

Búsqueda por Similitud para Recuperación de Imágenes

Adrián Planas¹, Andrés Pascal¹, Anabella De Battista¹, Alejandra Díaz¹, Norma Edith Herrera²

¹Dpto. de Sistemas de Información – Univ. Tecnológica Nacional – FRCU – Entre Ríos
Argentina
{planasa, pascala, debattistaa, [adiaz](mailto:adiaz}@frcu.utn.edu.ar)}@frcu.utn.edu.ar

²Dpto. de Informática – Universidad Nacional de San Luis – Argentina
nherrera@unsl.edu.ar

Resumen. *En la búsqueda de imágenes por similitud en grandes bases de datos, es tan importante la eficiencia del sistema como su eficacia. La eficacia depende principalmente del preprocesamiento de las imágenes, de la técnica de extracción de características y de la función de distancia o disimilitud que se emplee. Por otro lado, los factores de mayor relevancia para la eficiencia del proceso son el costo de la función de distancia y el tipo de índice que se utilice para acelerar la búsqueda. En este trabajo se propone y evalúa experimentalmente una implementación completa orientada al uso de índices métricos para la búsqueda de imágenes por similitud, que adapta métodos y técnicas existentes para integrar en forma eficiente y eficaz estos elementos.*

Abstract. *Efficiency and effectiveness are both important issues in searching images by similarity in large databases. Efficiency depends mainly on preprocessing the images, features extraction techniques and the distance or dissimilarity function used. On the other hand, the factors most relevant to efficiency are the cost of the distance function and the index used to search. In this paper we propose and evaluate experimentally a complete implementation of similarity search of images using metric indexes and adapting methods and techniques to integrate efficiently and effectively these elements.*

1. Introducción

Las bases de datos tradicionales se construyen basándose en el concepto de búsqueda exacta: la base de datos es dividida en registros, teniendo cada registro campos completamente comparables; una consulta a la base retorna todos aquellos registros cuyos campos coincidan con los aportados en la búsqueda.

En la actualidad, muchas aplicaciones tienen como necesidad buscar en grandes bases de datos objetos que sean similares a uno. Este tipo de búsqueda se conoce con el nombre de *búsqueda por similitud* y surge en diversas áreas [3] entre las cuales encontramos el reconocimiento de imágenes. Como ejemplos ampliamente conocidos podemos mencionar las bases de datos de imágenes médicas, las bases de datos CAD y las imágenes en la web, entre otras. En todos estos casos, estructurar las imágenes en registros para adecuarlos al concepto tradicional de búsqueda exacta, es difícil y hasta imposible si la base de datos cambia más rápido de lo que se puede estructurar, como es el caso de la web. Aún cuando pudiera hacerse, las consultas que se pueden satisfacer con la tecnología tradicional son limitadas a variaciones de búsquedas exactas que carecen de sentido en el caso de las imágenes.

La mayoría de los sistemas comerciales existentes para recuperación de imágenes asocian texto a cada una de las imágenes y transforman el problema de búsqueda de imágenes en el problema de búsqueda en texto. Pero en muchas aplicaciones esto no es suficiente. Un verdadero sistema de recuperación de imágenes debe permitir dar una imagen como objeto de consulta y debe poder determinar la similitud entre ese objeto y cada una de las imágenes de la base de datos en forma eficiente, a fin de responder la consulta. Estos sistemas se denominan Content-based Image Retrieval Systems (CBIR) [4, 6, 7, 12], y están en pleno desarrollo.

El problema de búsqueda de imágenes puede abstraerse en un universo de objetos X y una función de distancia d que modela la similitud entre los objetos del universo. El par (X, d) se denomina *espacio métrico*. La base de datos será un subconjunto finito $U \subseteq X$. En este nuevo modelo de bases de datos [3], una de las consultas típicas que implica recuperar objetos similares es la *búsqueda por rango*, que consiste en, dado un elemento q , recuperar todos los objetos de la bases de datos cuya distancia a q esté dentro de un radio de tolerancia dado. Para resolver eficientemente las búsquedas por rango se preprocesa la base de datos con el fin de construir un *índice* diseñado para ahorrar cálculos al momento de responder la consulta.

La búsqueda por similitud aplicada a imágenes implica transformar las imágenes en vectores de características de esas imágenes, para luego insertar esos vectores en un índice métrico. Luego, ante una consulta, se debe transformar la imagen de consulta de la misma manera en que se transformaron las imágenes de la base de datos para poder proceder a buscar usando el índice.

Existen dos tareas que son cruciales en este proceso: una es convertir las imágenes en vectores; la otra es definir una función de distancia que permita comparar las imágenes. La primera tarea afecta directamente la eficacia del sistema, dado que las búsquedas se realizarán en base a las características extraídas de cada imagen. La segunda tarea afecta tanto la eficacia como la eficiencia; la eficacia porque la función de distancia modela formalmente lo que se entiende por similitud y la eficiencia porque el costo de búsqueda en el índice se ve directamente afectado por el costo de cálculo de la función de distancia y por la distribución de distancias que genera.

Si bien hay numerosos trabajos de investigación que se concentran en el preprocesamiento de las imágenes y extracción de características [5,13,14], las funciones de distancia [11,15,16] y los índices métricos [1,3], la mayoría lo hace por separado, sin estudiar la integración de estos aspectos. En este artículo proponemos una implementación particular de diversas técnicas que abarcan el proceso completo, adaptadas para que actúen en conjunto en forma eficiente y eficaz.

Este artículo está organizado de la siguiente manera: comenzamos en la sección 2 describiendo brevemente el trabajo relacionado. En la sección 3 describimos nuestra propuesta cuya evaluación experimental es presentada en la sección 4. Finalizamos en la sección 5 dando las conclusiones y las futuras líneas de trabajo.

2. Trabajo Relacionado

2.1. El Modelo de Espacios Métricos

Un *Espacio Métrico* se define como un par (X, d) donde X es el universo de objetos válidos del espacio y $d : X \times X \rightarrow \mathbb{R}^+$ es una función de distancia mide el grado de similitud entre los elementos de X . Esta función d cumple con las propiedades características de una función de distancia: *positividad* ($d(x, y) \geq 0$), *simetría* ($d(x, y) = d(y, x)$) y *desigualdad triangular* ($d(x, y) \leq d(y, z) + d(z, y)$). La base de datos es cualquier subconjunto finito $U \subseteq X$.

Una de las consultas típicas en este nuevo modelo de bases de datos es la búsqueda por rango, que se denota $(q, r)_d$ y se define como $(q, r)_d = \{u \in U : d(u, q) \leq r\}$. Una forma trivial de resolver una búsqueda por rango es examinando exhaustivamente la base de datos. Para evitar esta situación, se preprocesa la base de datos por medio de un algoritmo de indexación con el objetivo de construir una índice, diseñado para ahorrar cálculos en el momento de resolver una búsqueda.

En [3] se presenta un desarrollo unificador de las soluciones existentes en la temática. En dicho trabajo, se muestra que todos los enfoques para la construcción de índices en espacios métricos consisten en particionar el espacio en clases de equivalencia e indexar esas clases de equivalencia. Durante la búsqueda, por medio del índice, se descartan algunas clases, y se busca exhaustivamente en las restantes. En dicho trabajo se muestra que existen dos grupos de algoritmos de indexación: *basados en pivotes* y *basados en particiones compactas*. La diferencia entre estos dos grupos de algoritmos está en la forma en que definen las clases de equivalencia.

Uno de los principales obstáculos en el diseño de buenas técnicas de indexación es lo que se conoce con el nombre de *maldición de la dimensionalidad* [2, 3, 18]. El concepto de dimensionalidad está relacionado a la dificultad o facilidad de buscar en un determinado espacio métrico. La dimensión intrínseca de un espacio métrico se define en [3] como $\rho = \mu^2 / 2\sigma^2$, siendo μ y σ^2 la media y la varianza respectivamente de su histograma de distancias. Es decir que, a medida que la dimensionalidad intrínseca crece, la media crece y su varianza se reduce. Esto significa que el histograma de distancia se concentra más alrededor de su media, lo que influye negativamente en los algoritmos de indexación.

2.2. Búsquedas por Similitud sobre Imágenes

Tal como lo mencionáramos, la transformación de imágenes en vectores de características es una de las tareas principales que se deben realizar para aplicar búsquedas por similitud a un espacio métrico de imágenes. Claramente no todas las características de una imagen son importantes para todas las aplicaciones y pueden existir aplicaciones en las que sólo ciertas características específicas sean de utilidad. Como ejemplos de características de imágenes podemos mencionar la distribución de colores, la posición de los objetos en la imagen, la sombra de los objetos, la textura, etc.

Muchos sistemas de recuperación de imágenes se basan en histogramas de colores, que representan la distribución de colores en la imagen calculando la frecuencia de ocurrencia de cada color. Los histogramas tienen la ventaja de que contienen información importante sobre la imagen que representan y que son rápidos de computar, lo que facilita su aplicación a grandes bases de datos de imágenes.

Un punto importante aquí es el modelo de color usado para trabajar. Un modelo de color permite especificar los colores de manera normalizada. En este trabajo hemos elegido el modelo *RGB* dado que la mayoría de las imágenes digitalizadas están definidas en este espacio. Este modelo está compuesto por 3 dimensiones: *R*(rojo), *G*(verde) y *B*(azul), cada una de las cuales posee una cierta cantidad de divisiones. Entonces cada color está representado por un punto en este espacio; esto implica que para una profundidad de 24 bits existen $256^3 = 16.777.216$ de colores posibles.

Por esta razón, para calcular el histograma de colores de una imagen se divide el espacio de colores en un número determinado de regiones y luego, para cada región, se cuenta la cantidad de pixels que tienen colores en ese rango. Formalmente un histograma h para un espacio de colores de n regiones se puede definir como un vector $h = (h_1, h_2, \dots, h_n)$ donde h_i es la cantidad de pixels cuyo color pertenece a la región i .

Para poder aplicar búsquedas por similitud se necesita definir una función que mida distancia entre histogramas. Entre las funciones de distancia más ampliamente usadas encontramos la familia de distancias de Minkowski, en particular la distancia de Manhattan y la distancia Euclideana, la divergencia de Jeffrey y χ^2 para histogramas fijos, y la distancia cuadrática y la Earth Mover's Distance (EMD) cuando no hay una correspondencia fija entre los elementos de los histogramas [11, 16, 17].

3. Búsquedas de imágenes por similitud en grandes bases de datos

Como mencionamos en las secciones previas, la búsqueda de imágenes por similitud requiere técnicas que permitan determinar eficazmente cuales son los objetos de mayor parecido a una consulta dada. Para ello, no alcanza con la definición de una función de distancia (o disimilitud), sino que además se necesita tener una forma de preprocesar la imagen para obtener una representación adecuada al problema que se está tratando. Por otro lado, cuando se tratan grandes bases de datos, el grado de eficiencia de las consultas se vuelve un factor crítico para la utilidad de estos sistemas.

Este último problema, en nuestra implementación, se resuelve utilizando índices sobre espacios métricos. Sin embargo, para ello es necesario que la dimensionalidad intrínseca del espacio generado sea relativamente baja, ya que de otra manera los índices pierden su eficiencia debido al efecto de la *maldición de la dimensionalidad*.

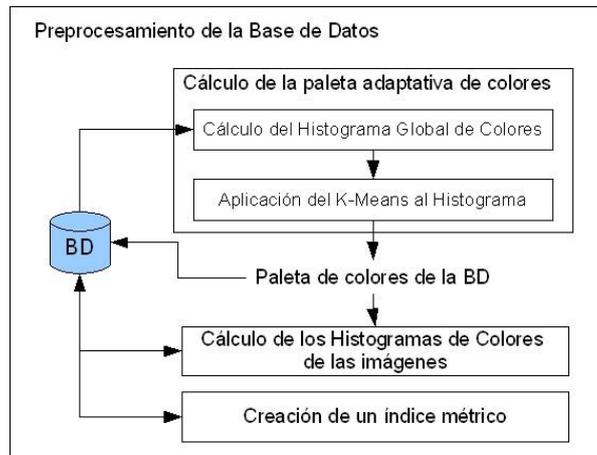


Figura 1. Preprocesamiento de la Bases de Datos de Imágenes.

3.1. Búsquedas por Similitud sobre Imágenes

En este trabajo suponemos que las imágenes se encuentran en el espacio RGB y están normalizadas en cuanto a la profundidad de color, por ejemplo, en 16 o 24 bits.

El proceso propuesto consta de una etapa de preprocesamiento de la base de datos (Figura 1) y otra posterior para la realización de consultas (Figura 2). La etapa de preprocesamiento tiene dos objetivos:

1. Calcular la paleta de colores reducida y adaptada a las imágenes de la base de datos.
2. Calcular los histogramas de colores de las imágenes, en base a la paleta definida en el punto anterior.

Para el primer punto, se procesa la base de datos (o una muestra) como si fuese una sola imagen, calculando el histograma real de colores. Como no es posible trabajar directamente con este histograma debido a la cantidad de dimensiones que se generarían en los vectores característicos, se utiliza el algoritmo k-means [9] para la construcción de una paleta de colores adaptada [8] a las características de la base de datos. Mediante esta técnica se produce una reducción significativa de la cantidad de dimensiones, pasando por ejemplo, de más de 16 millones de colores a sólo 25 y manteniendo la apariencia visual de la imagen, tal como se muestra en la Figura 3.

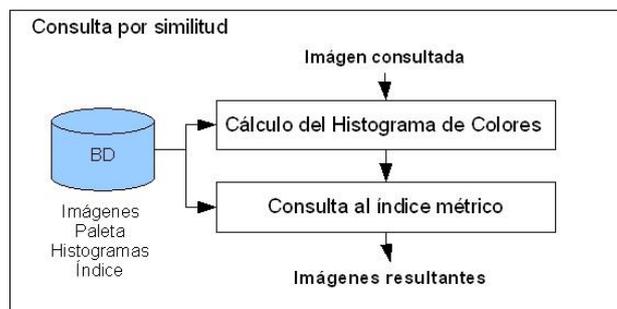


Figura 2. Búsqueda de imágenes por Similitud.

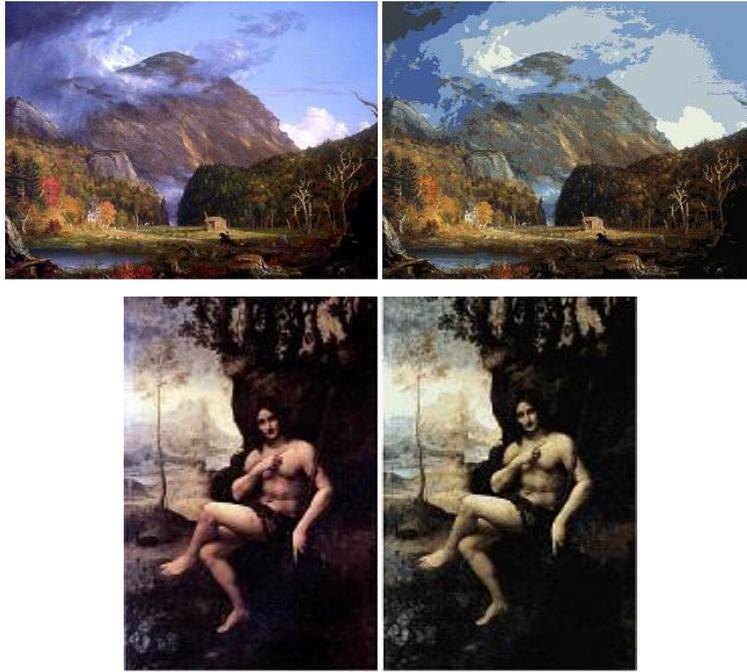


Figura 3. Dos ejemplos. Imagen original a la izquierda (16 millones de colores) y transformada con paleta adaptativa a la derecha (25 colores).

Una vez obtenida la paleta de colores, se procesan las imágenes calculando los histogramas mediante la transformación de cada color en el más cercano de la paleta. Para clasificar cada pixel se utiliza la distancia euclídeana entre colores. Posteriormente se construye un índice métrico utilizando los histogramas como vectores característicos de cada imagen, y empleando como función de distancia la distancia euclídeana o alguna variante, como veremos más adelante. Luego del preprocesamiento, la base de datos contiene las imágenes, la paleta de colores, los vectores característicos correspondientes a cada imagen y el índice métrico.

Ante una consulta, primero se calcula el vector característico de la imagen consultada en base a la paleta de la base de datos, y luego se realiza la búsqueda sobre el índice métrico.

Funciones de distancia utilizadas

Existen muchas funciones propuestas para medir la el grado de similitud entre dos imágenes [11, 15,17]. Una comparación exhaustiva de dichas medidas excede el alcance de este trabajo. En su lugar, utilizamos algunas de las funciones más comunes, tales como la distancia euclídeana y la distancia de Manhattan y dos propuestas de variantes de la función Δ -distance [10].

Las distancias Euclídeana y de Manhattan forman parte de la familia de distancias de Minkowski, y se definen de la siguiente manera: sean X e Y los vectores característicos de dos imágenes de k dimensiones:

$$\textit{Distancia Euclídiana: } de(X, Y) = \left[\sum_{i=1}^k |x_i - y_i|^2 \right]^{1/2}$$

$$\textit{Distancia de Manhattan: } dm(X, Y) = \sum_{i=1}^k |x_i - y_i|$$



Figura 4. Falsos positivos en la búsqueda de imágenes.

Si bien con estas funciones se obtienen buenos resultados, existe un problema cuando se comparan imágenes que poseen distribuciones similares de colores, pero donde los puntos de colores equivalentes se encuentran en distintas partes de las imágenes. La Figura 4 muestra un ejemplo de este caso: la imagen que actúa como query (izquierda) genera un falso positivo (derecha).

Para solucionar este problema utilizaremos otras dos funciones que surgen de adaptar dos casos particulares de la familia de funciones Δ -distance. Esta familia de funciones divide recursivamente en cuadrantes las imágenes, tal como se muestra en la Figura 5 (para dos divisiones recursivas), para el cálculo de los histogramas. Luego se calcula el histograma global, los cuatro resultantes de la primera división, los ocho correspondientes a la segunda división, y así sucesivamente. Esto genera un árbol donde el primer nivel representa la imagen completa y cada nivel posterior es el resultado de una nueva división. Luego la función de distancia se calcula en base a los histogramas de todos los niveles. El problema de este mecanismo es que genera una cantidad importante de dimensiones, que a su vez produce un incremento importante en la dimensionalidad intrínseca, situación que se quiere evitar. En su lugar, utilizamos funciones que sólo toman un nivel del árbol, disminuyendo así la dimensionalidad.

De esta manera en nuestra función adaptada cada imagen se divide en n rectángulos de igual tamaño y se calculan los histogramas locales correspondientes a cada uno de ellos, generando una matriz característica de n filas (cuadrantes) por k columnas (colores).

Las funciones se definen de la siguiente manera: sean X e Y las matrices características de dos imágenes de k dimensiones y n cuadrantes:

$$\textit{Distancia Euclideana por Cuadrantes: } dec = \frac{\sum_{j=1}^n de(x_j, y_j)}{n}$$

$$\textit{Distancia Manhattan por Cuadrantes: } dmc = \frac{\sum_{j=1}^n dm(x_j, y_j)}{n}$$

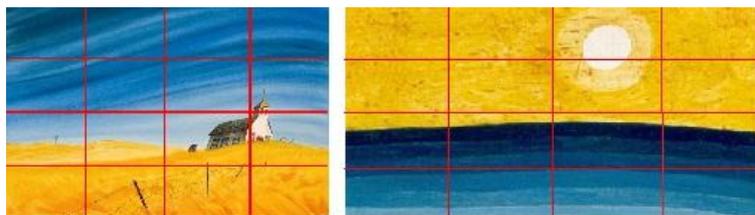


Figura 5. División en cuadrantes para el cálculo de histogramas

Cuando se comparan dos imágenes, se calcula la Distancia Euclideana entre cada par de cuadrantes ubicados en el mismo lugar de ambas imágenes y luego se promedian los valores resultantes. Así se controla que la distribución de los colores sea similar en los n cuadrantes de las imágenes comparadas. Estas funciones conservan las propiedades de toda función métrica, ya que se calculan como una combinación lineal de las funciones Euclideana y Manhattan para cada uno de los vectores correspondientes a los n cuadrantes. Por otro lado, tal como lo muestran los resultados experimentales, la dimensionalidad intrínseca de los espacios generados por estas funciones se mantiene en valores suficientemente bajos como para el uso eficiente de índices métricos.

4. Resultados Experimentales

Para la evaluación experimental del método, se utilizó una base de datos de 1000 imágenes de pinturas extraídas de Internet, en formato JPEG de 24 bits de profundidad, en el espacio de colores RGB, normalizadas a 640 pixels como máximo para el ancho o alto (el que sea mayor). Luego se definieron tres grupos de imágenes de consulta para verificar la eficacia del método ante distintas situaciones comunes de consulta, tales como que la imagen consultada varíe en brillo, contraste o contenga ruido.

Las consultas se prepararon tomando imágenes al azar de la base de datos y aplicando las siguientes modificaciones: *Rotación*: giro de la imagen de 15 grados y 30 grados en ambas direcciones. *Recorte*: eliminación del 20% de la imagen para cada uno de sus lados por separado. *Brillo*: aumento y disminución de brillo de la imagen en un 10% y 20%. *Contraste*: aumento y disminución de brillo de la imagen en un 10% y 20%. *Ruido*: modificación del color de un 15% de los pixels al azar.

Algunos ejemplos de consultas se ven en la Figura 6. La primera imagen es la original y luego aparecen las consultas en las cuales se modifica el brillo, el contraste, se rota en sentido horario y por último se recorta la parte superior de la imagen.

De esta manera se obtuvieron 72 imágenes de consulta que simulan consultas reales donde la imagen que se busca contiene variaciones respecto a la que se encuentra en la base de datos. Además, se agregó un grupo de prueba de imágenes de consulta inexistentes en la base de datos, para verificar que las distancias resultantes de comparar imágenes distintas sea mayor que las obtenidas al comparar la misma imagen pero con modificaciones. Se definieron dos indicadores para medir la eficacia del método:

1. Porcentaje de aciertos en primer lugar, es decir, el porcentaje de respuestas donde la imagen consultada es una modificación de la imagen de la base de datos con menor distancia.
2. Porcentaje de aciertos en los primeros cinco lugares; porcentaje de respuestas en la cual la imagen consultada es una modificación de una de las cinco imágenes de la base de datos con menor distancia.



Figura 6. Tratamiento de imágenes.

Cuadro 1. Comparación de Funciones de Distancia

Funcion de Distancia	Aciertos en 1er. Lugar	%	Aciertos dentro de los 5	%	D.I.
Euclideana por Cuadrantes	60	83,33	64	88,89	11,41
Manhattan por Cuadrantes	64	88,89	66	91,67	19,59
Euclideana	48	66,67	50	69,44	3,51
Manhattan	55	76,39	58	80,56	5,50

Las funciones de distancia utilizadas fueron las cuatro anteriormente nombradas: Euclideana, Manhattan, Euclideana por Cuadrantes y Manhattan por Cuadrantes (en ambos casos con $n = 16$ cuadrantes) Por otro lado, se evaluó la Dimensionalidad Intrínseca del espacio generado por cada una de las cuatro funciones, como medida que constituye el factor más importante en la eficiencia de las consultas utilizando índices métricos cuando el costo computacional de las distintas funciones de distancia es del mismo orden computacional.

Los resultados obtenidos se muestran en el Cuadro 1. Como podemos observar, la función que obtiene el mayor porcentaje de eficacia es la Distancia Manhattan por Cuadrantes, aunque es la que genera un espacio el espacio de mayor dimensionalidad, es decir, la de menor eficiencia. Por otro lado, se verifica que la introducción de la división en cuadrantes ya sea para la Distancia de Manhattan como para la Euclideana, mejora la calidad de los resultados entre un 7% y hasta más de un 20%. Como era de esperar, también aumenta la dimensionalidad del espacio, debido a que la cantidad de dimensiones se multiplica por 16, es decir, de 25 a 400 dimensiones. Existe entonces una solución de compromiso entre la eficacia y la eficiencia del método, determinada por la medida de disimilitud que se elija. En esta situación, consideramos que la Distancia Euclideana por Cuadrantes, posee un buen comportamiento en cuanto a la calidad de sus resultados, y una dimensionalidad intrínseca que se encuentra dentro de los valores aceptables para el uso de índices métricos.

5. Conclusiones y Trabajo Futuro

En este artículo presentamos una implementación completa de búsqueda de imágenes por similitud orientada al uso de índices métricos para mantener la eficiencia en bases de datos de gran tamaño. El proceso incluye el preprocesamiento de la base de datos, la extracción de características de las imágenes y las funciones de distancia que pueden ser utilizadas considerando como factores importantes la eficacia de las consultas y que la dimensionalidad intrínseca del espacio generado esté dentro de los valores aceptables para el uso de índices métricos. También se presentan resultados experimentales que verifican el buen comportamiento de la combinación de las técnicas empleadas para un caso particular. Actualmente estamos trabajando en la inclusión de funciones de distancia que consideren otros aspectos tales como textura o forma.

Referencias

- [1] E. Chavez and K. Figueroa. *Faster proximity searching in metric data*. Proceedings of MICAI 2004. LNCS 2972, Springer, 2004.

- [2] E. Chavez and G. Navarro. *Probabilistic proximity search: Fighting the curse of dimensionality in metric spaces*. Information Processing Letters, 2002.
- [3] E. Chavez, G. Navarro, R. Baeza-Yates, and J.L. Marroquin. *Searching in metric spaces*. ACM Computing Surveys, 33(3): 273–321, sep 2001.
- [4] R. Datta, D. Joshi, J. Li and J. Z. Wang. *Image retrieval: ideas, influences, and trends of the new age*. ACM Computing Surveys, 2008.
- [5] T. Deselaers, D. Keysers, and H. Ney. *Features for image retrieval - a quantitative comparison*. In DAGM Symposium, Pattern Recognition, 228-236, 2004.
- [6] T. Deserno, S. Antani, and L. Rodney Long. *Content-based image retrieval*. International Journal of Healthcare Information Systems and Informatics, 2009.
- [7] M. Koskela. *Interactive image retrieval using self-organizing maps*. Helsinki University of Technology: Dissertations in Computer and Information Science, 2003.
- [8] W. Kheng Leow and R. Li. *The analysis and applications of adaptive-binning color histograms*. Comput. Vis. Image Underst., 94:67–91, April 2004.
- [9] A. Likas, N. Vlassis, Aristidis Likas, Nikos Vlassis, and J.J. Verbeek. *The global k-means clustering algorithm*. Pattern Recognition, 36:451–461, 2001.
- [10] Maude Manouvrier, Marta Rukoz, and Genevieve Jomier. *A generalized metric distance between hierarchically partitioned images*. MDM/KDD workshop, 2005.
- [11] Y. Rubner, C. Tomasi, and L. J. Guibas. *The earth movers distance as a metric for image retrieval*. International Journal of Computer Vision, 40:99–121, 2000.
- [12] R. C. Veltkamp and M. Tanase. *Content-based image retrieval systems: A survey*. Technical Report UU-CS-2000-34, October 2000.
- [13] T. Deselaers, D. Keysers and H. Ney. *Features for Image Retrieval: An Experimental Comparison*. Information Retrieval, v.11 n.2, p.77-107, April 2008.
- [14] A. Shahbahrami and D. Juurlink. *Comparison Between Color and Texture Features for Image Retrieval*. Proceedings of the 19th Annual Workshop on Circuits, Systems and Signal Processing, The Netherlands, November 2008.
- [15] Y. Rubner, J. Puzicha, C. Tomasi and J. Buhmann. *Empirical Evaluation of Dissimilarity Measures for Color and Texture*. Computer Vision and Image Understanding, 2001.
- [16] J. Smith. and S. Chang. *Integrated spatial and feature image query*. Multimedia Systems, vol. 7, pag.129-140, Marzo 1999.
- [17] J. Puzicha, T. Hofmann and J. M. Buhmann. *Non-parametric similarity measures for unsupervised texture segmentation and image retrieval*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pag. 267-272, 1997.
- [18] E. Chavez and G. Navarro. *A Probabilistic Spell for the Curse of Dimensionality*. Proc. 3rd Workshop on Algorithm Engineering and Experiments, 2001.