CEIG - Spanish Computer Graphics Conference (2018)
I. García-Fernández and C. Ureña (Editors)

A prototype of virtual reality system for the visualization, exploration and modeling of huge point clouds

J. Ortega-Donaire[1], R. J. Segura-Sánchez[1], C.J. Ogayar-Anguita[1] and A. J. Rueda-Ruiz[1]

[1]Universidad de Jaén, CEATIC

Abstract

The use of specific techniques for the management and visualization of huge point clouds is necessary to solve the drawbacks of inefficiency derived from the size of the dataset and the techniques used to visualize it. This work presents a prototype of VR system for the visualization and management of extensive point clouds in 3D with the ability to edit specific points. For this, the tool incorporates multiresolution techniques, which improve the performance and efficiency of the system. The prototype also incorporates the management of the point cloud stored in an unstructured database; so the prototype can request parts of the dataset from the required fractions generated by an octree. This allows the progressive processing of 3D point clouds, which is very useful to control and visualize a large data set in real time.

CCS Concepts

•***Computing methodologies*** → *Virtual reality; Real-time simulation; Modeling methodologies;* •***Human-centered computing*** → *Information visualization;*

1. Introducción

Hoy en día, en campos como la industria, la topografía o la arquitectura, es esencial la utilización de un gran volumen de elementos geométricos muy precisos para el aseguramiento de la calidad de los proyectos. Para ello, la utilización de nubes de puntos permite plasmar de forma concisa y rigurosa modelos o entornos sin requerir una malla de triángulos 3D. La gestión y visualización de estas nubes de puntos exigen un tratamiento de datos y gestión de la memoria exigentes. Una forma de reducir el impacto que producen en el rendimiento y en la gestión de los recursos, consiste en la utilización de técnicas de multirresolución. En este sentido, la organización de la información en estructuras jerárquicas como octrees o kd-trees permiten discernir e identificar una agrupación de puntos y simplificar el tratamiento de los mismos [WRFH14]. Dada la simplicidad de su construcción, el uso de octrees es lo más frecuente.

La realidad virtual proporciona al usuario mecanismos de interacción mucho más potentes e intuitivos que los entornos de visualización 3D convencionales tales como Cloud Compare [Clo] o MeshLab [Lab]. Sin embargo, la utilización en el manejo de información científica, y en concreto de nubes de puntos, es escasa debido fundamentalmente al tamaño del dataset empleado en estos tipos de problemas, lo que impide que el propio sistema de realidad virtual funcione con los tiempos de latencia y respuesta requeridos para garantizar la inmersión e interacción.

En este trabajo se presenta un prototipo de sistema de realidad virtual basado en los dispositivos HTC Vive [Cor] que incluye funcionalidades básicas de interacción con la nube de puntos y gestión progresiva de la información. El resto del artículo se estructura como sigue: en primer lugar se presenta nuestra propuesta, que hace posible la representación de un prototipo para la visualización y gestión de extensas nubes de puntos 3D en realidad virtual, con capacidad de edición de puntos específicos. Posteriormente se discuten los resultados de la experimentación realizada, y por último se presentan las conclusiones obtenidas.

2. Nuestra propuesta

Actualmente, existe una amplia variedad de software capaz de visualizar y gestionar nubes de puntos. No obstante,la mayoría no posee la capacidad de editar puntos concretos, o son ineficientes a la hora de acceder a un elemento concreto en una nube de puntos de gran tamaño [KBK08]. Además, proporcionan métodos de interacción complejos o incómodos que implican cierta dificultad en el proceso de aprendizaje.

La mayor parte de las soluciones software existentes proporcionan entornos 3D interactivos convencionales orientados a mallas poligonales, o al tratamiento de nubes de puntos limitadas a la capacidad de la memoria disponible. Si se requiere procesar una cantidad de puntos mayor, suele ser habitual trabajar con herramientas out-of-core. Este tipo de soluciones suelen ser prototipos con poca o ninguna implantación, o bien sistemas propietarios, típicamente suministrados por las empresas

DOI: 10.2312/ceig.20181152

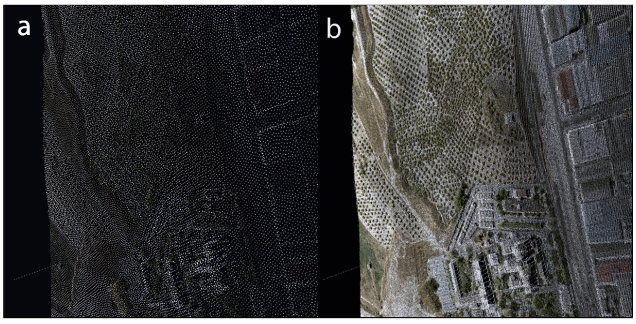

Figura 1: *Diferencia de nube de puntos dependiendo de la resolución: (a) Representación de la nube de puntos de un muestreo del 40% . (b) Representación de la nube de puntos de alta resolución. Nube de puntos resultante del proyecto PNOA [AVVA05]*

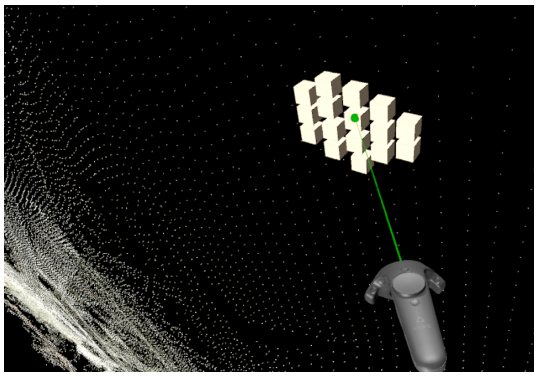

Figura 2: *Selección de puntos en la nube de puntos resultante del proyecto PNOA [AVVA05].*

fabricantes de escáneres de tiempo de vuelo. Así pues, nuestra propuesta está basada en la construcción de un sistema basado en el uso del dispositivo de realidad virtual HTC Vive, y la utilización del motor gráfico Unity [UT]. Dicho sistema permite visualizar una enorme nube de puntos almacenada en ficheros binarios (en formato LAZ), o en una base de datos (de tipo NoSQL en nuestro caso). Mediante la interacción con el dispositivo de realidad virtual, se puede editar las propiedades de la nube de puntos, incluyendo la selección y procesamiento de puntos específicos.

Para posibilitar el manejo de grandes nubes de puntos, el sistema ha diseñado técnicas de multirresolución con el fin de utilizar un procesamiento progresivo, en el que dependiendo de la posición que se encuentre el usuario en la escena 3D, visualiza con mayor o menor nivel de detalle la nube de puntos [ABCN10]. La distancia entre el usuario y una nube de puntos dada, es fraccionada en relación al nivel de detalle, formando así un conjunto de datos con baja resolución que permite ajustarse a la memoria de la GPU. De esta manera se evita la pérdida de información [DGBN*15], aunque visualmente motiva la reducción de información que se presenta al usuario a largas distancias de la nube de puntos. Este esquema es de especial importancia para garantizar una interactividad fluida con el dispositivo de entrada utilizado, además de compensar el modesto rendimiento del motor gráfico de Unity.

Como se ha mencionado anteriormente, manipular grandes volúmenes de puntos es un proceso complejo. Los archivos en formato binario son de gran ayuda, ya que permiten un acceso a los datos de forma eficiente. Además, la utilización del formato LAZ permite un ratio de compresión muy favorable, lo que disminuye las necesidades de ancho de banda durante las comunicaciones con el servidor que aloja los datos. Por otro lado, con una base de datos no estructurada, como es MongoDB [Mon], se puede almacenar eficientemente las nubes de puntos asociadas a cada nodo de la estructura jerárquica [Nal14], esto es, del octree. A medida que el usuario requiere una nube de puntos concreta, se realizan peticiones a la base de datos de MongoDB para obtener muestras [HW11] de una determinada resolución que contenga un mayor nivel de detalle o mayor cantidad de puntos. (Figura 1) Por contra, si el usuario se aleja de la nube de puntos en el entorno 3D, las muestras que son obtenidas de la base de datos contienen menor nivel de detalle (menor cantidad de puntos).

El esquema multirresolución se implementa sobre un octree que se almacena de forma implícita en la base de datos de MongoDB. Para cada nodo se genera un documento que contiene la información necesaria, incluyendo un fichero LAZ que contiene los puntos situados dentro de dicho nodo. La identificación del nodo se realiza mediante notación Base64, y contiene implícita la ruta desde el nodo raíz. De esta forma los nodos pueden buscarse y accederse de forma directa e independiente. Mediante el uso del octree se asegura una cantidad aceptable de celdas, lo cual es beneficioso para extensas nube de puntos, pero al mismo tiempo puede llegar a ser desfavorable al contener numerosas celdas pequeñas, lo que provoca un mayor número de peticiones a la base de datos cuando el usuario necesita los mayores niveles de detalle. La celdas generadas por el octree son de gran ayuda a la hora de visualizar una zona determinada a máximo nivel de detalle sin requerir la información de otras zonas de la nube de puntos. Al visitar una parte de la nube de puntos se genera únicamente dicha parte y se descarta los puntos restantes, lo que causa una mejora de rendimiento notable en la memoria del sistema, siendo menor que tener que procesar la nube de puntos completamente. En el caso de que se desee visitar otras celdas adyacentes, se extenderán los puntos, completando más aún la nube de puntos resultante.

Por otro lado, el uso de la realidad virtual para visualizar grandes nubes de puntos 3D en tiempo real, hace que se requiera una considerable cantidad de recursos [Sch16]. Para facilitar y mejorar la eficiencia a la hora de visualizar los datos se emplean las técnicas mencionadas de multirresolución, que además sirven para que el usuario no sufra tantos mareos debido a la escasez de fotogramas por segundo, ni que se produzcan grandes tiempos de carga en el procesamiento de la nube de puntos. El sistema de realidad virtual permite una inmersión en el entorno virtual donde suele ser más sencillo identificar los puntos en el espacio tridimensional. Además, el usuario se puede sentir más cómodo al seleccionar los puntos para la edición. Para ello se utiliza un cubo, que al coincidir con cada uno de los puntos o grupo de puntos, toma el color de los mismos. (Figura 2)

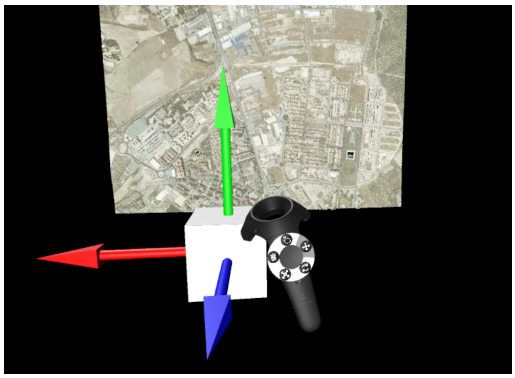

Figura 3: *Selección de transformación geométrica con el menú radial.*

2.1. Interacción con la nube de puntos

Los puntos seleccionados por el usuario se identifican dependiendo de las celdas del octree en el nivel de detalle más profundo. Una vez identificados, el usuario tiene la posibilidad de editarlos. Una de las opciones más comunes es el marcado para eliminación, una opción imprescindible en el procesamiento de nubes de puntos procedentes de escaneado 3D, donde siempre existen puntos erróneos en zonas ilógicas de la escena o que no interesan para el uso al que se destinarán los datos.

Además de lo anterior, es conveniente que una nube de puntos sea capaz de realizar transformaciones geométricas como traslación, rotación o escalado. Para ello, se muestra el sistema de coordenadas tridimensionales como si de un software de edición habitual se tratase. El usuario puede seleccionar la coordenada dominante hacia donde desea mover, alrededor de la que quiere girar, o sobre la que desea escalar la nube de puntos 3D. Mediante la interacción de un menú radial en el mando de HTC Vive, se puede seleccionar fácilmente la transformación geométrica que se quiere realizar. (Figura 3) Además, al seleccionarse una determinada transformación, cambia también la representación del sistema de coordenadas.

3. Experimentación y pruebas

Para la realización de las distintas pruebas del estudio sobre rendimiento y usabilidad, se ha utilizado un equipo basado en Intel Core i7-7700 3.60GHz, con 16B RAM y un tarjeta gráfica NVIDIA GeForce GTX 1060 de 3GB. Como dispositivo de realidad virtual se ha utilizado la mencionada HTC Vive. El servidor de base de datos MongoDB ha estado accesible a través de una red gigabit estándar. Con esta configuración se han obtenido los resultados mostrados en la siguiente tabla, (1), que muestra cómo varía el nivel de detalle relacionado con el mínimo espacio entre los puntos, la cantidad de puntos totales de la nube, el tamaño que ocupan en memoria y el rendimiento en el sistema.

La sucesión entre resoluciones de la nube de puntos es dinámica, la cual realiza una precarga de los puntos antes de ingresar en un nuevo nivel de detalle, para posteriormente mostrar dichos puntos. El sistema progresivo resulta útil como intento de eludir los tiempos

Nube de puntos		Puntos	Memoria	Rendimiento
(1)	Original	3.345.318	21.656 KB	92.8 FPS
	Downsample 1	1.632.499	12.064 KB	118.3 FPS
	Downsample 2	742.635	5.968 KB	125.5 FPS
	Fracción	136.443	1.056 KB	130.5 FPS
(2)	Original	81.147.343	874.802 KB	3.2 FPS
	Downsample 1	937.490	7.830 KB	122.7 FPS
	Downsample 2	239.559	2.108 KB	128.1 FPS
	Fracción	9.866.197	68.927 KB	73 FPS

Table 1: *Análisis del rendimiento utilizando diferentes nubes de puntos. (1) Nube de puntos resultante del proyecto PNOA [AVVA05]. (2) Nube de puntos LiDAR UAV Malibú (California).*

de espera en la carga y visualización de puntos entre resoluciones. En la tabla 2 (2), se detalla la variación de tiempo entre el sistema de progresión estático y dinámico. En el sistema de progresión estático no existe una precarga de puntos previa, sino que se realiza una carga y visualización simultánea al ingresar en un nuevo nivel de detalle.

Nube de puntos		Estático	Dinámico
(1)	Original	4,45 s	1,21 s
	Downsample 1	2,75 s	0,64 s
	Downsample 2	2,33 s	0,52 s
	Fracción	1,64 s	0,41 s
(2)	Original	27,87 s	6,98 s
	Downsample 1	2,43 s	0,62 s
	Downsample 2	2,19 s	0,52 s
	Fracción	8,31 s	2,01 s

Table 2: *Comparación de tiempo entre el sistema de progresión estático y dinámico. (1) Nube de puntos resultante del proyecto PNOA [AVVA05]. (2) Nube de puntos LiDAR UAV Malibú (California).*

Para obtener una latencia menor en el sistema de progresión dinámico, éste evalúa y calcula lo necesario cuando el sistema debe precargar los puntos antes de la visualización. Sabiendo la máxima velocidad a la que puede desplazarse el usuario (1,72 unidades por segundo en el estudio) y teniendo una trayectoria rectilínea hacia la nube de puntos, se emplea la siguiente fórmula:

$$d_\alpha = d_\beta + \frac{t_0}{t_1}$$

Siendo t_0 el tiempo de anticipación en segundos; d_α la distancia entre la carga y visualización inicial; d_β la nueva distancia entre la carga y la visualización con t_0 segundos de anticipación; t_1 el tiempo que emplea el usuario en desplazarse una unidad.

Por ejemplo, para anticiparnos 0,65 segundos y obtener una duración menor en el sistema de progresión dinámico, necesitaremos añadir 11,12 unidades más de distancia. (3)

$$d_\alpha = 10 + \frac{0,65}{0,58} = 11,12 unidades$$

Para el estudio de usabilidad, se ha realizado una selección de 5

Nube de puntos		Dinámico
(1)	Original	0,56 s
	Downsample 1	0 s
	Downsample 2	0 s
	Fracción	0 s
(2)	Original	6,33 s
	Downsample 1	0 s
	Downsample 2	0 s
	Fracción	1,36 s

Table 3: *Tiempos de espera entre la carga y visualización de puntos utilizando el sistema de progresión dinámico con una distancia de 11,12 unidades. (1) Nube de puntos resultante del proyecto PNOA [AVVA05]. (2) Nube de puntos LiDAR UAV Malibú (California).*

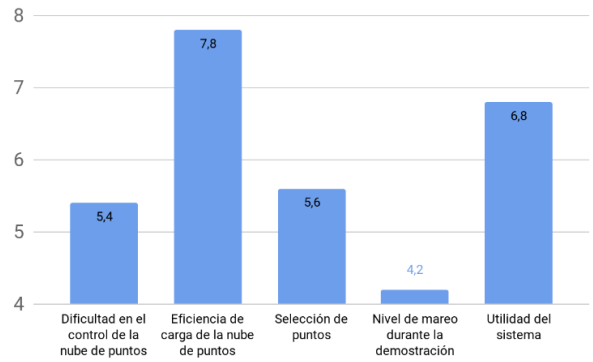

Figura 4: *Valoración media de 5 encuestados de una demostración sobre la interacción con el sistema.*

usuarios familiarizados con el empleo de herramientas habituales para procesamiento de nubes de puntos. Se ha realizado una prueba de demostración para justificar la bondad de la interacción con el sistema. La encuesta consta de una valoración (escala 1-10) acerca de 5 aspectos fundamentales de dicha prueba del sistema: (Figura 4)

- Nivel de dificultad en el control de la nube de puntos.
- Eficiencia de carga de la nube de puntos.
- Selección de puntos con el mando.
- Nivel de mareo durante la demostración.
- Utilidad del sistema.

4. Conclusiones y trabajo futuro

La edición de puntos en realidad virtual permite que el usuario se sumerja en un entorno donde puede visualizar e interactuar mejor con los datos en el espacio tridimensional. Esta capacidad es especialmente útil cuando se tiene una gran cantidad de puntos concentrados en una zona, donde las soluciones actuales no dan resultados tan buenos como con el sistema desarrollado. Las técnicas empleadas en este trabajo hacen posible el tratamiento ubicuo y transparente de grandes nubes de puntos con hardware convencional, especialmente en cuanto a las limitaciones de memoria se refiere. También se consigue un buen rendimiento en el sistema de realidad virtual en sí, sin percibir grandes variaciones de fotogramas por segundo, manteniendo estable la tasa de refresco de imagen y por tanto reduciendo la aparición de mareos.

Con la aplicación de técnicas de multirresolución, el sistema de progresión dinámica y el uso de una base de datos NoSQL como MongoDB, se consigue una mejora notable en el rendimiento y se facilita la visualización de la nube de puntos en tiempo real. Para mejorar el tiempo de espera entre la carga de los puntos y la visualización de los mismos, se debe modificar la distancia entre el punto de carga y en el de visualización. La distancia depende de la complejidad de los datos, del hardware del equipo donde se ejecuta, del rendimiento de servidor, así como del ancho de banda y la latencia de la red.

Los encuestados que justifican la bondad de la interacción con el sistema consideran principalmente que no se producen mareos, y también observan una diferencia notable en la carga y visualización de puntos en comparación con otros programas utilizados.

Referencias

[ABCN10] ANDUJAR C., BRUNET P., CHICA A., NAVAZO I.: Visualization of large-scale urban models through multi-level relief impostors. *Computer Graphics Forum 29*, 8 (nov 2010), 2456–2468. URL: https://doi.org/10.1111/j.1467-8659.2010.01757.x, doi:10.1111/j.1467-8659.2010.01757.x. 2

[AVVA05] AROZARENA-VILLAR A., VILLA-ALCÁZAR G.: Plan nacional de ortofotografía aérea de españa (pnoa). *Topografía y cartografía: Revista del Ilustre Colegio Oficial de Ingenieros Técnicos en Topografía 22*, 127 (2005), 30–41. 2, 3, 4

[Clo] CLOUDCOMPARE: 3d point cloud and mesh processing software. URL: http://www.cloudcompare.org/. 1

[Cor] CORPORATION H.: Vive virtual reality. URL: https://www.vive.com/. 1

[DGBN*15] DÍAZ-GARCÍA J., BRUNET P., NAVAZO I., PÉREZ F., VÁZQUEZ P.-P.: Feature-preserving downsampling for medical images. In *EuroVis* (2015). 2

[HW11] HONGCHAO M., WANG Z.: Distributed data organization and parallel data retrieval methods for huge laser scanner point clouds. *Computers & Geosciences 37*, 2 (feb 2011), 193–201. URL: https://doi.org/10.1016/j.cageo.2010.05.017, doi:10.1016/j.cageo.2010.05.017. 2

[KBK08] KREYLOS O., BAWDEN G. W., KELLOGG L. H.: Immersive visualization and analysis of LiDAR data. In *Advances in Visual Computing*. Springer Berlin Heidelberg, 2008, pp. 846–855. URL: https://doi.org/10.1007/978-3-540-89639-5_81, doi:10.1007/978-3-540-89639-5_81. 1

[Lab] LABORATORY V. C.: Meshlab. URL: https://www.meshlab.net/. 1

[Mon] MONGODB I.: Mongodb - document databases. URL: https://www.mongodb.com/. 2

[Nal14] NALE A.: Design and development of a generalized lidar point cloud streaming framework over the web. In *Università degli Studi di Padova, Italy* (2014). 2

[Sch16] SCHUETZ M.: Massive time-lapse point cloud rendering in virtual reality. NVIDIA SIGGRAPH 2016. 2

[UT] UNITY-TECHNOLOGIES: Unity - Game Engine. URL: https://unity3d.com/. 2

[WRFH14] WENZEL K., ROTHERMEL M., FRITSCH D., HAALA N.: An out-of-core octree for massive point cloud processing. In *Workshop on processing large geospatial data Cardiff* (2014). 1