Revista Venezolana de Computación
ISSN: 2244-7040
http://www.svc.net.ve/revecom
Vol. 1, No. 2, pp. 72-80, Diciembre 2014
Selección de los Mejores Artículos de CoNCISa 2014

Descriptores Visuales de MPEG-7 Empleando la GPU

Víctor Felipe[1], Esmitt Ramírez[2]
victor.felipe@ciens.ucv.ve, esmitt.ramirez@ciens.ucv.ve

[1] Centro de Cálculo Científico y Tecnológico, Universidad Central de Venezuela, Caracas, Venezuela
[2] Centro de Computación Gráfica, Universidad Central de Venezuela, Caracas, Venezuela

Resumen: El ser humano puede de forma intuitiva seleccionar un grupo de fotografías de un conjunto e identificarlas como similares. Para ello, se emplean criterios basados en procesos cognitivos de aprendizaje centrados en sus características visuales. En un computador, se trata de emular dicho proceso al calcular un vector de características para una imagen, el cual permite identificarlas por sus atributos como su luminosidad, color predominante, intensidad, tonalidad, entre otros. El estándar MPEG-7 es una representación de imágenes y video que define, entre diversos atributos, características llamadas descriptores visuales los cuales pueden ser empleados para la aplicación de funciones de similitud entre imágenes. Sin embargo, el costo computacional de obtener estos descriptores es elevado. En este trabajo proponemos realizar estos cálculos empleando arquitecturas paralelas ofrecidas por las tarjetas gráficas. De esta forma, se realizan modificaciones a la propuesta original de MPEG-7 para ser ajustadas a las GPUs y obtener resultados en menores tiempos manteniendo la eficacia. La experimentación realizada justifica nuestra propuesta al implementar descriptores visuales de color, textura y forma, aplicados a una gran cantidad de imágenes permitiendo determinar con precisión la similitud entre pares.

Palabras Clave: Descriptores Visuales; Procesamiento Digital de Imágenes; MPEG-7; GPUs; Fotomosaicos.

Abstract: The human being is able to intuitively select a group from a set of images, and identify them as similar. Thus, some criteria are used based on cognitive learning processes focused on their visual features. On a digital device, this process is emulated generating a vector of image features, allowing the identification of their attributes such as luminosity, predominant colors, intensity, tonality and others. MPEG-7 defines visual descriptors which can be used to apply similarity functions between images. However, the computational cost of this measurement is usually high. In this work, we propose an approach to accelerate these calculations using the parallel architectures offered by GPUs. Thus, modifications to the original MPEG-7 proposal were performed to adjust our implementation to these architectures in order to reduce the computational cost of these computations. The experimentation accomplished justified our work in the use of high amount of images to determine the precision of these visual descriptors to find similar images.

Keywords: Visual Descriptors; Digital Image Processing; MPEG-7; GPUs; Photomosaics.

I. Introducción

El MPEG (*Moving Picture Experts Group*) establecido en 1988 ha desarrollado una serie de estándares para el manejo, tratamiento, compresión y visualización de contenido digital audiovisual. Así nace el estándar MPEG-7 que se centra en proveer descripciones de imágenes, audio y video, lo cual contribuye al filtrado y la categorización de contenido. Para ello, MPEG-7 busca una forma simple de conectar los elementos del contenido audiovisual, así como encontrar y seleccionar de forma adecuada la información que un usuario requiere. El estándar permite el manejo de audio, modelos 3D, video e imágenes, siendo esta última el punto de interés para nuestro estudio.

Es conocido que el contenido presente en el mundo actual a través de los diversos medios es altamente audiovisual, con el objeto de llegar de forma directa a un usuario final. Diversos contenidos para publicidad, entretenimiento, educación, entre muchos otros, son primordiales y están presentes con mayor auge cada día. Así, la necesidad de mantener contenido inequívoco toma importancia. Un caso particular lo constituyen las imágenes, las cuales pueden estar duplicadas en diversos ámbitos dentro de un gran repositorio de datos (e.g. Internet), lo cual puede resultar en redundancia, necesidad de almacenamiento, mal uso, etc. Del mismo modo, resulta interesante encontrar muchas versiones ligeramente diferentes de una misma imagen. Por ejemplo, una aplicación para dispositivos móviles de reconocimiento de lugares turísticos que tome como entrada una fotografía de un lugar a ubicar (e.g. la Torre Eiffel en París, Francia).

Sin embargo, desde el punto de vista computacional el proceso de encontrar imágenes similares dentro de un gran banco de datos no es tarea trivial. Una buena técnica es llamada *Query by Example* que es empleada principalmente por sistemas de

consulta de imágenes mediante ejemplo (*Content-based Image Retrieval* - CBIR) [1], que permite buscar imágenes basadas en su contenido dentro del contexto de color, textura y forma.

Este trabajo se basa en la construcción de descriptores visuales del estándar MPEG-7 basados en color, textura y forma para conseguir un conjunto de imágenes similares dentro de un repositorio, teniendo como entrada una imagen base. Dado el alto cómputo requerido para obtener los descriptores del estándar, se realiza un diseño e implementación bajo una arquitectura paralela que ofrece un hardware de bajo costo como lo son las tarjetas gráficas. Para ello, se emplea la arquitectura CUDA (*Compute Unified Device Architecture*) como base de trabajo ofreciendo muy buenos resultados al aplicar las diversas funciones de similitud de los descriptores visuales. Así, nuestra propuesta presenta una implementación eficaz y eficiente del estándar aplicando modificaciones que permiten mejorar y adaptar los descriptores a un ambiente bajo la GPU (*Graphics Processing Unit*). En este sentido, se utilizan descriptores visuales para la generación de fotomosaicos como caso de estudio de una aplicación de tipo CBIR.

Este artículo se organiza como sigue: en la Sección II, muestra un resumen de los trabajos previos relacionados con nuestra investigación. La definición de los descriptores visuales del estándar MPEG-7 se presenta en la Sección III. La Sección IV presenta el enfoque utilizado para la implementación de cada uno de los descriptores, y en la Sección V se muestra la experimentación realizada y los resultados obtenidos de dicho enfoque. Finalmente, en la Sección VI se presentan las conclusiones de nuestra investigación y posibles trabajos futuros.

II. Trabajos Previos

Las aplicaciones basadas en CBIR, corresponden a una rama de estudios en diversos centros de investigación del mundo, existiendo actualmente gran cantidad de información sobre las tecnologías que las implementan, métricas, buenas prácticas, entre otras [2][3]. En especialidades como la medicina, se ha empleado para el diagnóstico de patologías conocidas basadas en imágenes radiográficas, muestras citológicas, MRI (*Magnetic Resonance Imaging*), entre otras.

De manera habitual, con el objetivo de mantener una consistencia adecuada entre diversas aplicaciones, éstas han optado por emplear el estándar MPEG-7 para poder persistir y ser de utilidad en diversos ámbitos [4].

Diversos trabajos asociados a los descriptores visuales del estándar MPEG-7 han sido desarrollados recientemente. En [5] se presenta el algoritmo *k*-medias que es utilizado para la obtención de características de color a partir de imágenes. Por otro lado, Sergyán [6] propone una medida de similitud que permite comparar imágenes en base a estas características. Recientemente, Felipe y Ramírez [7] presentaron una implementación eficiente del algoritmo *k*-medias empleando la GPU.

En cuanto al uso de descriptores de forma, Park et al. [8] proponen el cálculo de características de forma siguiendo un esquema en bloque a partir de sub-imágenes, considerando la información de los bordes presentes, tanto de forma local como global.

Por su parte Hosny presenta en su trabajo [9] una aproximación para el cálculo de los coeficientes de la transformada radial angular (ART), utilizada para determinar características de forma asociadas a imágenes, llevando a cabo un proceso de interpolación como lo muestra Xin [10] en su trabajo.

En este trabajo, planteamos un caso de estudio basado en el uso de fotomosaicos para demostrar la efectividad de los descriptores de MPEG-7 en la GPU. De este tópico, existen diversas investigaciones que presentan técnicas novedosas para su generación, variando su aspecto visual [11][12][13].

Enfocando la utilidad de los descriptores visuales a una aplicación, los buscadores convencionales permiten la búsqueda de elementos (e.g. imágenes) haciendo uso de sus metadatos tales como su nombre, lo cual dificulta en muchos casos encontrar los elementos deseados. La utilización de información que puede ser obtenida a partir de características tales como color, textura y forma, facilita la búsqueda de imágenes similares. Recientemente, la utilización de sistemas de esta naturaleza se ha incrementado debido a la alta disponibilidad y poder de procesamiento provisto por los dispositivos móviles (e.g. aplicaciones como *Google Goggles* [14] y *CamFind* [15]).

En la literatura presentada, las investigaciones existentes no hacen énfasis en el costo computacional que el cálculo de descriptores visuales representa, lo cual ha motivado la realización de este trabajo bajo una arquitectura paralela de grano fino como es el caso de las GPUs.

III. Descriptores Visuales de MPEG-7

Los descriptores visuales definidos en el estándar MPEG-7 describen contenido (i.e. imágenes y video) en base a características tales como color, textura, forma y movimiento:

- Color: Es una característica visual robusta a los cambios en el ángulo de visión, así como a traslaciones y rotaciones de las ROI (*Region of Interest*). El estándar MPEG-7 presenta diversos descriptores que representan distintos aspectos asociados al color.
- Textura: Contiene información estructural importante acerca de las superficies y su relación con el entorno haciendo referencia a los patrones visuales que tienen o sus propiedades homogéneas. Es una característica natural de cualquier superficie (e.g. césped, paredes de ladrillos, etc.).
- Forma: Provee información relevante para la búsqueda y comparación de imágenes, presentando formas elementales basadas en regiones y contornos, es decir, patrones estructurales de las superficies y su entorno.
- Movimiento: Los descriptores de color, textura y forma pueden ser empleados para la indexación tanto de imágenes como videos. Los descriptores de video proveen pistas poderosas relacionadas a la ubicación espacial de los objetos presentes en diversos cuadros.

Existen diversos descriptores presentes en el estándar MPEG-7 y para los fines de este trabajo se consideran tres de ellos, que permiten obtener información asociada a las características de color, textura y forma. A continuación, se muestra una breve descripción de cada uno.

A. Descriptor de Colores Dominantes

El descriptor de colores dominantes (*Dominant Color Descriptor* - DCD) provee una descripción compacta de los colores representativos en una imagen. Se define como un conjunto de k duplas como en (1), donde k representa el número de colores dominantes:

$$DCD = \{(c_i, w_i)\} \text{ donde } i \{0,1,...,k-2, k-1\} \quad (1)$$

El valor de c_i es representado por un vector de componentes de un espacio de color (e.g. en el caso del espacio RGB consta de 3 valores), mientras que w_i representa la proporción del número de píxeles de la imagen, asociados al color dominante c_i en el rango [0,1].

La tarea consiste en encontrar los k colores dominantes de una imagen. El algoritmo clásico para extraer dichos colores es el algoritmo k-medias.

B. Descriptor de Histograma de Bordes

El descriptor de histograma de bordes (*Edge Histogram Descriptor* - EHD) toma ventaja de la distribución espacial de los bordes presentes en una imagen. Para localizar la distribución de los bordes en regiones específicas de una imagen se divide el espacio en 4x4 sub-imágenes y éstas a su vez en bloques (ver Figura 1). De esta forma, se genera un histograma que representa esta distribución en una imagen.

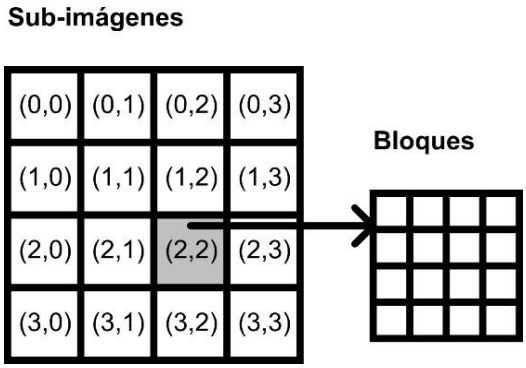


Figura 1: División del Espacio de una Imagen en Sub-Imágenes y Bloques

Cada bloque de una sub-imagen se puede clasificar como borde horizontal, vertical, diagonal 45º, diagonal 135º o no direccional. Luego de realizar el proceso de extracción del tipo de borde asociado a cada bloque, se determina la cantidad de bloques asociados a cada uno de los cinco tipos definidos. Dado que se tienen 16 sub-imágenes, se generan *5x16=80* contenedores para representar el histograma que recibe el nombre de histograma local de bordes (LEH).

Los valores de los contenedores son normalizados en base al número de bloques clasificados de un mismo tipo, obteniendo valores en el intervalo [0,1]. Por otro lado, se generan dos histogramas adicionales denominados histograma global de bordes (GEH) e histograma semi-global de bordes (SGEH), haciendo uso de la información del LEH, con el fin de incrementar la precisión del EHD.

C. Descriptor de Forma Basado en Regiones

El descriptor de forma basado en regiones (*Region-based Shape Descriptor* - RBSD) representa la distribución de píxeles dentro de un objeto o región en R^2. Este descriptor emplea la transformada radial angular (*Angular Radial Transform* - ART) [16] con valores complejos sobre un disco unitario en coordenadas polares.

Los coeficientes de la ART de orden p y q de una función de intensidades $I(x,y)$ están definidos por la proyección de una imagen de entrada en las funciones de base de la ART como se muestra en (2).

$$F_{pq} = \int_0^{2\pi}\int_0^1 V^*_{pq}(r,\theta) f(r,\theta) r dr d\theta \quad (2)$$

La función de base de la ART V^*_{pq} de orden p y q representa polinomios ortogonales continuos definidos en coordenadas polares sobre un disco unitario, mientras que el símbolo * representa su conjugada compleja, siendo la función $f(r,\theta)$ la correspondencia de la función de intensidades en coordenadas polares.

Con el objetivo de resolver las integrales de (2), se lleva a cabo una conversión (ver (3)), a una ecuación discreta aproximada para los coeficientes de la ART de orden p y q.

$$F_{pq} = \frac{1}{2\pi} \sum_{\forall i} \sum_{\forall j} f(r_i, \theta_{ij}) I_p(r_i) I_q(\theta_{ij}) \quad (3)$$

siendo las componentes I_p e I_q las representaciones de las partes radial y angular respectivamente de la función base de la ART. La función $f(r_i, \theta_{ij})$ se puede deducir a partir de la función original (i.e. imagen de entrada) empleando una interpolación basada en un spline cúbico.

IV. Enfoque Propuesto

Como se mencionó anteriormente, la generación de vectores característicos que representen de forma óptima una imagen no es una tarea trivial, dado que es posible obtener una gran cantidad de información para su representación. Los descriptores visuales de MPEG-7 permiten obtener información relevante a partir de imágenes en base a sus características tales como color, textura y forma.

Nuestra propuesta permite obtener estos descriptores visuales empleando la GPU, seleccionando el descriptor de colores dominantes, de histograma de bordes y de forma basado en regiones. Así, para dos imágenes de entrada I e I', se busca obtener los valores de sus descriptores y determinar su grado de similitud.

La obtención de los descriptores visuales de MPEG-7 involucra una gran cantidad de operaciones computacionales y por ello el empleo de la GPU. Desde este punto al referirse a hilos, se hace referencia a la secuencia de instrucciones más pequeña que es manejada independientemente por la tarjeta gráfica. A continuación, se describe el procedimiento para el cálculo de cada uno de los descriptores.

A. Colores Dominantes

El cálculo de los colores dominantes se realiza basado en el algoritmo k-medias. La idea básica es agrupar en k grupos una población de n individuos, basado en un criterio de similitud.

En un proceso de reducción de *C* colores iniciales de una imagen *I* (cuantización), se identifican los *k* grupos que representan los colores dominantes y los *n* individuos como los píxeles de *I*. En nuestra propuesta el cálculo computacional se realiza en la GPU y la CPU.

En la GPU, dada una imagen *I* y un conjunto de colores *C*, se determina para cada píxel el color que mejor lo represente empleando la medida de distancia euclidiana. Debido a la independencia entre los píxeles de *I*, se emplea un enfoque paralelo donde se agrupan hilos en conjuntos que denominamos bloques *B*.

En la CPU, se actualizan los valores centrales y se realiza el proceso de cuantización de la imagen en base a la información obtenida. Un bloque B_i procesa la fila *i* de *I*, mientras que un hilo *j* del mismo bloque se encarga de procesar los píxeles en las posiciones *(i, j + k x n)*, donde $k \geq 0$ y *n* representa el número de hilos de cada bloque. La Figura 2 ilustra esta distribución de bloques e hilos con *n = 4*.

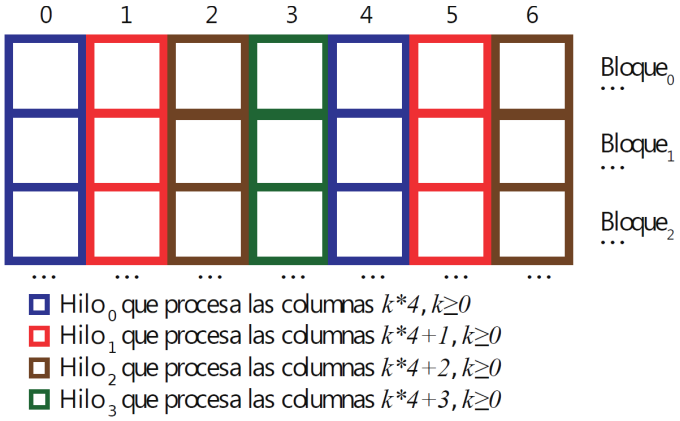


Figura 2: Esquematización de la Distribución de Píxeles en Hilos de Distintos Bloques con n = 4

El descriptor de colores dominantes basado en el algoritmo *k*-medias permite involucrar ponderaciones en el DCD que representan el número de ocurrencias de colores dominantes en la imagen *I* (i.e. imagen cuantizada).

Para calcular la distancia entre las imágenes *I* e *I'* se emplea la distancia entre las duplas generadas por sus respectivos descriptores I^A e I^B, como se muestra en (4).

$$D(I,I') = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left\|c_i^I - c_j^{I'}\right\| * \left|p_i^I - p_j^{I'}\right| \qquad (4)$$

Los valores de *M* y *N* representan el número de colores dominantes obtenidos de las imágenes *I* e *I'* respectivamente.

B. Histograma de Bordes

El descriptor de histograma de bordes (EHD) obtiene los tipos de borde por cada bloque creado en la imagen *I* y los clasifica de acuerdo a su tipo de borde de acuerdo a 5 posibles. Independientemente de la dimensión de *I*, se divide en un número predeterminado de bloques, existiendo una dependencia entre sus dimensiones y la resolución de la imagen.

Cada bloque se subdivide en cuatro sub-bloques y se calcula el valor medio de la luminancia de cada uno de ellos. Esta luminancia se emplea para obtener los bordes a través de una operación de convolución para calcular la magnitud de éstos (i.e. de los 5 bordes a clasificar). Si la mayor de las magnitudes obtenidas es superior a un umbral, el bloque se clasifica como la dirección asociada a dicha magnitud.

El estándar MPEG-7 define únicamente el uso del LEH, sin embargo el uso de información local puede no ser suficiente para una descripción correcta de la imagen. En este trabajo, consideramos dos histogramas adicionales para mejorar la precisión de este descriptor: el histograma global de bordes (GEH) y el histograma semi-global de bordes (SGEH). El GEH emplea información de todo el espacio de *I* y el SGEH considera distintas distribuciones de los bordes de *I*.

Es posible obtener los valores del GEH y SGEH directamente del LEH sin necesidad de llevar a cabo un procesamiento adicional.

La implementación de este descriptor en la GPU, utiliza un conjunto de hilos para procesar cada una de las 16 sub-imágenes (i.e. configuración de *4x4* sub-imágenes). A su vez, cada hilo realiza el cómputo requerido para determinar el tipo de borde asociado a cada uno de los bloques de las sub-imágenes, dando cobertura a todo el espacio de *I*.

Dado que los bloques seleccionados deben poseer la misma dimensión en sus valores de ancho y alto, existen regiones de las sub-imágenes que no son procesadas y que no repercuten en la clasificación del tipo de borde. En la Figura 3 se ilustra esta distribución sobre una imagen, donde las cuadrículas de color morado representan las sub-imágenes, las de color verde los bloques y las franjas de color naranja son las regiones no procesadas.

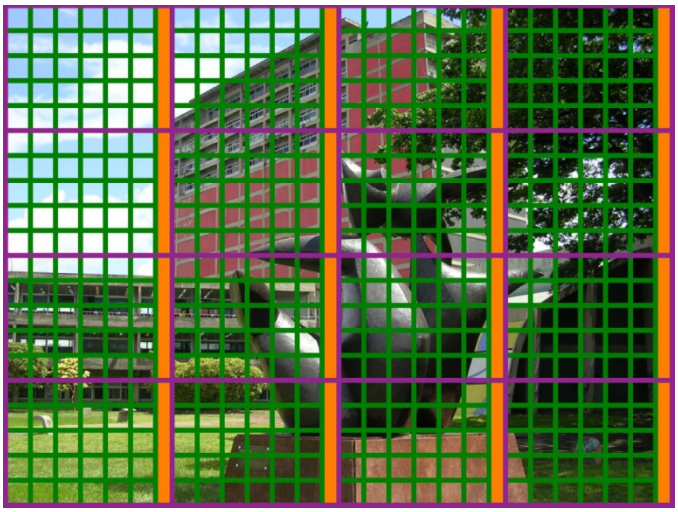


Figura 3: Un ejemplo de Distribución de las Sub-Imágenes, Bloques y Regiones no Procesadas

Con este descriptor, para calcular la distancia entre las imágenes *I* e *I'* se emplean los valores de los 3 histogramas calculados como una diferencia ponderada valor a valor de cada contenedor del LEH, GEH y SGEH.

C. Basado en Regiones

El descriptor de forma basado en regiones (RBSD) representa la distribución de píxeles dentro una imagen *I*. Dado que está basado en los bordes de los objetos y en sus píxeles internos, es posible describir objetos complejos dentro de la imagen que posean múltiples regiones discontinuas, así como objetos simples con o sin agujeros. Este descriptor pertenece a la amplia variedad de técnicas basadas en momentos, haciendo uso de la transformada radial angular (ART) con valores complejos, sobre un disco unitario en coordenadas polares.

Entonces, una imagen *I* expresada en coordenadas polares, se puede representar por un conjunto de valores radiales y angulares. Para cada valor radial se define un conjunto de valores angulares, como se muestra en la Figura 4. A medida que el valor radial aumenta, el número de componentes angulares generadas es mayor.

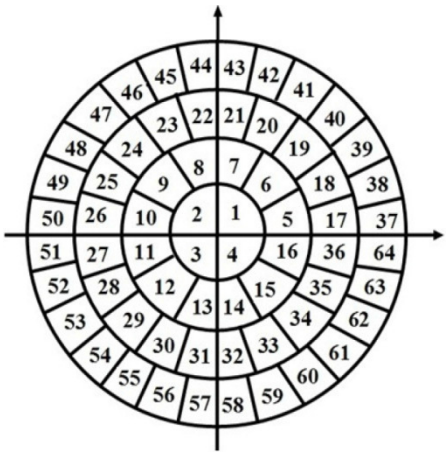


Figura 4: Representación de una Imagen en Coordenadas Polares para la Aplicación del RSBD

La idea consiste en dividir un disco unitario en sectores circulares no solapados y realizar una correspondencia directa a una imagen *I* de tamaño *M x N* (la cual se divide en rectángulos no solapados). Posteriormente, el cálculo de los coeficientes de la ART se realiza mediante (3).

En nuestra implementación paralela de (3), cada conjunto de hilos de la tarjeta gráfica ejecuta un valor angular y radial de las componentes de orden $0 \leq p < 3$ y $0 \leq q < 12$. El conjunto de valores generados se determina por el conjunto de hilos como se muestra en la Figura 5, acumulándolos una vez calculado.

Existen $3 \times 12 = 36$ posibles coeficientes de la ART así como un máximo de $4xN+2$ componentes angulares (donde N representa la dimensión de una imagen cuadrada). Cada conjunto de hilos calcula el aporte a (3) para cada uno de los posibles valores de estas componentes de la ART de orden p y q. Entonces, se requiere el doble de espacio de almacenamiento, $8xN+4$, por cada combinación de valores de p y q, dado que los coeficientes vienen representados por números complejos, almacenando sus componentes real e imaginaria.

Al emplear este descriptor para comparar dos imágenes *I* e *I'*, la similitud entre ambas puede determinarse al calcular la diferencia normalizada de las magnitudes de los coeficientes de la ART de *I* e *I'*.

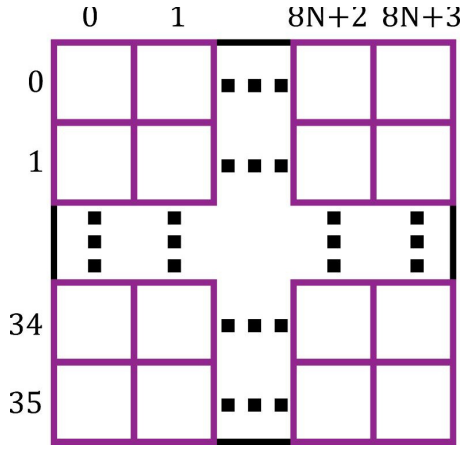


Figura 5: Distribución de los Conjuntos de Hilos para la Obtención del RSBD

Una vez calculados los tres descriptores visuales, se realizaron experimentos para determinar la eficacia y eficiencia de nuestra propuesta en un ambiente paralelo dentro de la tarjeta gráfica.

V. Experimentos y Resultados

Las pruebas realizadas fueron ejecutadas en una PC convencional con procesador *Core 2 Quad*, memoria RAM de 4 GB, y una tarjeta gráfica NVIDIA *GeForce* GT 520 con soporte para CUDA [17] en su versión 5.5. El software empleado fue bajo el sistema operativo *Windows 8* con el compilador de *Visual Studio 2012* y la biblioteca *OpenCV* [18] en su versión 3.0 para la manipulación de las imágenes. El lenguaje de programación CUDA C fue empleado para la paralelización de los descriptores.

Dada la versatilidad de nuestra propuesta, se dividió la realización de las pruebas en cuatro grandes módulos que consisten en la experimentación de cada uno de los descriptores propuestos y un caso de estudio para demostrar la precisión de éstos, empleando fotomosaicos.

A. Descriptor de Colores Dominantes

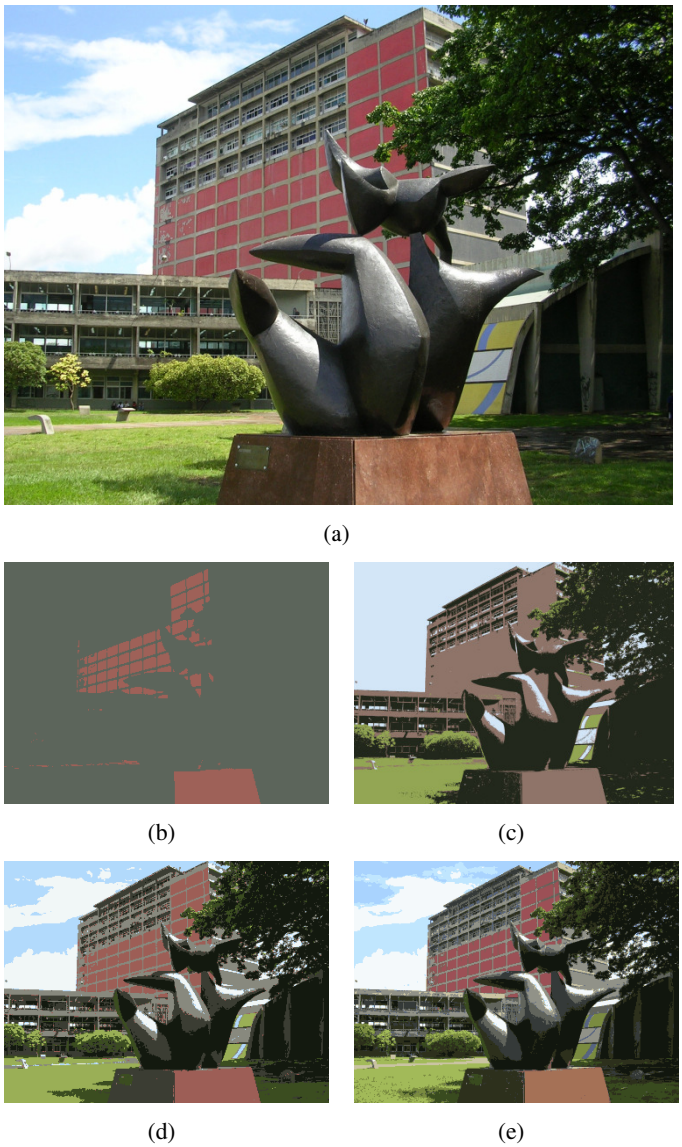
Se ha realizado una comparación entre 3 posibles valores que puede adoptar el parámetro k del algoritmo k-medias utilizado por este descriptor para la cuantización de una imagen de 2048×1536 píxeles. El tamaño de la imagen se debe a la consideración de un número de píxeles significativo para probar la eficiencia de nuestra propuesta (i.e. un total de $3.145.728$ píxeles).

La Tabla I muestra una comparación de los tiempos de ejecución del proceso de cuantización de una imagen para distintos valores de k (de un total de 15 ejecuciones), y el máximo error cuadrático medio porcentual (MSE) entre una imagen y su cuantización.

Tabla I: Tiempos y Error Cuadrático Medio de las Imágenes Cuantizadas

Valor de *k*	Tiempo medido en ms	Error Cuadrático Medio
2	274	79,86 %
4	504	38,09 %
8	1.021	9,16 %
16	1.580	6,67 %

En nuestras pruebas, se considera un valor de $k = 8$ como óptimo dado que dicho número de colores es suficiente para representar la información de color relevante con un error aceptable, como se presenta en la Figura 6. De esta manera, el DCD permite clasificar de forma adecuada una imagen dadas las características de color, con un algoritmo de agrupamiento como *k*-medias empleando 8 colores representativos de toda la imagen.



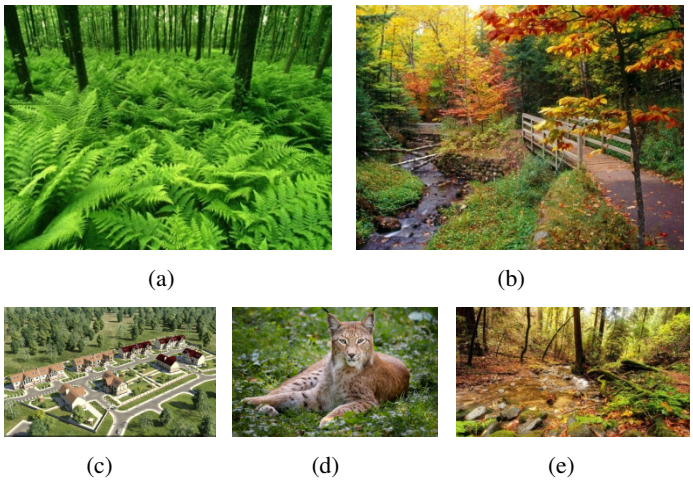
(a) (b) (c) (d) (e)

Figura 6: Diferentes Valores *k* de una: (a) Imagen Original, (b) k = 2, (c) k = 4, (d) k = 8 y (e) k = 16

B. Descriptor de Histograma de Bordes

Para este caso, nuestros experimentos consistieron en demostrar que el GEH y el SGEH incrementan la precisión del descriptor al considerar no solamente información local de bordes sino también patrones presentes en regiones más amplias. Para ello, se construyó una base de datos de 2.340 imágenes heterogéneas para que, dada una imagen original I y una imagen I' de la base de datos, se obtenga una lista de imágenes $I'_1, I'_2,\ldots, I'_k$ que represente los k mejores candidatos (i.e. según la medida de similitud para este descriptor) similares a I.

La Figura 7 muestra los k mejores candidatos para el EHD sin considerar el GEH y el SGEH, con $k = 4$.

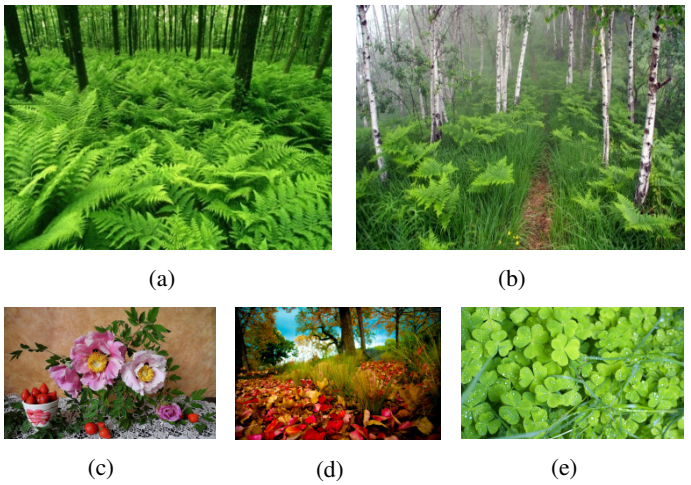


(a) (b) (c) (d) (e)

Figura 7: Aplicación del EHD sin Utilizar el GEH y SGEH sobre una Base de Datos de Imágenes y Como Imagen Base (a), y los Mejores Candidatos Obtenidos, de Mayor a Menor Similitud de (b) a (e)

Por otro lado, la incorporación de los histogramas GEH y SGEH permite obtener mejores candidatos al considerar información no solamente local a los bordes. Como se muestra en la Figura 8, el EHD presenta una mejoría en la similitud visual, considerando las imágenes I'_i de la base de datos.

Nótese que emplear el GEH y SGEH permite obtener detalles globales de la imagen original y determinar la presencia de helechos, troncos de árboles y ubicación de éstos. Las imágenes restantes presentan hojas y tallos asociados a la estructura de bordes de la imagen. Es importante destacar que el color no es considerado por este descriptor, basándose únicamente en los valores de sus intensidades.



(a) (b) (c) (d) (e)

Figura 8: Aplicación del EHD Empleando el GEH y SGEH sobre una Base de Datos de Imágenes y como Imagen Base (a) y los Mejores Candidatos Obtenidos, de Mayor a Menor Similitud de (b) a (e)

El tiempo promedio empleado para el cálculo del EHD asociado a una imagen es de aproximadamente 457 ms, siendo inferior al tiempo empleado por el DCD, dada la independencia

de las tareas que se llevan a cabo, aprovechando de mejor manera el paralelismo provisto por la arquitectura.

C. Descriptor de Forma Basado en Regiones

El EHD junto con el RBSD son los descriptores que permiten explotar en mayor medida el paralelismo provisto por la arquitectura CUDA. En el caso de este último, dado que el cálculo de los coeficientes de orden p y q de la ART es independiente para cada uno.

Para demostrar la efectividad de nuestra propuesta, se construyó una base de datos de 257 imágenes segmentadas a dos colores de marcas, logos y objetos con formas simples, donde se ha evaluado la precisión del RBSD. Tal como se explicó en la sección anterior, se calculan los coeficientes de la ART e igualmente que el EHD, se obtiene una lista de imágenes I'_1, I'_2,…, I'_k que representan los k mejores candidatos similares a una imagen base I.

Es importante destacar que este descriptor funciona de forma eficiente para imágenes con poco ruido ya que es muy sensible a cambios bruscos y continuos de frecuencia en la señal de una imagen. Por ello, se debe realizar un pre-procesamiento de la imagen llevando a cabo un proceso de umbralización (i.e. segmentación a dos tonos).

Otro aspecto a resaltar consiste en el tamaño de las imágenes, el cual no es superior a *100 x 100* píxeles debido a la gran cantidad de cálculos y el tiempo en obtener el resultado. Para tener una medida del tiempo, se realizó una versión secuencial empleando MATLAB para compararla con la versión desarrollada en CUDA C. La proporción en tiempo de ejecución es aproximadamente *38x* más rápida en la versión CUDA C con un tiempo promedio de 247 ms para 20 ejecuciones.

Una implementación secuencial en lenguaje C/C++ es aproximadamente *1.25x* más veloz que su contraparte en MATLAB - bajo las mismas condiciones - siendo igualmente más lenta que una versión desarrollada bajo algún ambiente paralelo como CUDA C.

Los descriptores de manera individual funcionan adecuadamente para un dominio de imágenes con ciertas restricciones de acuerdo a la similitud que se desea lograr (e.g. dimensiones, número de colores, variaciones abruptas y continuas de la frecuencia de la señal, etc.). Así, resulta interesante explorar estos tres descriptores de forma combinada para obtener un resultado adecuado en dominios generales dentro del procesamiento digital de imágenes.

De esta forma, se plantea un caso de estudio basado en fotomosaicos como una operación dentro de un sistema de búsqueda de imágenes de acuerdo a su color, textura y forma (i.e. sistema CBIR – *Content-Based Image Retrieval*).

D. Caso de Estudio: Fotomosaicos

Según Silvers [19], un mosaico se define como una obra compuesta por piezas de madera, piedra, cerámica, vidrio u otro material unidas mediante algún aglomerante, las cuales poseen diversas formas y colores, formando composiciones decorativas. De esta definición se desprende que un fotomosaico es un mosaico donde las piezas que lo componen son imágenes, las cuales se ordenan en una malla de parches sin que se solapen ni existan vacíos entre ellas.

La particularidad de los fotomosaicos radica en que, si existe una corta distancia entre éstos y el punto de vista, pueden apreciarse las imágenes que los componen. A medida que esta distancia aumenta, puede percibirse la silueta de una imagen formada por las imágenes observadas desde una corta distancia.

Para nuestros experimentos, se emplea la misma base de datos de 2.340 imágenes utilizada en las pruebas del EHD. Entonces, para cada imagen de la base de datos se calculan los descriptores de color, textura y forma explicados en nuestra propuesta. Posteriormente, se selecciona una imagen base I_b o imagen a utilizar para generar el fotomosaico, la cual se divide en parches. Para cada parche de I_b se calculan estos descriptores y se busca en la base de datos la imagen que presente mayor similitud con respecto al parche. El objetivo es conseguir y sustituir el candidato óptimo para cada parche de I_b.

Tomando una imagen I_b de tamaño *812 x 812* píxeles, ver Figura 9, con los descriptores normalizados en el rango [0,1] y fijando un número máximo de 100 veces que una imagen candidata puede ser seleccionada como óptima para evitar su continua aparición en la malla de parches, se tiene la Figura 10 que presenta algunos resultados con ciertas modificaciones en los parámetros de entrada.

Primeramente, la selección de I_b para la generación del fotomosaico no es la ideal debido a la gran presencia de colores en degradación sobre una misma región. Sin embargo, nuestro objetivo es verificar la eficacia de los descriptores de color, textura y forma. Los fotomosaicos generados que se observan en la Figura 10 presentan una gran cantidad de ruido debido a la elevada diferencia visual entre cada parche y sus adyacentes. Con una colección homogénea de imágenes se permitirá la atenuación de este ruido, ya que se dispone de mayor cantidad de imágenes similares que pueden ser seleccionadas por los descriptores visuales para ser sustituidas en la malla de parches definida sobre I_b.

Los fotomosaicos de la Figura 10a y Figura 10c han sido generados estableciendo la misma ponderación para cada uno de los descriptores visuales. Estos capturan la silueta básica de la imagen pero dada la heterogeneidad de las imágenes presentes en la base de datos (lo cual se ve reflejado en la diversidad de texturas y formas presentes), degradan el rendimiento del EHD y el RBSD.

Por otro lado, los fotomosaicos de la Figura 10b y Figura 10d han sido generados estableciendo una ponderación total para el DCD, ignorando la información dada por el EHD y el RBSD. Estos fotomosaicos presentan una mejora sustancial en su calidad visual, especialmente el fotomosaico de la Figura 10d con respecto a la Figura 10c, presentándose el ruido de forma más homogénea en toda la imagen. Esto muestra que la sensibilidad del DCD es menor que los otros descriptores visuales considerados (para el caso de bases de datos de imágenes heterogéneas).

Tabla II: Parámetros Empleados para la Generación de los Fotomosaicos

Figura	Número de parches	Tamaño de parches	Tamaño del fotomosaico	Ponderación *DCD/EHD/RBDS*
10ª	50 x 50 = 2.500	100 x 100 píxeles	5.000 x 5.000 píxeles ~71.5Mb	0.34 / 0.33 / 0.33
10b	50 x 50 = 2.500	100 x 100 píxeles	5.000 x 5.000 píxeles ~71.5Mb	1.00 / 0.00 / 0.00
10c	100 x 100 = 10.000	100 x 100 píxeles	10.000 x 10.000 píxeles ~286Mb	0.34 / 0.33 / 0.33
10d	100 x 100 = 10.000	100 x 100 píxeles	10.000 x 10.000 píxeles ~286Mb	1.00 / 0.00 / 0.00

Figura 9: Imagen Base I_b Empleada para la Generación de Fotomosaicos

Por otro lado, los fotomosaicos de la Figura 10b y Figura 10d han sido generados estableciendo una ponderación total para el DCD, ignorando la información dada por el EHD y el RBSD. Estos fotomosaicos presentan una mejora sustancial en su calidad visual, especialmente el fotomosaico de la Figura 10d con respecto a la Figura 10c, presentándose el ruido de forma más homogénea en toda la imagen. Esto muestra que la sensibilidad del DCD es menor que los otros descriptores visuales considerados (para el caso de bases de datos de imágenes heterogéneas).

En la Tabla II se presentan las características empleadas entre la imagen I_b y los fotomosaicos generados de la Figura 10.

VI. Conclusiones y Trabajos Futuros

En esta investigación se ha mostrado una versión de descriptores visuales del estándar MPEG-7 haciendo uso de la GPU para acelerar los procesos involucrados, manteniendo la eficacia de éstos con gran nivel de precisión en imágenes similares.

En nuestra experimentación se consideró cada uno de los descriptores de manera individual y su evaluación con una base de datos de imágenes bajo un esquema paralelo enfocado en la arquitectura CUDA. Así, cada descriptor permite identificar imágenes basado en un criterio de búsqueda para diversas aplicaciones.

Por otro lado, seleccionar como caso de estudio la generación de fotomosaicos basados en sistemas de tipo CBIR permite simular el uso de vectores característicos, tal como ocurre en los motores de búsqueda convencionales, para ubicar de forma rápida un conjunto de candidatos basados en una implementación paralela en la GPU de descriptores visuales del estándar MPEG-7. Es importante destacar, que el uso de las GPUs para acelerar el tiempo de cómputo se considera una rama de investigación actual debido a sus bajos costos, fácil uso y escalabilidad en las aplicaciones computacionales.

La calidad visual de los fotomosaicos depende enteramente de la combinación de los descriptores visuales. Esta calidad puede verse incrementada al realizar una selección de una base de datos de imágenes apropiada que sea lo suficientemente representativa en cuanto a las necesidades de una aplicación de esta naturaleza. Esta clase de aplicaciones es sumamente sensible a la base de datos de imágenes utilizada, por lo que es ideal que su contenido sea lo más homogéneo posible, tal que no se generen cambios visuales drásticos entre los parches de los fotomosaicos.

En el futuro se plantea la idea de implementar y probar otros descriptores visuales del estándar con el fin de agregar una mayor cantidad de parámetros en aplicaciones, en el contexto de búsquedas de tipo CBIR y realizar pruebas visuales basadas en las funciones de similitud definidas. Igualmente, se propone estudiar con más detalle los impactos al emplear un tipo de descriptor visual de acuerdo a ciertos criterios de búsqueda dentro de aplicaciones especializadas, con el fin de precisar el descriptor y sus parámetros de acuerdo a la naturaleza de dicha búsqueda (i.e. tomando el cuenta si se desea buscar por similitud de tono, forma, relieve, entre otros).

Referencias

[1] Y. Rui, T.S. Huang, and S.F. Chang, *Image Retrieval: Current Techniques, Promising Directions, and Open Issues*, Journal of Visual Communications and Image Representation, vol. 10, pp. 39–62, 1999.

[2] A.W. Smeulders, S. Member, M. Worring, S. Santini, A. Gupta, and R. Jain, *Content-Based Image Retrieval at the End of the Early Years*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1349–1380, 2000.

[3] R. Datta, D. Joshi, J. Li, and J.Z. Wang, *Image Retrieval: Ideas, Influences, and Trends of the New Age*, ACM Computing Surveys, vol. 40, no. 2, pp. 5:1–5:60, Abril 2008.

[4] M. Angelides and H. Aguis, *The Handbook of MPEG Applications: Standards in Practice*, 1st edition, John Wiley & Sons, 2011.

[5] J. Orallo, M. Ramírez y C. Ferri, *Introducción a la Minería de Datos*, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Pearson Prentice Hall, 2008.

[6] S. Sergyán, *Precision Improvement of Content-Based Image Retrieval Using Dominant Color Histogram Descriptor*, in proceedings of 1st WSEAS International Conference on Image Processing and Pattern Recognition at Budapest, pp. 197–203, Hungary, December 2013.

[7] V. Felipe y E. Ramírez, *K-Medias Empleando la GPU*, en las memorias del III Simposio Científico y Tecnológico en Computación (SCTC), Sesión de Posters, pp. 152, 2014.

[8] D. Park, Y. Jeon, and C. Won, *Efficient Use of Local Edge Histogram Descriptor*, in proceedings of the ACM Workshops on Multimedia, pp. 51–54, 2000.

[9] K. Hosny, *Accurate Computation of ART Coefficients for Binary Images*, in proceeding of the 1st Taibah University International Conference on Computing and Information Technology, vol. 1, 2012.

[10] Y. Xin, M. Pawlak, and S. Liao, *Accurate Computation of Zernike Moments in Polar Coordinates*, IEEE Transactions on Image Processing, vol. 16, pp. 581–587, 2007.

[11] M. Slomp, M. Mikamo, B. Raytchev, T. Tamaki, and K. Kaneda, *GPU-Based SoftAssign for Maximizing Image Utilization in Photomosaics*, International Journal of Networking and Computing, vol. 1, no. 2, pp. 211–229, 2011.

[12] G. Di Blasi, G. Gallo, and M. Petralia, *Smart Ideas for Photomosaic Rendering*, in proceedings of the Eurographics Italian Chapter, pp. 267–271, 2006.

[13] J. Kim and F. Pellacini, *Jigsaw Image Mosaics*, in proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH, pp. 657–664, 2002.

[14] Google Inc. *Google Goggles*, http://www.google.com/mobile/goggles.

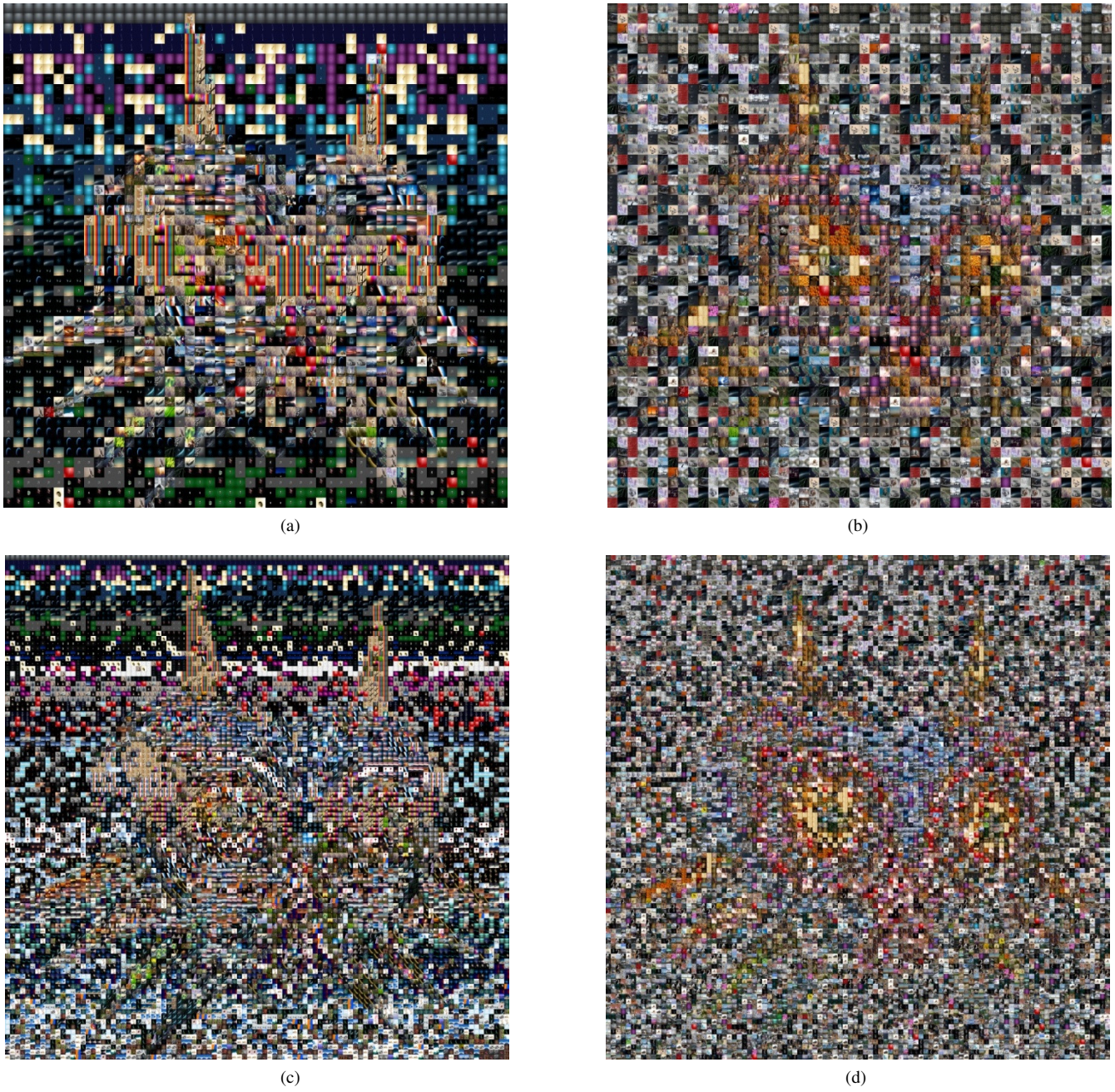
[15] Image Searcher Inc. *CamFind*, http://www.camfindapp.com.

[16] J. Ricard, D. Coeurjolly, and A. Baskurt, *Generalization of Angular Radial Transform*, in proceedings of the International Conference on Image Processing, vol. 4, pp. 2211–2214, October 2004.

[17] NVIDIA Corporation. *CUDA*, https://developer.nvidia.com/cuda-zone.

[18] Itseez. *OpenCV*, http://opencv.org.

[19] R. Silvers, *Photomosaics: Putting Pictures in Their Place*, M.S. thesis, Program in Media Arts & Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1996.



(a) (b)

(c) (d)

Figura 10: Fotomosaicos Generados con Diversos Parámetros en el Número de Parches y en la Ponderación de los Descriptores de Color, Textura y Forma. Con parches de 50 x 50 (a y b) y de 100 x 100 (c y d)