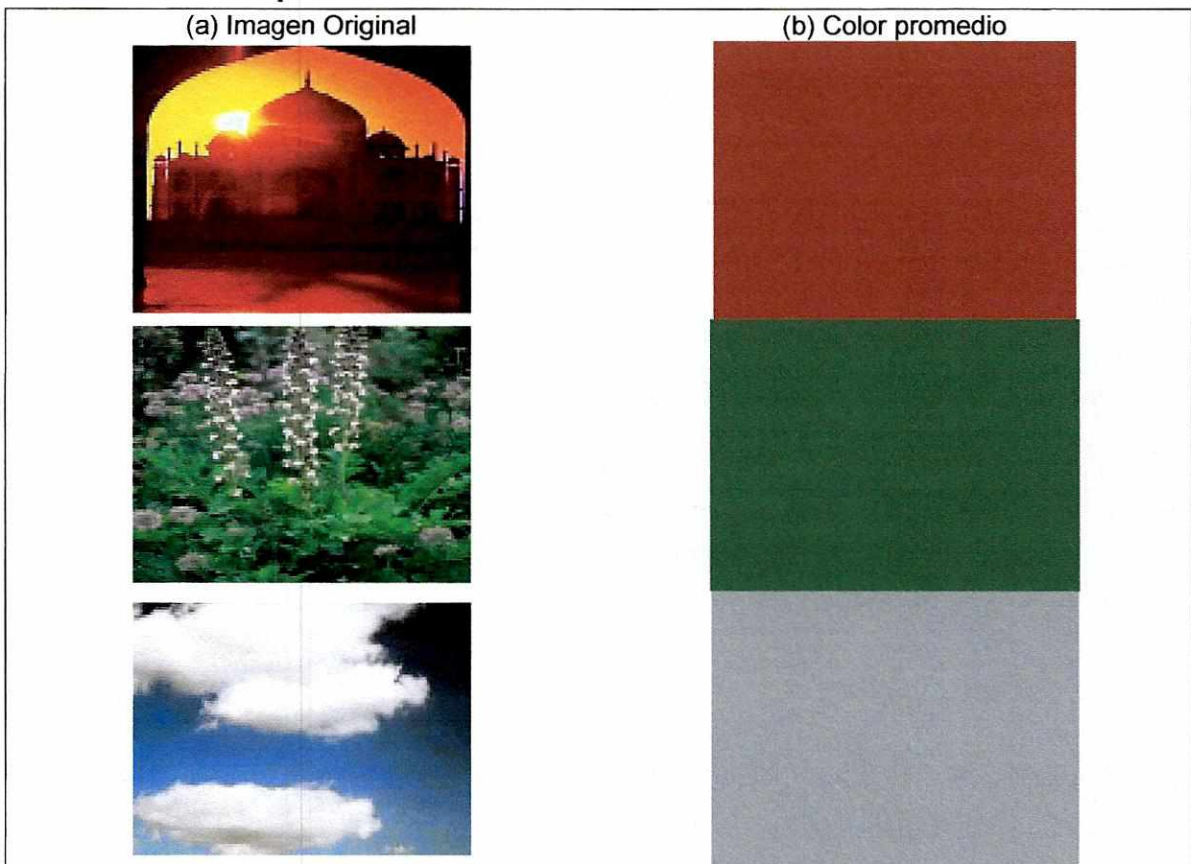
```

color_promedio (imagen) : color
{
    color_total = 0 // diferencia mínima

    Para y = 1 .. imagen.altura // por cada fila
    {
        Para x = 1 .. imagen.ancho // por cada columna
        {
            color_total = color_total + color_pixel(x, y)
        }
    }
    resultado = color_total / imagen.numero_pixeles
}

```

Gráfica 4. Color promedio.

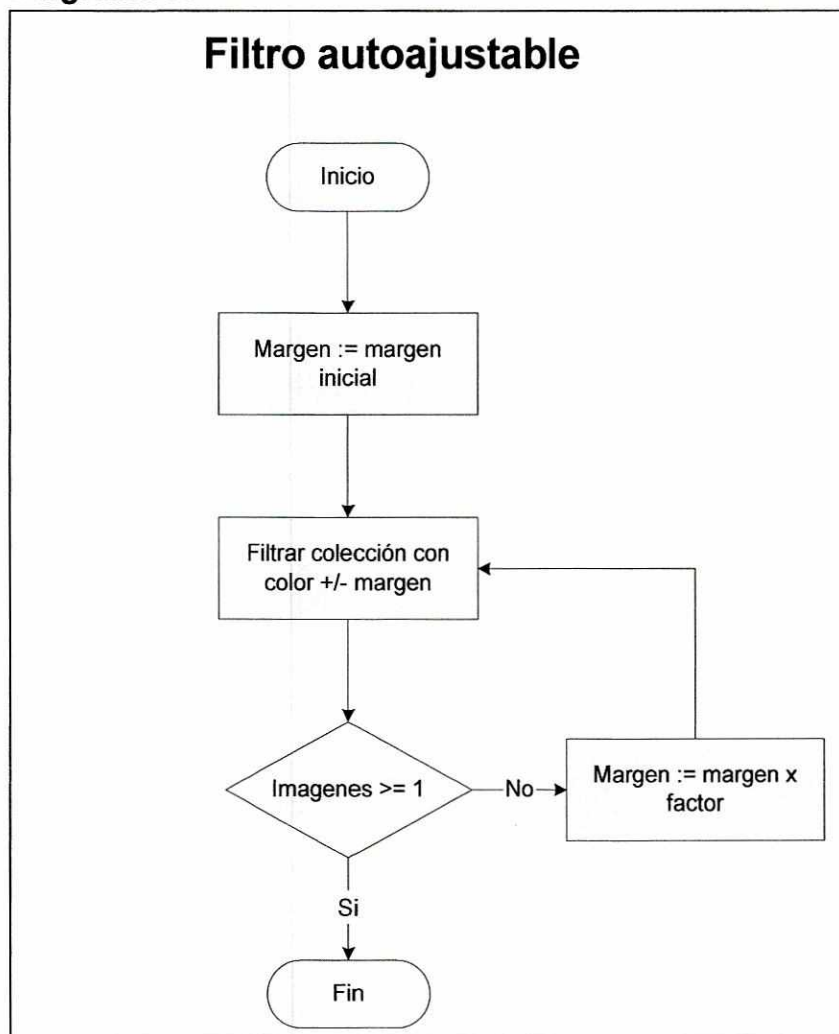


b. Margen de selección

Utilizando un *byte* por cada canal, existen millones de combinaciones de colores posibles. Al buscar las imágenes en la colección de imágenes con un color específico, es posible que no se tenga ninguna imagen con ese color. Entonces se hace necesario definir un rango de colores para filtrar las imágenes. Por otro lado, si la colección de imágenes no es muy grande, puede ser que no existan

imágenes que se encuentren dentro del rango definido del filtro, entonces la definición del rango debe ser dinámica; el margen se debe ajustar automáticamente hasta que exista por lo menos una imagen en el rango definido.

Diagrama 3.



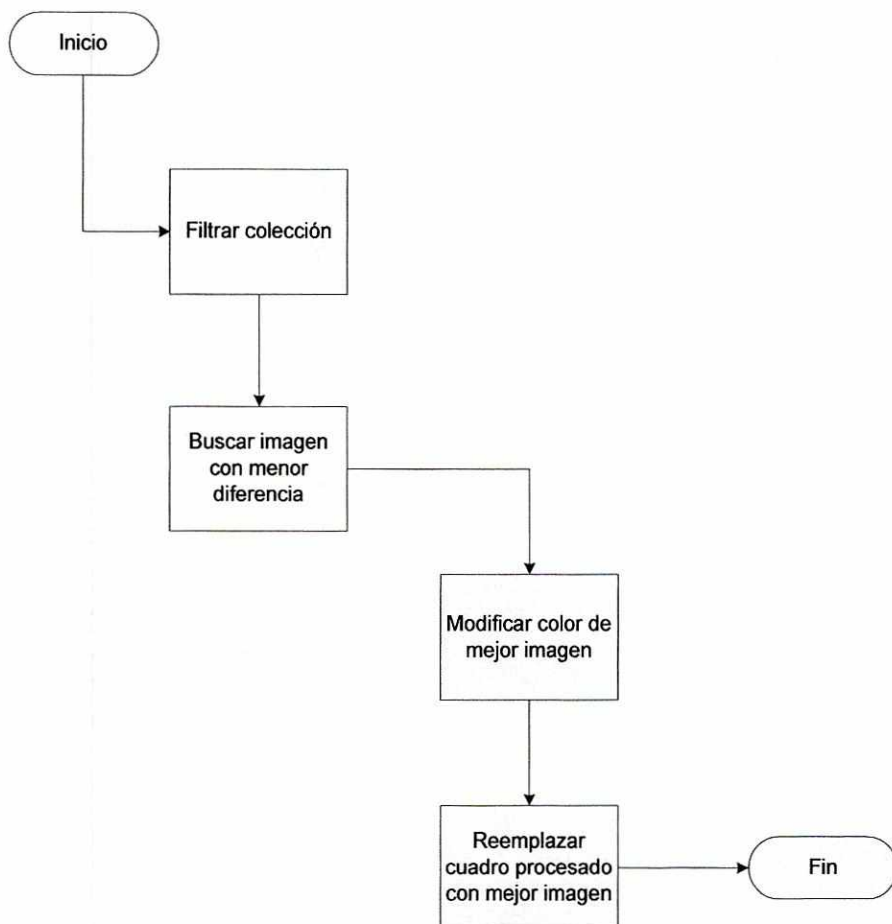
7. Alteración de colores

Si el rango de colores de selección definido en la sección anterior se vuelve muy grande, es posible que se tengan imágenes con color promedio muy distinto al color promedio del cuadro original que está siendo procesado. Esto puede corregirse con la alteración de los colores en la imagen que va a ser sustituida, esta debe ser lo más similar posible al cuadro original procesado.

Algoritmo 5. Modificación de color.

```
modificar_color (imagen, modificacion) : imagen
{
    Para y = 1 .. imagen.altura           // por cada fila
    {
        Para x = 1 .. imagen.ancho       // por cada columna
        {
            imagen(x, y) = imagen(x, y) + modificacion
        }
    }
    resultado = imagen
}
```

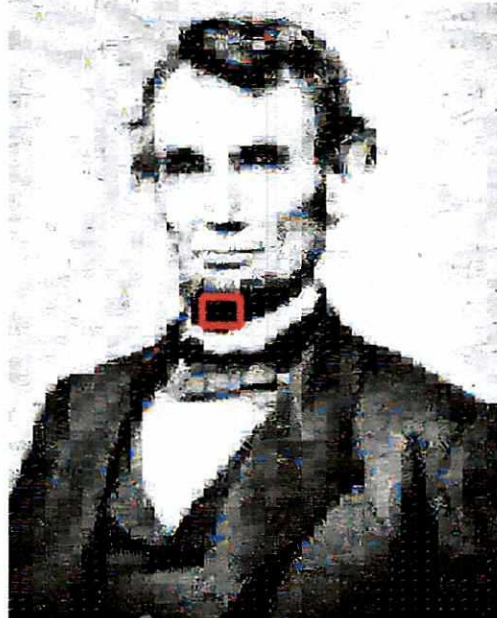
El valor de la modificación puede ser muy grande, y las imágenes de los cuadros individualmente podrán verse distorsionadas causando un efecto indeseable, pero el color general del fotomosaico es mejorado.

Diagrama 4.**Nuevo procesar cuadro**

8. Reutilización de imágenes

Si se tiene una colección de imágenes muy pequeña y la imagen original tiene muy poco detalle, es posible que el proceso repita una imagen ya utilizada. Este efecto, en la mayoría de casos no es deseable por estética; a distancia un fotomosaico puede lucir bien pero al acercarse y percibir el detalle, existen series de imágenes repetidas como puede apreciarse en la Gráfica 5.

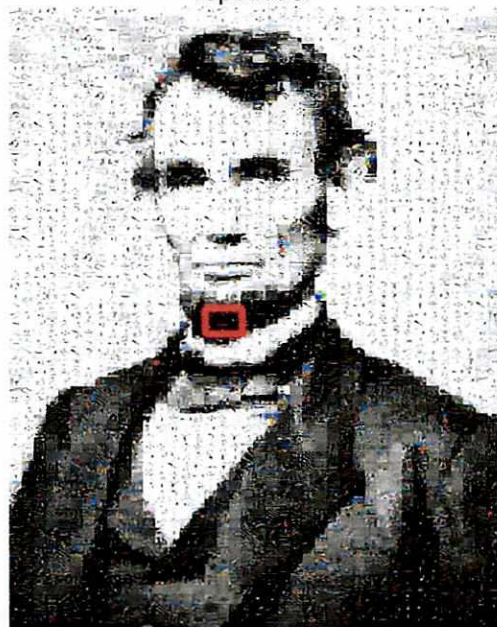
Gráfica 5. Reutilización de imágenes.



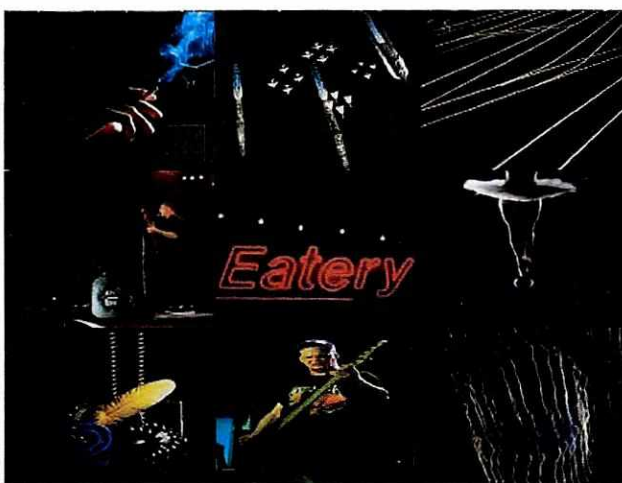
5a. Fotomosaico de Lincoln. Imágenes repetidas.



5b. Ampliación del fotomosaico de Lincoln. Repitiendo imágenes.



5c. Fotomosaico de Lincoln. Sin repetir imágenes.

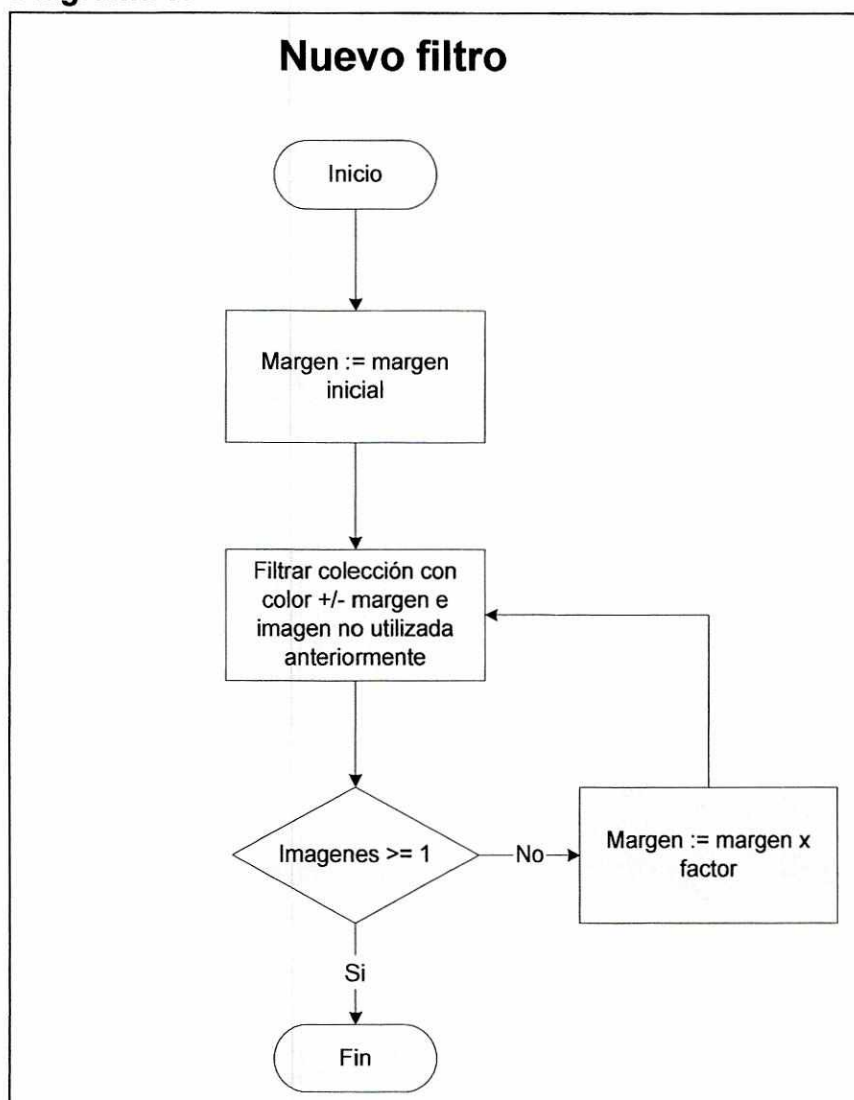


5d. Ampliación de fotomosaico de Lincoln. Sin repetir imágenes.

Este efecto se puede contrarrestar, marcando cada imagen utilizada en la colección de imágenes y agregar la condición al filtro de selección de imágenes de que las imágenes ya utilizadas una vez no pueden utilizarse nuevamente. Esto puede hacerse fácilmente utilizando un campo de tipo *boolean* (verdadero o falso) como bandera en la colección de imágenes.

Esta condición obliga a disponer de una colección de imágenes suficientemente grande como para poder sustituir toda la imagen original. Al utilizar esta técnica, es posible que la calidad del fotomosaico sea disminuida ya que con cada cuadro que se procesa, se va disminuyendo la disponibilidad de imágenes.

Diagrama 5.



9. Prioridad

Si se utiliza la técnica para evitar la repetición las imágenes ya utilizadas, la imagen se va deteriorando a medida que se procesa la imagen original. Al procesar los últimos cuadros se tiene una menor disponibilidad de imágenes en la colección de imágenes que en los primeros ya que una buena parte de la colección de imágenes habrá sido marcada como utilizada.

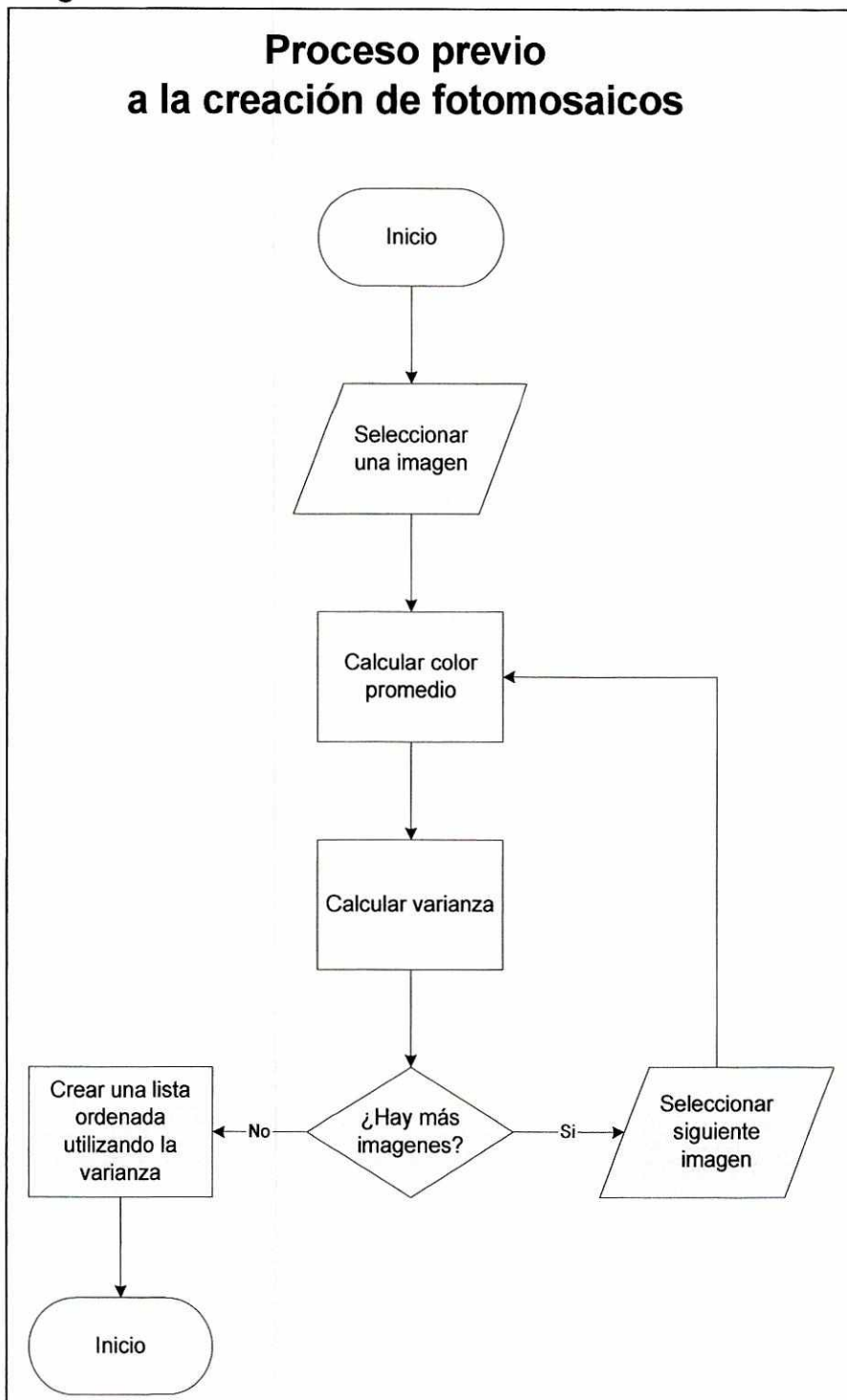
Algunos cuadros necesitan especial atención. Debe dárseles prioridad a los cuadros que tengan más contornos, ya que éstos definen las figuras, o a aquellos que tengan una mayor variación de colores. Con la varianza, derivada de cálculos estadísticos, es posible encontrar la variación de colores en los cuadros. La varianza indica la distancia que hay entre el color promedio de una imagen y cada uno de los píxeles que la componen.

Algoritmo 6. Varianza.

```
calcular_varianza (imagen, color_promedio) : entero
{
    varianza = 0 // varianza minima
    Para y = 1 .. imagen.altura // por cada fila
    {
        Para x = 1 .. imagen.ancho // por cada columna
        {
            varianza = varianza + (imagen(x, y) - color_promedio)
        }
    }
    resultado = varianza
}
```

Antes de iniciar con el proceso de elaboración del fotomosaico es necesario calcular la varianza de cada uno de los cuadros y con esto construir una lista de prioridades. Los cuadros con mayor varianza deben ser procesados primero para que estos tengan una mejor disponibilidad de imágenes. La calidad del fotomosaico es mejorada con esta técnica.

Diagrama 6.



V. Compresión JPEG

A. Acceso a la colección de imágenes

Por cada cuadro se examina una gran cantidad de imágenes de la colección de imágenes. El proceso en general se ve limitado por los accesos a la unidad de almacenamiento secundaria. . «El mejor optimizador está entre los dos oídos» (Abrash, 1997). Se podría pensar que la solución es optimizar las funciones rescribiéndolas en *Assembler* y probablemente se lograría optimizar una parte; el problema no está en las rutinas que ejecuta el procesador, sino en la lectura de la colección de imágenes. Así que es mejor «programar pacientemente para crear código más rápido» (Abrash, 1997).

Normalmente la colección de imágenes se encontraría en un dispositivo de almacenamiento secundario: un disco duro es la opción mas común. La mayoría de estos dispositivos son de naturaleza mecánica, es decir tienen partes móviles; por esta razón sufren desgaste y suelen ser dispositivos periféricos lentos.

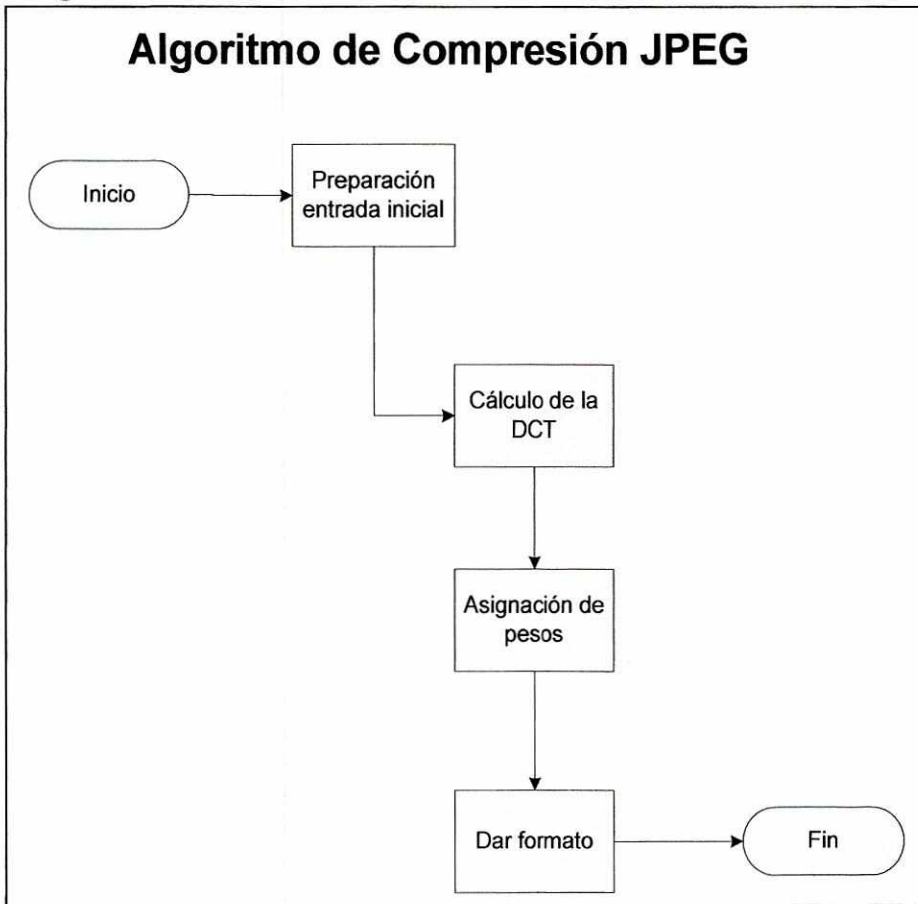
JPEG, es un estándar de compresión controlado por ISO (*International Standards Organization*). Tiene la capacidad de comprimir una imagen con muy poca degradación. Pueden obtenerse proporciones de compresión de hasta 100:1 para imágenes de 24 *bits*. Este algoritmo fue desarrollado por el *Joint Photographic Expert Group* entre finales de los ochenta y principios de los noventa.

B. Proceso de compresión

El proceso de compresión de JPEG está dividido en cuatro subprocesos:

- Preparación de la entrada inicial
- Cálculo de la DCT (Transformada Coseno Discreta)
- Asignación de pesos
- Dar formato

Diagrama 7.



1. Preparación de la entrada inicial

La imagen original es dividida en matrices de 8 por 8. Los colores son convertidos al sistema YUV; utilizando este sistema es más fácil eliminar información innecesaria de la imagen ya que separa la intensidad de los colores de su tonalidad. Opcionalmente puede promediarse la información de píxeles vecinos en los canales U y V para reducir la cantidad de información a ser codificada.

$$\begin{aligned}
 Y &= 0.299 R + 0.587 G + 0.114 B \\
 U &= -0.1687 R - 0.3313 G + 0.5 B + 128 \\
 V &= 0.5 R - 0.4187 G - 0.0813 B + 128
 \end{aligned}$$

2. Transformada Coseno Discreta (DCT)

En cada matriz de 8 por 8 se debe calcular la DCT bidimensional. La DCT permite expresar cada matriz en el dominio de las frecuencias. Este proceso no comprime la información, sólo la expresa de otra forma. Calculando la inversa de la DCT se

obtiene nuevamente la matriz original. El resultado de la DCT es otra matriz 8 por 8.

$$F(u, v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) * \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) * \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right]$$

where: $C(u), C(v) = 1/\sqrt{2}$ for $u, v=0$;

$C(u), C(v) = 1$ otherwise.

3. Asignación de pesos

Como el ojo humano es más sensible a las frecuencias más bajas, con el cálculo de la Transformada Coseno Discreta (DCT) se le asigna un peso mayor a los elementos de la matriz que representan las frecuencias más bajas, que son los que están en la parte superior izquierda. El estándar JPEG presenta unas tablas de pesos sugeridas.

Ejemplo 1

Ejemplo de una matriz de pesos.

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Cada elemento de la matriz resultante es dividido entre su correspondiente elemento en la matriz de pesos. El elemento en la parte superior izquierda de la matriz tiene peso mayor y el elemento en la esquina inferior derecha tiene el menor peso.

Ejemplo 2

Ejemplo de matriz resultante de la DCT antes de la asignación de pesos.

92	3	-9	-7	3	-1	0	2
-39	-58	12	17	-2	2	4	2
-84	62	1	-18	3	4	-5	5
-52	-36	-10	14	-10	4	-2	0
-86	-40	49	-7	17	-6	-2	5
-62	65	-12	-2	3	-8	-2	0
-17	14	-36	17	-11	3	3	-1
-54	32	-9	-9	22	0	1	3

Ejemplo 3

Ejemplo de matriz después de la asignación de pesos

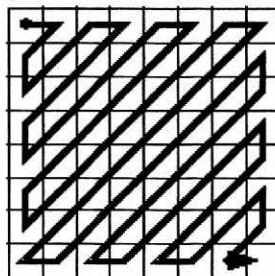
90	0	-7	0	0	0	0	0
-35	-56	9	11	0	0	0	0
-84	54	0	-13	0	0	0	0
-45	-33	0	0	0	0	0	0
-77	-39	45	0	0	0	0	0
-52	60	0	0	0	0	0	0
-15	0	-19	0	0	0	0	0
-51	19	0	0	0	0	0	0

Muchos elementos de la matriz pueden ser redondeados a cero después de la división por la matriz de pesos.

4. Dar formato

La matriz resultante de la asignación de pesos es almacenada utilizando compresión RLE (*Run Length Encoding*) y para aprovechar mejor las series de valores repetidos, la matriz es recorrida en zigzag, como se muestra en la Gráfica 6.

Gráfica 6. Recorrido en zigzag de la matriz resultante.



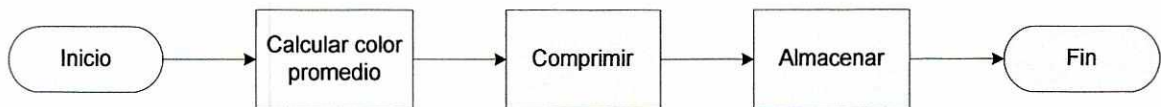
Por último, la información es comprimida utilizando el algoritmo Huffman muy utilizado en otros métodos de compresión como el utilizado en los archivos ZIP.

C. JPEG y los fotomosaicos

Con las técnicas de compresión de imágenes JPEG se puede reducir la cantidad de información almacenada en la colección de imágenes y mejor aun puede reducirse la cantidad de comparaciones necesarias para encontrar la diferencia entre dos imágenes.

Diagrama 8.

Proceso de colección de imagenes (Por cada imagen)



VI. Otras técnicas

A. Superposición de imágenes

Muchas aplicaciones disponibles en la actualidad representan los colores utilizando el sistema RGBA. Este sistema utiliza los tres canales del RGB y agrega el canal *Alpha* representado por la letra "A", el cual mide la transparencia de un color cuando se trabaja con capas superpuestas. Un color con *Alpha* de 0%, es totalmente transparente; y un color con *Alpha* de 100% es totalmente opaco.

Si una imagen "A" se sobrepone a una imagen "B", y si se ajusta el canal *Alpha* de la imagen "A" a 0%, entonces la imagen "A" es totalmente traslucida y la imagen resultante es simplemente la imagen "B".

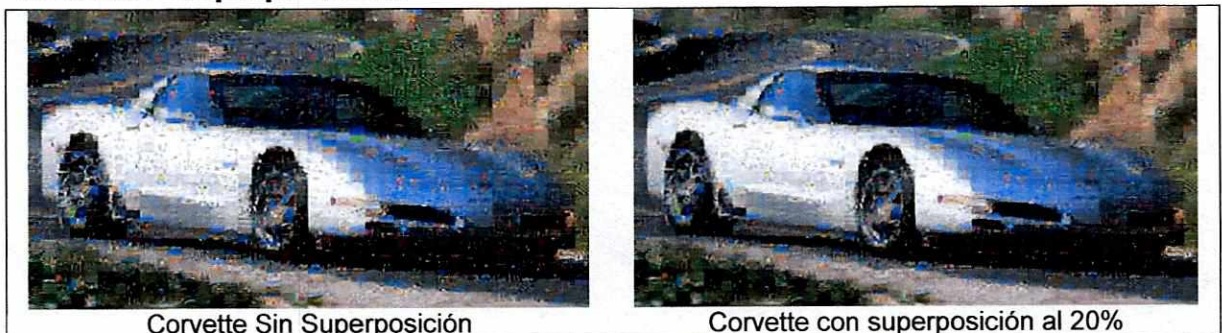
Si se ajusta el canal *Alpha* de la imagen "A" a 100%, entonces la imagen "A" es totalmente opaca y la imagen resultante es simplemente la imagen "A".

Si se ajusta el canal *Alpha* de la imagen "A" a un valor entre 0% y 100%, la imagen "A" será semiopaca.

Con un *Alpha* al 50%, la imagen resultante será la combinación de las dos imágenes cada una con igual presencia.

En la elaboración de los fotomosaicos, la técnica consiste en colocar la imagen original del mosaico, sobre la imagen del mosaico ya terminado a un porcentaje de *Alpha* muy bajo. Luego, hay que comparar el efecto, utilizando diversos porcentajes del canal *Alpha*, ya que puede ser distinto para cada fotomosaico, aunque este valor no suele exceder del 25%; un valor *Alpha* muy alto, degrada el efecto de los fotomosaicos.

Gráfica7. Superposición.



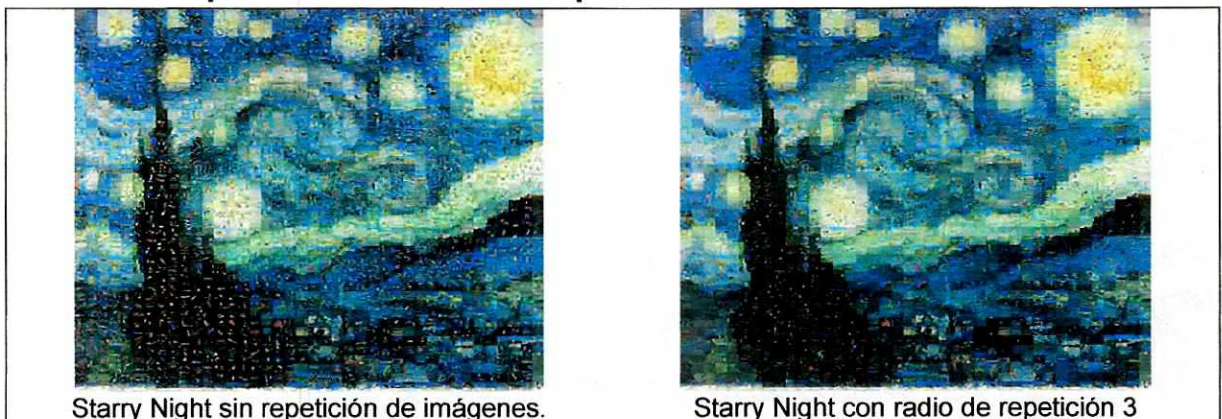
B. Reutilización de imágenes por radio de repetición

Si la cantidad de imágenes en la colección de imágenes es reducida, es posible que estas no sean suficientes para cada cuadro de la imagen original; o la calidad del fotomosaico puede verse deteriorada por la insuficiencia de imágenes.

Si se especifica un radio en el que una imagen no puede repetirse, pueden reutilizarse las imágenes, pero sin tenerlas consecutivas. Por ejemplo: si se utiliza un radio de repetición cinco, una imagen no puede reutilizarse a menos de que haya sido utilizada a una distancia de por lo menos cinco cuadros del cuadro en proceso.

Esta técnica requiere de estructuras de datos complicadas y un proceso previo muy elaborado. Cada cuadro es representado por una estructura con punteros a los cuadros circundantes en por lo menos cuatro direcciones, arriba, abajo, izquierda y derecha.

Gráfica 7. Aplicación del radio de repetición.



1. Tamaño de la colección de imágenes

En la mayoría de casos con una colección de imágenes más grande se producen mejores fotomosaicos. Para un fotomosaico se requieren por lo menos una imagen por cada cuadro si no se repiten las imágenes; y por lo menos una imagen por cada cuadro en el radio de repetición si este se utiliza.

Un fotomosaico de 15 x 10 cuadros requiere por lo menos de 150 imágenes en la colección si no se repiten las imágenes.

Si se utiliza un radio de 3 se requiere de por lo menos 49 imágenes en la colección.

Es muy difícil determinar con precisión cuántas imágenes se requieren para un buen fotomosaico.

En la Gráfica 8 puede apreciarse cómo se comporta la diferencia acumulada utilizando varios tamaños de colección de imágenes. Los valores más bajos en la diferencia acumulada representan mejores fotomosaicos. La curva del mosaico "Letras" dio resultados más altos que los demás, esto se debe a la presencia cuadros con más contornos. Se aprecia también que las curvas de cada mosaico presentan una forma similar: son más pronunciadas con una colección de imágenes más pequeña. Con colecciones mayores a 10,000 imágenes, la curva es cada vez menos pronunciada.

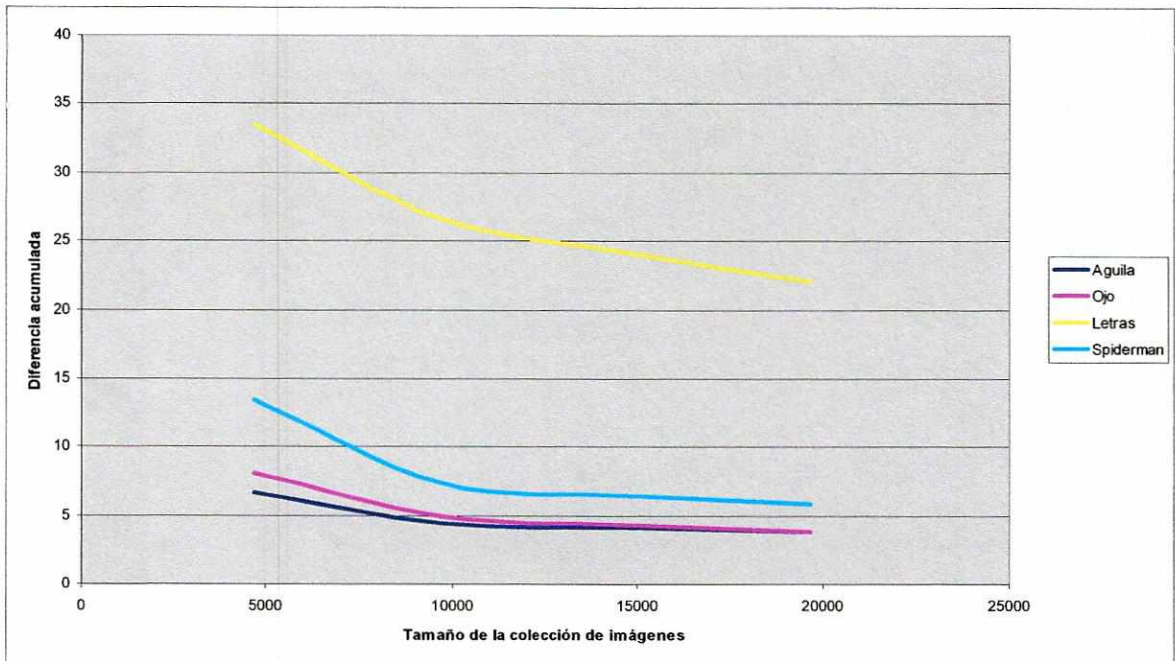
Se puede concluir que:

Una imagen con más contornos requiere una colección de imágenes mayor para obtener resultados aceptables.

El efecto de una colección cada vez más grande es cada vez menos visible.

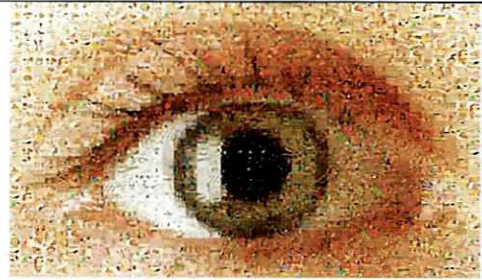
La cantidad de imágenes necesarias para obtener resultados aceptables dependen de la imagen a ser convertida en fotomosaico y la colección de imágenes; una colección de imágenes debe ser preparada para un fotomosaico específico.

Gráfica 8-a. Comparación de diversos tamaños de colecciones de imágenes.





Gráfica 8-b. Fotomosaico Águila



Gráfica 8-c. Fotomosaico Ojo



Gráfica 8-d. Fotomosaico Spiderman

Guatemala

Gráfica 8-e. Fotomosaico de las letras

C. Determinación de contornos

Los cuadros que contienen los contornos de las figuras, son los más importantes en la elaboración de los fotomosaicos. Existen diversos algoritmos para la determinación de los contornos de una imagen. El operador de Sobel es el más adecuado por ser de los que requiere menos procesamiento y se obtienen resultados aceptables.

El operador consiste de dos matrices de convolución, una para los contornos horizontales y otra para los contornos verticales. Uniéndolos los resultados de las matrices es posible encontrar una resultante.

Matriz de Sobel. Contornos eje X

1	0	-1
2	0	-2
1	0	-1

Matriz de Sobel. Contornos eje Y

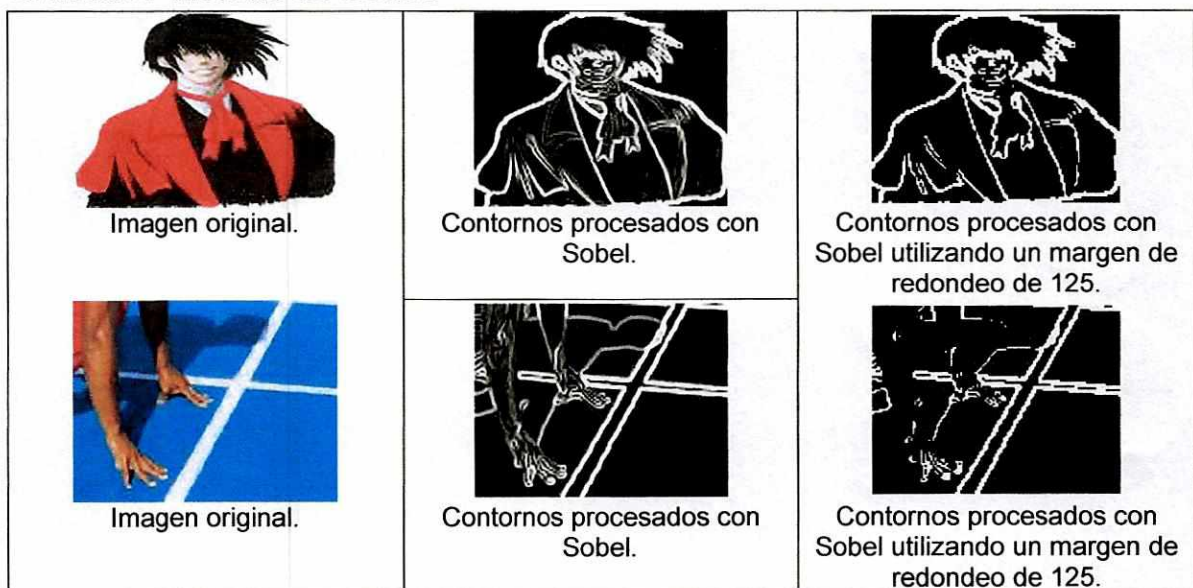
1	2	1
0	0	0
-1	-2	-1

$$\text{resultante} = \sqrt{\text{Sobelx}^2 + \text{Sobely}^2}$$

Con esta técnica se obtienen muy buenos resultados, pero su cálculo demanda una alta capacidad de procesamiento y se requiere de una gran capacidad de almacenamiento para almacenar los contornos de la colección de imágenes. Si los contornos son comprimidos con JPEG se pierde su precisión. Por eso ésta técnica debe utilizarse en sistemas con alta disponibilidad de recursos.

Se puede reducir la cantidad de información almacenada, utilizando redondeo y almacenando la imagen resultante del proceso de Sobel, utilizando un solo *bit* por cada píxel. En la Gráfica 9 se ejemplifican estos conceptos.

Gráfica 9. Método de Sobel.



D. Wavelet

La nueva especificación de JPEG 2000 sugiere la utilización de DWT (*Discrete Wavelet Transform*) como sustituto de la DCT. En la compresión de imágenes, el resultado de la DWT es superior a la de DCT, pero la comparación de los valores más significativos de una imagen produce efectos inesperados.

Para un uso adecuado de la técnica de la DWT, se recomienda hacer un estudio mas profundo de sus características.

VII. Aplicación de las técnicas

Para la demostración de las técnicas, algoritmos y estructuras de datos presentadas en este trabajo, se ha desarrollado una aplicación.

El programa fue desarrollado con Delphi 7.0 para *Windows* de *Borland* por las características que lo favorecían:

Produce ejecutables rápidos comparables a los de los compiladores de C++.

- Código muy legible y fácil de mantener.
- Código portable.
- Objetos que facilitan el manejo de Gráficas.
- Librería JEDI de código abierto que facilita algunos procesos.

Para almacenar y mantener la colección de imágenes se utilizó la base de datos MySQL 4.0 por las siguientes características:

- Rapidez.
- Fácil integración con aplicaciones Windows.
- Código portable.

Para el almacenamiento de la colección de imágenes se calcula la DCT de cada imagen y se almacenan los 64 coeficientes más significativos con precisión de 8 *bits*.

Para futuros proyectos se recomienda la creación de un sistema de almacenamiento más específico. Aunque MySQL satisface las necesidades para la creación de fotomosaicos, los recursos pueden ser mejor utilizados. Estas son algunas recomendaciones para desarrollos posteriores:

- Crear estructuras en la memoria principal para mantener los descriptores de las imágenes.
- Optimizar la búsqueda de rangos sobre tres llaves para mejora en la búsqueda de imágenes utilizando los tres canales R, G y B.

A. Sistemas distribuidos

Debido a la alta demanda de recursos en la creación de fotomosaicos, es recomendado que se consideren los sistemas distribuidos y los sistemas *Multithreaded*. Estos son los puntos más importantes a considerar:

- El cuello de botella es el traspaso de la imagen original o de los cuadros.
- Si se utiliza el radio de repetición debe desarrollarse un mecanismo de bloqueo para evitar que se traslapen los cuadros que se procesan en distintos equipos.
- El proceso se complica si el fotomosaico no permite la reutilización de las imágenes.

VIII. Conclusiones y recomendaciones

Los fotomosaicos son el resultado de la unión del arte con la computación, por lo que la apreciación de su calidad muchas veces es muy subjetiva.

Un proceso puede ser optimizado utilizando algunas técnicas ya establecidas o recetas, pero la verdadera optimización se alcanza cuando se rediseña la secuencia y la lógica detrás de los algoritmos. «El mejor optimizador esta entre tus dos oídos».

Con el tiempo aparecerán nuevas técnicas y nuevos procesos que pueden ser utilizados para hacer mejores mosaicos más rápidamente. El incremento en la capacidad de las computadoras permitirá el desarrollo de nuevos procesos para la creación de fotomosaicos. Por ser también una forma de arte, en el futuro podrá diseñarse nuevas formas creativas de hacer fotomosaicos.

Los programas para crear fotomosaicos consumen muchos recursos computacionales por lo que hay que encontrar el balance entre la calidad del fotomosaico resultante y la velocidad en la que este es producido. Y es necesario diseñar un algoritmo muy eficiente para la determinar la diferencia entre dos imágenes.

Apéndice A

A. Glosario de términos

Algoritmo. Una fórmula o una serie de pasos para resolver un problema. Un algoritmo debe tener una serie de reglas que no sean ambiguas y tener un final bien establecido. Pueden ser expresados en cualquier lenguaje, en español, francés o cualquier lenguaje de programación.

Arreglo bidimensional. Una serie de elementos expresados en dos dimensiones.

Assembler. Lenguaje de programación muy cercano al lenguaje de máquina.

Base de datos. Una colección de información organizada de tal manera que un programa de computadora puede seleccionar rápidamente porciones de datos.

Biometría. Conjunto de técnicas y métodos para la medición electrónica del cuerpo humano.

Boolean. Tipo de dato normalmente representado por un *bit*, que puede tomar valores de falso o verdadero.

Código abierto. Se refiere a un programa de computadora cuyo código está disponible al público en general para su uso o modificación gratuitamente.

Código portable. Un programa de computadora que puede ser utilizado en otra plataforma sin hacer modificaciones en el código.

Convolución. Una función matemática sencilla que es el fundamento en muchos operadores de procesamiento de imágenes. Provee una manera de "multiplicar" dos arreglos de números, generalmente de tamaños distintos, pero la misma dimensionalidad, para producir un tercer arreglo de números de la misma dimensionalidad.

CPU. Central Processing Unit. Componente de una computadora encargado de realizar la mayoría de las operaciones matemáticas y lógicas. Este componente define la velocidad general de una computadora.

Disco duro. Un disco magnético en el que se almacenan datos de computadora. El término duro es usado para distinguirlo de uno suave, o *floppy*.

Estructura de datos. Una organización de datos relacionados. Las estructuras de datos más utilizadas en programación son las listas encadenadas y los árboles.

Fotomosaico. Una imagen construida a partir de otras más pequeñas.

Imagen digital. Una imagen convertida a una serie de datos numéricos.

Internet. Una red global de millones de computadoras intercomunicadas.

Matriz de imágenes. Un arreglo en el que cada elemento es una imagen.

OCR. *Optical Character Recognition*, se encarga de la investigación del reconocimiento de caracteres ya sea impresos o escritos a mano.

RAM. *Random Access Memory*. Memoria utilizada temporalmente para la ejecución de procesos en una computadora.

Recursos computacionales. Cualquier dispositivo que se pueda usar en una computadora. Normalmente se refiere a la capacidad de procesamiento del CPU, a la disponibilidad de memoria RAM y al espacio en los discos duros.

Sistema *Multithreaded*. Un sistema que puede ejecutar diversas porciones de un mismo programa al mismo tiempo.

Supercomputadora. El tipo más rápido de computadoras. Son extremadamente caras y son utilizadas para aplicaciones muy especializadas que requieren una gran cantidad de cálculos matemáticos.

Bibliografía

- Abrash, Michael. 1997. Michael Abrash's Graphics Programming Black Book Special Edition. Estados Unidos. Coriolis Group Books. 1342 pp. ISBN 1-57610-174-6
- Kawick, Mickey. 1999. Real-time Strategy Game Programming using MS DirectX 6.0. Estados Unidos. Worldware Publishing, Inc. 712 pp. ISBN 1-55622-644-6
- Hunt, William. 1999. The Phototiled Images Homepage. Última visita: 27/04/2004. <http://home.earthlink.net/~wlhunt/History/History.html>
- JOINT PHOTOGRAPHIC EXPERTS GROUP. JPEG page. Última visita: 27/04/2004. <http://www.jpeg.org/public/jpeglinks.html>
- DR. ZE-NIAN LI / Simon Fraser University. 2000. Image compression - JPEG. Canadá. Última visita: 27/04/2004. <http://www.cs.sfu.ca/CourseCentral/365/li/material/notes/Chap4/Chap4.2/Ch>
- ROCHESTER INSTITUTE OF TECHNOLOGY / Department of Computer Science. Color Conversion Algorithms. Última visita: 27/04/2004. http://www.cs.rit.edu/~ncs/color/t_convert.html
- RIEMERSMA, Thiadmer. 2001. Colour Metric. Holanda. Última visita: 27/04/2004. <http://www.compuphase.com/cmetric.htm>
- KAPLAN, Ian. 2002. The Daubechies D4 Wavelet Transform. Última visita: 27/04/2004. http://www.bearcave.com/misl/misl_tech/wavelets/daubechies/
- FISCHER, R. Sobel Edge Detectors. Última visita: 27/04/2004. <http://www.dai.ed.ac.uk/HIPR2/sobel.htm>