

Búsqueda por Similitud en Bases de Datos de Objetos 3D

Benjamin Bustos**
Centro de Investigación de la Web,
Departamento de Ciencias de la Computación
Universidad de Chile
bebustos@dcc.uchile.cl

Resumen Los modelos 3D son un tipo importante de dato multimedia, con una amplia gama de aplicaciones prácticas en áreas como la producción industrial, simulación, visualización y entretenimiento. La definición de similitud entre modelos 3D y la implementación de algoritmos de búsqueda por similitud son vitales para la puesta en marcha de bases de datos de objetos 3D. Sin embargo, esto conlleva al mismo tiempo problemas difíciles de resolver. En este artículo se presentan y discuten métodos para implementar la recuperación eficaz de modelos 3D en bases de datos multimedia.

1. Introducción

El desarrollo de métodos eficientes y eficaces de búsqueda para tipos de datos multimedia, como imágenes y video, se ha convertido en un tema de investigación importante debido a la creciente disponibilidad de información audiovisual. Un desarrollo similar se espera para los datos 3D, dado que los modelos 3D son un medio interesante para la difusión y el procesamiento de información en aplicaciones en las áreas de diseño industrial, simulación y entretenimiento, por dar algunos ejemplos. Todas estas aplicaciones tienen en común que las consultas realizadas a la base de datos no son búsquedas exactas (como en las bases de datos tradicionales), sino que son *búsquedas por similitud*, es decir, la consulta es un objeto 3D y uno desea recuperar todos los objetos 3D en la base de datos que sean *geométricamente similares* a la consulta. A éste tipo de búsqueda se le conoce también como *búsqueda por contenido*.



Benjamín Bustos es profesor Asistente del DCC. Doctor en Ciencias Naturales de la Universidad de Konstanz, Alemania (2006), Magíster en Computación (2002) e Ingeniero Civil en Computación de la Universidad de Chile (2001). Su principal línea de investigación es Bases de Datos Multimedia.

** Parcialmente financiado por el Nucleo Milenio Centro de Investigación de la Web, P04-067-F, Mideplan, y por el Proyecto FONDECYT 11070037.

Si bien existen muchas formas de definir y diseñar modelos 3D, una de las más comunes es a través de una *mall*a de triángulos, aunque también existen representaciones basadas en aproximaciones volumétricas o en nubes de puntos. Más información sobre formas de representar objetos 3D se puede encontrar en Campbell y Flynn [8]. Cualquiera de estas representaciones puede utilizarse para realizar búsquedas por similitud en bases de datos 3D. La Figura 1 muestra un ejemplo de un modelo 3D representado como una mall

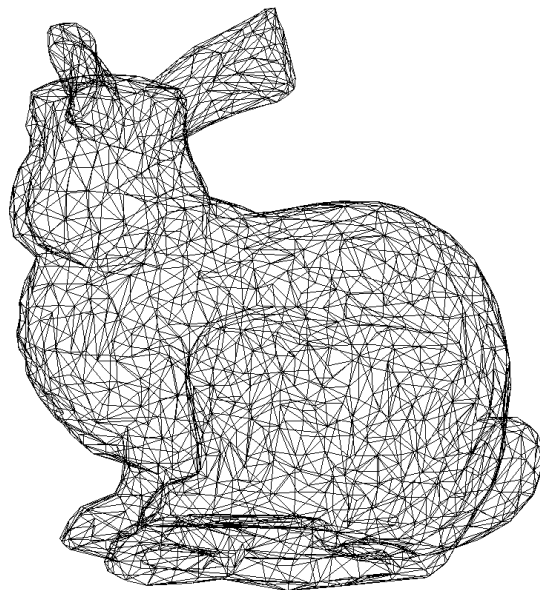


Figura 1. Mall

1.1. Aplicaciones

El problema de buscar objetos 3D similares tiene muchas aplicaciones prácticas. Algunos ejemplos son los siguientes:

- En medicina, las tomografías computacionales pueden ser utilizadas para detectar deformaciones de órganos similares a las almacenadas en bases de datos especializadas, lo cual puede ayudar a realizar diagnósticos médicos [9]
- La clasificación estructural es una de las operaciones básicas en biología molecular. Esta clasificación puede realizarse a través de búsquedas por similitud, donde las proteínas y moléculas son modeladas como objetos 3D [1].
- Algunos centros de reportes meteorológicos incluyen “pronósticos de polen” para prevenir y ayudar a las persona alérgicas a los diferentes tipos de polen. Ronneberger et al. [11] desarrollaron un sistema de reconocimiento de patrones que clasifica el polen a partir de información volumétrica (3D).

- Una base de datos de objetos 3D puede servir de apoyo a herramientas de tipo CAD (*Computer Aided Design*). Por ejemplo, partes estándar en una industria manufacturera pueden ser modeladas como objetos 3D. Al diseñar un nuevo producto, compuesto por muchas partes pequeñas, se puede intentar reemplazar algunas de sus partes por piezas estándar si éstas son similares, reduciendo así los costos de producción.

1.2. Búsquedas por similitud

El método estándar para realizar búsquedas por contenido en bases de datos multimedia está basado en el uso de los llamados “vectores característicos”. En este método se extraen atributos numéricos desde el objeto multimedia, los cuales describen características importantes del objeto. Estos atributos numéricos se utilizan para construir un vector, el *vector característico* del objeto, el cual es una representación del objeto multimedia original. Estos vectores característicos poseen habitualmente una alta dimensionalidad (del orden de cientos de dimensiones). De esta forma, el problema de comparar objetos multimedia complejos se reduce a buscar puntos cercanos en un espacio vectorial. Como medida de cercanía (similitud) de los vectores característicos se pueden utilizar diversas funciones de distancia, como por ejemplo la distancia Euclidiana o la distancia Manhattan. Es decir, mientras más cercanos estén dos vectores en el espacio vectorial, más similares son.

Hay dos consultas típicas de búsqueda por similitud. Por una parte, la *consulta por rango* retorna todos aquellos objetos de la base de datos que se encuentren a lo más a una cierta distancia ε (definida por el usuario) del objeto de consulta. Por otra parte, la *consulta por k -vecinos más cercanos* retorna los k objetos de la base de datos más cercanos al objeto de consulta.

2. Extracción de características desde objetos 3D

Existen muchos métodos distintos para describir objetos 3D como vectores característicos [5]. En esta sección se revisarán requisitos básicos para todos los descriptores 3D, así como un modelo general de extracción de características.

2.1. Requerimientos de invarianza

Considerando el método basado en descriptores, se pueden definir varios requerimientos que estos descriptores deben cumplir. Un buen descriptor 3D debe ser *invariante* a cambios en la orientación (traslación, rotación y reflexión) y en la escala del modelo 3D. Esto significa que el sistema de búsqueda debe ser capaz de recuperar objetos 3D geoméricamente similares con orientaciones y tamaños diferentes. Además, un buen descriptor 3D debe ser *robusto* con respecto a pequeños cambios en el nivel de detalle, geometría y topología de los modelos, es decir, el vector característico de un modelo 3D no debiera variar mucho frente a pequeños cambios en el modelo original.

Las propiedades de invarianza y robustez se pueden obtener en forma implícita por descriptores que consideran propiedades relativas de los objetos 3D, por ejemplo, la distribución de la curvatura de la superficie del objeto. Para otros descriptores, estas propiedades se pueden obtener a través de un *proceso de normalización* que transforma los objetos de manera que queden representados en un sistema de referencia canónico, donde las distancias y las direcciones son comparables entre

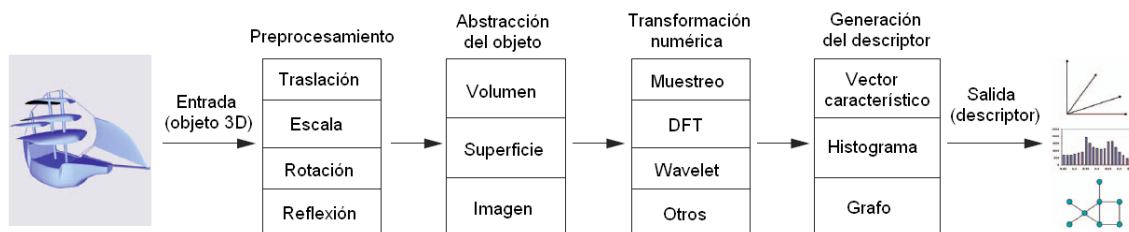


Figura 2. Modelo del proceso de extracción de descriptores 3D.

modelos 3D distintos. El método predominante para encontrar dicho sistema de referencia canónico está basado en el análisis de componentes principales (PCA) [10,12]. La idea de este método es alinear el objeto 3D considerando su centro de masa y sus ejes principales. El objeto se traslada en el espacio de forma que su centro de masa coincida con el origen del sistema de coordenadas (invarianza a traslaciones). Luego es rotado alrededor del origen de forma que los ejes x , y y z coincidan con las tres componentes principales del objeto (invarianza a rotaciones). La invarianza a reflexiones se puede obtener mediante un test basado en momentos, y la invarianza a escalamientos se puede obtener mediante el escalamiento del objeto por un factor canónico.

2.2. Modelo del proceso de extracción de características

Los distintos métodos de extracción de características para objetos 3D se pueden representar en un modelo general del proceso [6]. Este proceso se compone de varias etapas (ver Figura 2). Para un objeto 3D dado, usualmente representado como una malla de triángulos, se realiza primero un preprocesamiento (normalización) para obtener las propiedades de invarianza y robustez. Luego se realiza una abstracción del objeto 3D, de forma de caracterizarlo a partir de propiedades en su superficie, propiedades volumétricas o a partir de proyecciones en 2D (imágenes) del objeto. A continuación se puede realizar un análisis numérico de la forma 3D (por ejemplo un muestreo, o aplicar la transformada discreta de Fourier o la transformada Wavelet, etc.), y de este resultado se extrae finalmente el vector característico.

- i. *Preprocesamiento.* Si se requiere, se normaliza el objeto 3D para obtener invarianza a rotaciones, traslaciones, escalamientos y reflexión.
- ii. *Tipo de abstracción.* Hay tres tipos distintos: *superficie* (se miden características de la superficie del objeto), *volumen* (se miden características del volumen que ocupa el objeto 3D en el espacio), *imagen* (se toman proyecciones 2D del objeto en diferentes direcciones y se analizan posteriormente).
- iii. *Transformación numérica.* Las características principales de la malla de triángulos se pueden capturar numéricamente usando distintos métodos. Por ejemplo, grillas de celdas o conjuntos de imágenes se pueden transformar utilizando la transformada de Fourier, o se pueden realizar muestreos sobre la superficie del objeto.
- iv. *Generación del descriptor.* Puede ser de tres tipos: un *vector característico* (cuyas coordenadas la conforman los atributos numéricos extraídos del objeto), un *histograma* (que resume alguna característica medida del objeto) o un *grafo* (que puede representar la estructura del objeto 3D).

3. Calidad de la recuperación

En Bustos et al. [6] se describen los 16 descriptores 3D a evaluar en la comparación experimental.

3.1. Evaluación experimental

La base de datos utilizada para realizar los experimentos consiste en 1,838 objetos 3D recopilados de Internet¹. Una parte de este conjunto (472 objetos) fue clasificada manualmente por similitud geométrica en 55 clases distintas de modelos. El resto de los objetos fue considerado como “no clasificado”. Cada objeto perteneciente a una de estas 55 clases se utilizó en la evaluación experimental como objeto de consulta, y los objetos pertenecientes a su misma clase son los considerados como relevantes para la respuesta. Para comparar los distintos descriptores, se utilizaron diagramas de *precisión vs. recuperación* [2]. *Precisión* es la fracción de objetos recuperados que son relevantes a la consulta, y *recuperación* es la fracción del total de objetos relevantes recuperados. Todos los diagramas de precisión vs. recuperación se basan en los once niveles estándar de recuperación (0 %, 10 %, ..., 100 %), y se promediaron los resultados sobre todas las consultas en cada nivel de recuperación. Adicionalmente se utilizó la *R-precisión* (un valor escalar), que mide el valor de la precisión cuando el sistema retorna R objetos, donde R es el número de objetos *relevantes* para la consulta.

3.2. Comparación de la eficacia de distintos métodos

La Figura 3 (superior) muestra los resultados obtenidos en la evaluación experimental. El descriptor más eficaz (en promedio) es el denominado *Depth Buffer* (un descriptor basado en proyecciones 2D del objeto 3D), con una R -precisión promedio de 32 %. La diferencia de eficacia entre el mejor y el peor descriptor (*Depth Buffer* y *Shape Spectrum*, respectivamente) es significativa (factor 3×). Sin embargo, la diferencia entre los mejores descriptores es pequeña. Esto significa que en la práctica los mejores descriptores son, en promedio, similares en su eficacia.

La Figura 3 (inferior) muestra la influencia de la dimensión del vector característico en la calidad de la respuesta obtenida (medida como R -precisión). Se observa que la eficacia crece con la dimensión, pero la tasa de mejora disminuye rápidamente a partir de aproximadamente 64 dimensiones para casi todos los descriptores estudiados, alcanzándose un punto de saturación.

3.3. Análisis de los resultados

A partir de los resultados obtenidos, se puede concluir que los mejores descriptores 3D (en promedio) son aquellos basados en proyecciones del objeto original (e.g., *Depth Buffer*, *Silhouette*, *Rays-SH*). También obtuvieron buenos resultados algunos descriptores que extraen características volumétricas de los objetos 3D (e.g., *Voxel*, *3DDFT*). Los descriptores basados en características de la superficie del objeto 3D mostraron en general una eficacia baja. Todos los descriptores implementados mostraron ser robustos con respecto al nivel de detalle de los objetos.

Sin embargo, también se observó una varianza alta con respecto a la eficacia de los descriptores cuando se compararon los resultados entre distintas clases de objetos. Por ejemplo, para la clase “autos Fórmula 1” el mejor descriptor fue *Depth Buffer* (el mejor descriptor en promedio), mientras

¹ La base de datos está disponible en <http://merkur01.inf.uni-konstanz.de/CCCC/>.

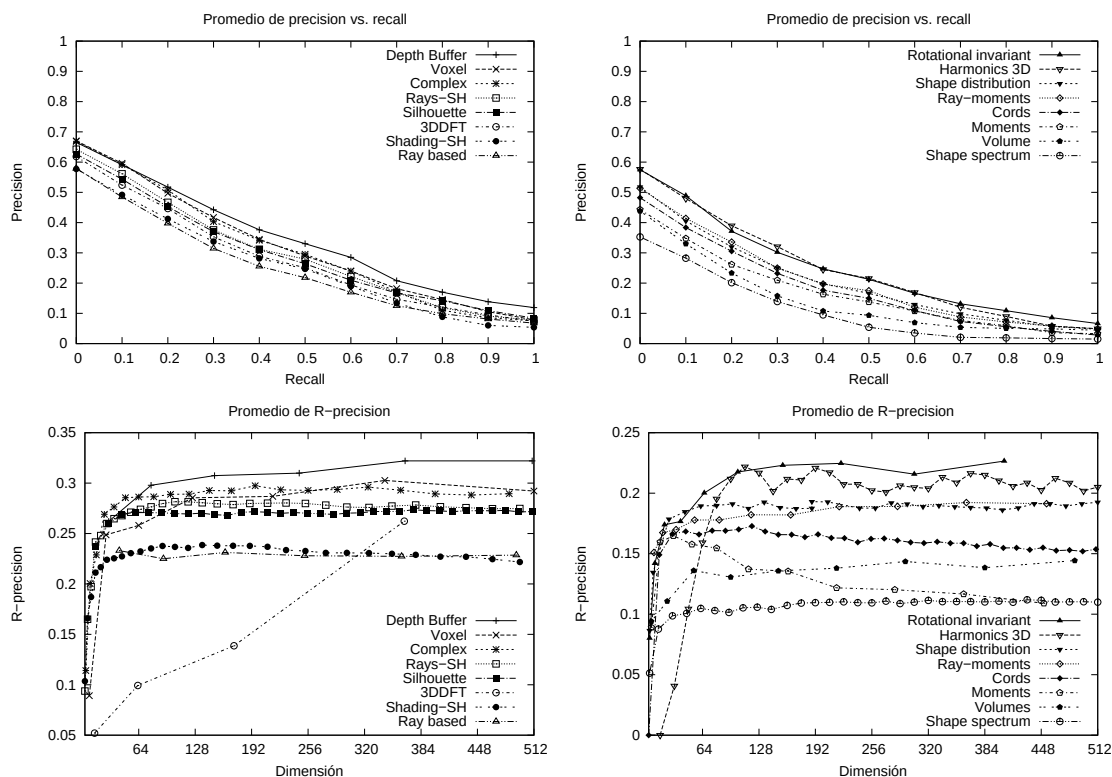


Figura 3. (Figuras superiores) *Precision vs. recall* para 16 descriptores 3D. (Figuras inferiores) *R-precision* en función de la dimensión para los mismos descriptores 3D.

que para la clase “animales marinos” el mejor descriptor fue *Silhouette*, y su eficacia fue casi el doble mejor que con *Depth Buffer*. Inclusive, para una clase muy particular, “figuras humanas”, el mejor descriptor fue *Shape Spectrum*, que resultó ser el peor en promedio.

Aparte de esta última notable excepción, no fue posible encontrar una correlación fuerte entre la geometría del objeto 3D y el mejor descriptor para dicha geometría. Por último, también se observó que para todos los descriptores estudiados se alcanza un punto de saturación a partir de una cierta dimensión, por lo que no es posible mejorar la eficacia de la búsqueda añadiendo más coordenadas a los vectores característicos una vez alcanzado este punto de saturación.

4. Combinando descriptores

Una forma de superar las limitaciones presentadas en la Sección 3.3 es utilizando combinaciones estáticas o dinámicas de descriptores 3D. La idea es combinar distintos descriptores para describir un objeto 3D, dado que los distintos descriptores capturan características distintas del objeto y por lo tanto la calidad de la respuesta podría mejorar al considerar toda esta información en conjunto al realizar una búsqueda. La Figura 4 muestra un ejemplo de tres búsquedas con el mismo objeto de

consulta (un modelo 3D de un auto Fórmula 1), utilizando primero dos descriptores por separado y luego utilizando una combinación simple de ellos. Las consultas con los descriptores utilizados independientemente retornan algunos objetos no relevantes (marcados en la figura), en cambio al combinar ambos descriptores sólo se obtienen objetos relevantes en la respuesta.

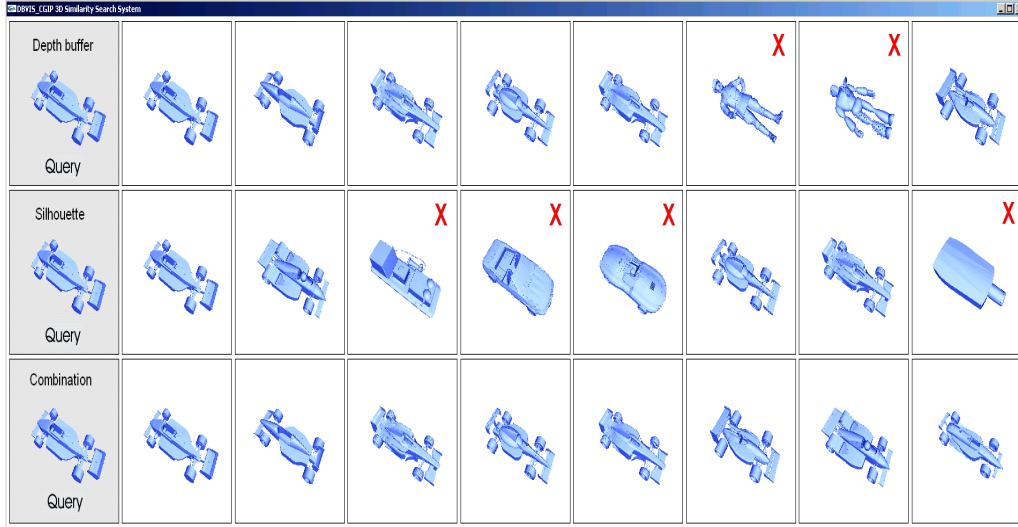


Figura 4. Comparación de dos búsquedas con descriptores independientes y su combinación estática.

4.1. Combinaciones estáticas

Una forma simple de implementar las combinaciones de descriptores 3D es simplemente sumando las distancias obtenidas con cada una de ellas, para luego utilizar este valor para generar el ranking final de la respuesta (éste fue el método utilizado para producir la Figura 4). Este método se denomina *combinación estática*, ya que se utiliza la misma combinación de descriptores para realizar todas las consultas.

Se probaron experimentalmente todas las posibles combinaciones estáticas de dos o más descriptores (hasta combinaciones de 10 descriptores), y el mejor resultado se obtuvo combinando los 5 mejores descriptores según los resultados presentados en la Figura 3. Con esta combinación estática se obtiene un aumento significativo en la calidad de las búsquedas por similitud. La Figura 5 muestra el diagrama de precisión vs. recuperación comparando el mejor descriptor en promedio con la mejor combinación de descriptores. La figura muestra que la combinación no solo aumenta el valor de la precisión para cada punto de recuperación, sino que también el valor de la R-precisión aumenta de 0,32 a 0,42, es decir, un aumento relativo de un 30 % (para dar una idea, aumentos relativos de la R-precisión de un 10 % son significativos).

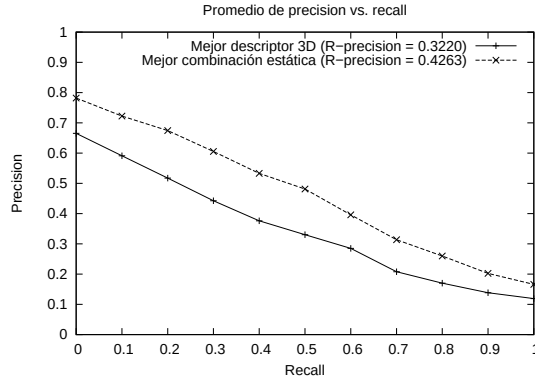


Figura 5. Comparación entre el mejor descriptor 3D (*Depth Buffer*) y la mejor combinación estática de descriptores.

4.2. Combinaciones dinámicas

Si bien se pueden obtener buenos resultados utilizando combinaciones estáticas de descriptores 3D, al analizar los resultados se llega a una conclusión similar que la obtenida con descriptores independientes: la mejor combinación a utilizar no siempre es la que funciona mejor en promedio, sino que nuevamente depende del objeto de consulta. El problema es que para algunos objetos de consulta un descriptor en particular puede ser muy útil, pero para otras consultas puede ser irrelevante.

Para resolver este problema, se propone el uso de *combinaciones dinámicas* de descriptores. En vez de fijar *a priori* la combinación a utilizar, ésta se elige dependiendo del objeto de consulta (que resulta de hacer una selección de descriptores a combinar). Alternativamente, se pueden utilizar todos los descriptores disponibles y lo que se asigna dinámicamente es el *peso* que tendrá cada descriptor en la suma de distancias. Si se decide que un descriptor es “malo” para una consulta dada, su peso debiera ser cercano a 0, en caso contrario, si es un “buen” descriptor para el objeto de consulta dado, su peso debiera ser cercano a 1.

El cálculo de los pesos se puede realizar utilizando una *base de datos de entrenamiento*. La hipótesis es que si la respuesta que retorna un descriptor en esta base de entrenamiento es *coherente* (i.e., los objetos retornados se parecen entre sí), entonces el descriptor se considera “bueno” para la consulta, y se considera “malo” en caso contrario. El método conocido como *entropy impurity* [4] está basado en esta hipótesis.

La figura 6 muestra los resultados obtenidos utilizando combinaciones dinámicas de descriptores, tanto con el método de selección como con el método *entropy impurity* (k es un parámetro del método). Los resultados muestran que el método más eficaz es el método *entropy impurity*, es decir, lo mejor es considerar todos los descriptores disponibles en el sistema de búsqueda, asignándoles su peso correspondiente al momento de realizar una consulta. Con este método, el valor de la R-precisión aumenta de 0,32 (mejor descriptor) a 0,46, es decir, un aumento relativo de un 43 %.

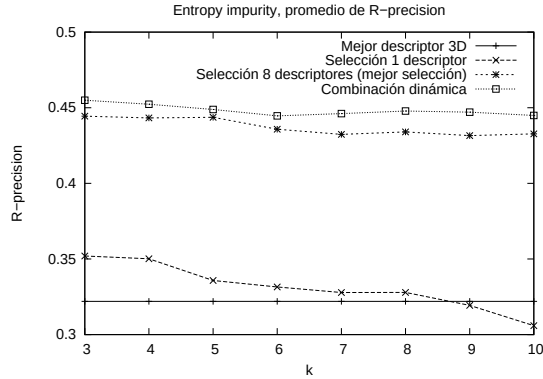


Figura 6. Comparación entre el mejor descriptor 3D (*Depth Buffer*), la mejor selección dinámica de descriptores y la mejor combinación dinámica de descriptores.

5. Problemas abiertos

Existen aún una serie de problemas abiertos relacionados con la recuperación de objetos 3D [3]. Los métodos actuales de búsqueda por similitud se enfocan principalmente en los aspectos geométricos de los modelos 3D, siendo ignorados otros atributos presentes en muchas bases de datos de objetos 3D, como por ejemplo el color, material y textura del objeto 3D. Otro problema abierto importante es el desarrollo de modelos para realizar búsquedas por similitud parcial en bases de datos de objetos 3D. En este tipo de búsqueda se desean recuperar los objetos de la base de datos que tengan alguna sección similar (y no necesariamente el modelo completo) al objeto de consulta, lo cual lo hace un problema mucho más complicado que el problema de búsqueda por similitud global descrito en este artículo. Por último, es tema de investigación actual desarrollar métodos eficientes de búsqueda por similitud, especialmente en el caso cuando se utilizan combinaciones dinámicas de descriptores [7].

Referencias

1. M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. Nearest neighbor classification in 3D protein databases. In *Proc. 7th International Conference on Intelligent Systems for Molecular Biology*, pages 34–43. AAAI Press, 1999.
2. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
3. B. Bustos, D. Fellner, S. Havemann, D. Keim, D. Saupe, and T. Schreck. Foundations of 3D digital libraries: Current approaches and urgent research challenges. In *Proc. 1st International Workshop on Digital Libraries Foundations (DLF'07)*, pages 7–12. DELOS Network of Excellence on Digital Libraries (www.delos.info), 2007.
4. B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranić. Using entropy impurity for improved 3D object similarity search. In *Proc. IEEE International Conference on Multimedia and Expo (ICME'04)*, pages 1303–1306. IEEE, 2004.
5. B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranić. Feature-based similarity search in 3D object databases. *ACM Computing Surveys*, 37(4):345–387, 2005.

6. B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranić. An experimental effectiveness comparison of methods for 3D similarity search. *International Journal on Digital Libraries, Special issue on Multimedia Contents and Management in Digital Libraries*, 6(1):39–54, 2006.
7. B. Bustos and T. Skopal. Dynamic similarity search in multi-metric spaces. In *Proc. 8th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'06)*, pages 137–146. ACM Press, 2006.
8. R. Campbell and P. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001.
9. D. Keim. Efficient geometry-based similarity search of 3D spatial databases. In *Proc. ACM International Conference on Management of Data (SIGMOD'99)*, pages 419–430. ACM Press, 1999.
10. E. Paquet, A. Murching, T. Naveen, A. Tabatabai, and M. Rioux. Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication*, 16:103–122, 2000.
11. O. Ronneberger, H. Burkhardt, and E. Schultz. General-purpose object recognition in 3D volume data sets using gray-scale invariants - classification of airborne pollen-grains recorded with a confocal laser scanning microscope. In *Proc. 16th International Conference on Pattern Recognition*, volume 2, pages 290–295. IEEE Computer Society, 2002.
12. D. Vranić, D. Saupe, and J. Richter. Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics. In *Proc. IEEE 4th Workshop on Multimedia Signal Processing*, pages 293–298, 2001.