

Editor de escenarios para aplicaciones de Realidad Virtual

Marcelo J. Vénere, M. Virginia Cifuentes, Juan Pablo D'Amato y Cristian García Bauza
PLADEMA-ISISTAN, {venerem,cifuentes,jpdamato,crgarcia}@exa.unicen.edu.ar

Resumen— En el presente trabajo se describen los desarrollos recientemente realizados en la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNICEN) para construir escenarios tridimensionales a ser utilizados en sistemas de simulación y entrenamiento de operarios utilizando técnicas de realidad virtual. El trabajo propuesto estructura un editor en una jerarquía de módulos, utilizando georeferenciación para todos los objetos 3D colocados en los escenarios, permitiendo además la personalización en la simulación de clima, definición de objetos estáticos y móviles, incorporación de objetos con comportamiento complejo que pueden requerir de simulaciones (como agua marítima, fuego o humo por ejemplo), etc. Los escenarios creados son luego utilizados en módulos visuales de simuladores existentes logrando representaciones realistas en tiempos aceptables.

Palabras clave— Realidad Virtual, Simulación Computacional, Algoritmos Geométricos, Computación Gráfica.

I. INTRODUCCIÓN

Después de tres décadas de crecimiento exponencial en la performance de las computadoras, no debería asombrarnos que una simple PC permita hoy realizar simulaciones que estaban restringidas a super-computadoras solo diez años atrás. Sin embargo no deja de ser impactante el cambio observado en los últimos tres años en materia de computación gráfica (sin duda impulsado por la industria de los juegos). Hoy una placa gráfica de 150u\$s permite renderizar cientos de millones de polígonos por segundo, implementando completamente en hardware a lenguajes como OpenGL. Esta situación nos pone en un momento excitante en que un mundo de aplicaciones de computación gráfica y realidad virtual esperan a ser desarrolladas. Un ejemplo interesante son las aplicaciones en medicina (ver [1] por ejemplo) que están yendo más allá del diagnóstico por imágenes para comenzar con operaciones en base a imágenes virtuales.

Los sistemas de realidad virtual son sin duda una herramienta de importancia para el entrenamiento de operarios, llegando en algunos casos a formar parte obligatoria de los cursos de formación. Por ejemplo, un piloto de aeronaves debe primero cumplir un cierto número de horas en un simulador antes de obtener su habilitación. La implementación de estos sistemas requiere del trabajo multidisciplinario de investigadores en computación gráfica, simulación computacional y hardware, como así también con los usuarios y operarios con experiencia en el área particular. Es sin duda por ello que son muy pocas las empresas en el mundo que ofrecen estos productos con la calidad requerida para formar y licenciar operarios en áreas estratégicas.

Particularmente en Argentina se está desarrollando desde el año 2001 un sistema llamado MELIPAL orientado a la simulación de embarcaciones, que pretende competir con

los mejores productos del área. Ya se encuentran en operaciones varios de sus módulos, por ejemplo, MELIPAL-R, presentado en [2], que contempla la consola de comandos del barco, un módulo para el manejo de cartografía y otro para radares-Arpa con toda su funcionalidad. El sistema consta de cinco puestos de entrenamiento (cada uno con tres computadoras) y un puesto para el instructor que monitorea el ejercicio y plantea situaciones a los alumnos.

Este sistema no contemplaba aun la visualización de escenarios tridimensionales, pero el desarrollo del simulador de radar requirió de elementos de geometría computacional no triviales (presentados en [3]). La primera aplicación realizada por nuestro equipo para el entrenamiento mediante la visualización de escenarios fue el sistema SIPER 2, presentado en [4], y consistió en la simulación del periscopio de un submarino utilizando lentes de realidad virtual.

El trabajo presentado en [5] marca la continuidad en el desarrollo de sistemas de simulación de entrenamiento. Dicho trabajo, denominado MELIPAL-P es un simulador de embarcación pesquera que permite evaluar a los alumnos en distintas condiciones ambientales y de operación. Para ello fue preciso modelar distintos instrumentos, como sonar de exploración y ecosondas, la visual a través de las ventanas e incluso una visión del fondo marino. Actualmente se encuentra funcional e instalado en la Escuela Nacional de Pesca, en la ciudad de Mar del Plata.

El caso que nos interesa en este trabajo es el desarrollo de un editor de escenarios para dar soporte a los módulos visuales de los sistemas de entrenamiento anteriormente nombrados. Para ello fue preciso diseñar distintos módulos de edición, los cuales van desde un editor de posición georeferencial hasta uno de simulación de clima para los escenarios generados. A continuación se describe la forma en que cada uno de los editores se desarrolló y se integró a la solución final, junto con las técnicas utilizadas para la visualización.

II. CARACTERÍSTICAS DEL SISTEMA

MELIPAL ha sido desarrollado como una arquitectura distribuida en una LAN, donde cada puesto de entrenamiento puede requerir de varias computadoras y la coordinación de todos los procesos se realiza desde la computadora asociada al puesto del instructor, utilizando para ello una arquitectura Cliente-Servidor.

El módulo instructor es el encargado de la creación y administración del ejercicio y los datos de entrada para cada uno de los puestos de entrenamiento. El envío de mensajes se encuentra dividido en dos grandes grupos: mensajes de control y mensajes de datos. El primer grupo, sirve para que todos los puestos de entrenamiento

sincronicen sus valores de entrada, y se mantenga una consistencia entre los diferentes módulos que integran un puesto de entrenamiento. En cambio, los mensajes de datos, se envían regularmente (cada 1.0 segundo) para mantener actualizada la información del ejercicio, como posiciones de barcos, cardúmenes, etc. Estos mensajes de datos son utilizados por los diferentes módulos de simulación incluidos en los puestos de entrenamiento. Al modificarse los datos iniciales del ejercicio constantemente por parte de los puestos de entrenamiento, el envío de mensajes debe ser bi-direccional. Así, cada módulo reenvía la información modificada hacia el puesto Instructor, para que éste tenga un seguimiento exacto de las acciones tomadas por el alumno en su puesto.

En particular, para los módulos visuales del sistema, encargados de la visualización de escenarios 3D mostrando una perspectiva desde el interior del barco, se propuso en [5] utilizar un repositorio de objetos comunes para acceder a las mismas variables y reducir el envío de mensajes en la red. Si bien se logró esta reducción, existían problemas en el manejo y control de la gran cantidad de variables para un escenario en particular. La generación en estos casos insumía grandes costos de tiempo y no se contaba con herramientas de edición adecuadas para lograr los escenarios 3D de los diferentes ejercicios con los que se entrenaban a los operarios.

Para reducir el costo de mantener dicha estructura de variables y optimizar los tiempos en la creación de escenarios ad-hoc, se diseñó el editor MELIPAL-ED presentado en este trabajo.

La interacción con los sistemas MELIPAL existentes se logra mediante los módulos visualizadores, los cuales están siendo actualizados para posibilitar el uso de los nuevos escenarios creados con nuestro trabajo usando la tecnología de un motor gráfico desarrollado en la UNICEN llamado IMPROMPTU, el cual fue ya presentado en [5].

III. FASES DE EDICION

El proceso de creación consiste de diferentes pasos y herramientas que conforman un *pipeline* de edición como el mostrado en la Fig. 1.

En un primer paso, se genera un mapa georeferenciado basándose en una imagen *raster* de una región en particular (también podría ser una imagen vectorial, pero típicamente es un mapa cartográfico digitalizado). Sobre esta imagen se desarrollará todo el trabajo de posicionamiento de objetos. En segundo lugar se incorpora un modelo de elevación digital (MDE) de la misma región, el cual también es necesario geo-referenciarlo, de forma de “encajarlo” precisamente con el mapa de base.

A continuación es posible comenzar a incorporar objetos al escenario ubicándolos espacialmente, tarea que se realiza en dos dimensiones sobre la imagen *raster*. La tercera dimensión se obtiene automáticamente desde el DEM incorporado, simplificando así la tarea del usuario. Si el objeto en particular no se encuentra sobre el suelo, entonces la tercera dimensión podrá modificarse con posterioridad. Dado que la imagen *raster* ya está geo-referenciada, los objetos nuevos incorporados quedarán también geo-referenciados en este proceso y por lo tanto podrán exportarse con esta información. De esta forma es posible trabajar con distintas imágenes *raster* de base para ubicar los objetos, lo cual facilita la definición con mayor detalle de determinadas regiones. Por ejemplo en el caso de una ciudad ubicada en una zona mucho más extensa

podremos utilizar una imagen del diagrama de calles para colocar las construcciones y otra imagen con la cartografía de la zona para ubicar los bosques y elementos de los alrededores.

Los objetos insertados en el mapa de edición poseen distintas propiedades, pudiéndose distinguir cuatro clases principales:

1. Objetos Estáticos
2. Objetos Dinámicos (con trayectorias predefinidas)
3. Objetos con comportamiento complejo
4. Zonas de objetos iguales

Cada uno de ellos se crea y maneja con un módulo específico.

Una vez que se añaden estos objetos, se dispone de un módulo adicional que permite la configuración en cuanto a la vista de la escena, teniendo en cuenta efectos de clima, estaciones y horarios del día.

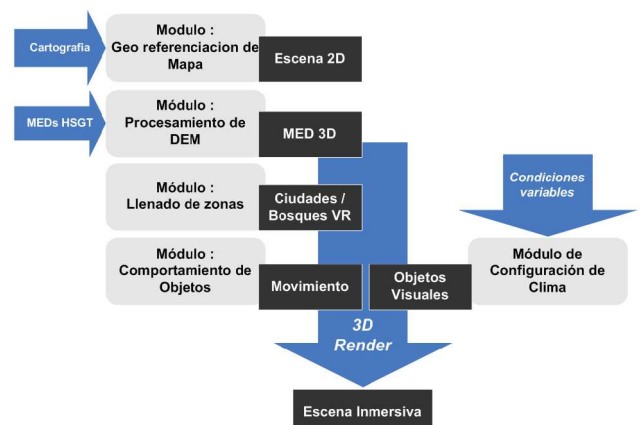


Fig. 1: Fases de edición en el sistema MELIPAL Ed

IV. MODELOS DE ELEVACION

Básicamente un MDE está constituido por una grilla georeferenciada donde cada elemento (o celda) tiene un valor de cota o altura. A su vez, cada elemento de la grilla lleva asociado un par de coordenadas que se corresponde con un sistema de representación terrestre (p.e. geodésicas o Gauss Kruger), en este caso, la ubicación de objetos en el editor sobre un determinado mapa utiliza el mismo sistema de representación terrestre que el MDE utilizado. De esta manera, el MDE constituye una representación más o menos fiel de una determinada porción de la superficie terrestre.

Obviamente, mientras más densa sea esta grilla, mayor nivel de detalle estará representado mediante el MDE, siendo normal hoy utilizar modelos con decenas o incluso centenas de millones de celdas. Este volumen de información puede ser necesario para muchas aplicaciones, pero representa un serio inconveniente para su visualización interactiva y más aún si se desea “navegar” en tiempo real esta topografía como se describe en [6], [7].

Existen en MELIPAL un conjunto de mapas que componen el paquete del sistema. Estos mapas difieren en el tamaño, con lo cual, la inclusión de la nueva técnica de utilización de MDE para los escenarios virtuales generados debía permitir:

- Fácil distribución
- Creación dinámica

Para soportar distintos modelos de elevación, se creó una jerarquía de clases que permite extensibilidad y modificabilidad en la utilización de formatos. Actualmente los MDE son obtenidos de una base de datos del Programa

de colaboración USGS EROS/NASA SRTM (*Shuttle Radar Topography Mission*).

Básicamente son archivos binarios, con valores de alturas, representando zonas de 1 grado por 1 grado de longitud. La resolución de estos MDE es de 90 metros por píxel. No contienen cabecera, ni archivos adicionales de configuración

Como mencionamos anteriormente, existe un conjunto de mapas que conforman las zonas de entrenamiento en el sistema, los cuales son de distinto tamaño. Al utilizar MDE de 1 grado por 1 grado, surgió la necesidad de unir MDEs de regiones para formar los MDE finales que acompañen a los mapas existentes y a los nuevos mapas creados de aquí en más. Este problema se solucionó realizando una búsqueda local entre los archivos de los SRTM que concuerden con los límites de la zona editada, posteriormente se forma el MDE definitivo en base a uniones de sus valores y se recortan los bordes. Existe una fase posterior donde se aplican filtros para eliminar ruidos generados en los MDE por la refracción emitida por zonas de agua.

El módulo de procesamiento de MDE permite también asignar texturas al modelo. Es posible asignar texturas por zonas, por ejemplo zonas de nieve y valle, y también se permite utilizar multitexturas usando una textura base y una textura de detalle.

En este punto del trabajo, para la posterior representación del MDE en la escena se aplica un algoritmo que permite reducir la cantidad de polígonos utilizados para representar superficies extensas, como la de los modelos topográficos usados, sin perder calidad en la imagen generada. La malla de triángulos resultante se construye aplicando un proceso de descomposiciones sucesivas al espacio bidimensional que define el campo escalar de alturas, pudiéndose incluso obtener diversas aproximaciones de un modelo variando la cota de error condicionante del particionamiento de una región. Se utiliza la curvatura local como indicador del error y una representación jerárquica basada en un quadtree con restricciones y templates en cada nodo terminal del árbol para generar la triangulación final de la superficie. De esta forma los lugares con baja curvatura pueden ser simplificados sin perder calidad, lo cual en general permite reducir la cantidad de polígonos en un factor diez y este factor es aún mayor si se tolera una cierta pérdida como se detalla en [8].

V. TIPOS DE OBJETOS

Las escenas creadas por el editor, forman parte de un catálogo de escenarios virtuales que son utilizados en simuladores de entrenamiento de maniobra, pesca y radar. Es común que en situaciones reales los operarios de embarcaciones utilicen objetos de la costa destacables desde su posición, como puede ser un bosque, un faro, o un edificio. Para este tipo de objetos (objetos estáticos representativos) es que el usuario del editor debe colocar la posición geo-referenciada exacta y ubicarlo en el mapa de edición.

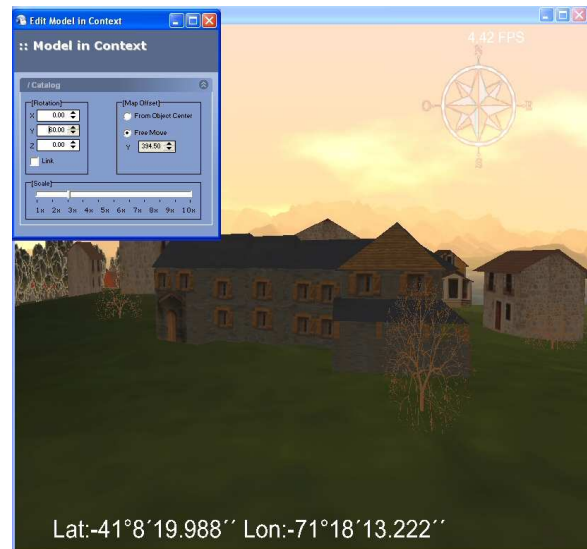


Fig. 2: El editor permite en tiempo real visualizar en la escena los cambios realizados en el modelo

El editor permite definir también objetos móviles con tres grados de libertad, los cuales en la escena generada serán movidos automáticamente por un módulo físico básico del motor gráfico. Así por ejemplo, se pueden incluir en la escena 3D, automóviles que recorren autopistas o bandadas de gaviotas que revolotean alrededor de un punto. En esta fase, se utilizan líneas rectas para indicar un camino de uno de estos objetos, interpolando linealmente entre cada punto intermedio (punto de quiebre de la línea) para lograr el movimiento deseado. Se incluye también un pequeño modelo de inercia para los recorridos de estos objetos para obtener movimientos suaves y sin saltos abruptos.

Existen objetos con un comportamiento más complejo, por ejemplo la representación de agua marítima o fuego, los cuales son configurados mediante módulos de edición propios, aunque vale aclarar que las características de implementación más importantes de estos objetos se encuentran en el motor gráfico que los representa y no en el editor.

Aparte de los tipos de objetos anteriormente nombrados existen lo que llamamos zonas de objetos iguales, las cuales son representadas en el editor como polígonos convexos, permitiendo determinar zonas del mapa de edición para que el usuario no tenga que definir los objetos uno por uno, sino sólo indicar una zona y el motor gráfico genera automáticamente dicha zona con modelos tridimensionales iguales (en realidad no son perfectamente iguales, sino que presentan características parecidas, por ejemplo son modelos de casas, o modelos de árboles, etc.). Es útil esta propiedad al momento de definir por ejemplo zonas de bosques (donde los objetos son todos homogéneos, es decir son todos árboles) o zonas de ciudades como barrios, etc. En un trabajo futuro puede incluirse la creación automática de los modelos 3D que conforman estas zonas homogéneas como se describe en [9].

Independientemente de su tipo, cada objeto tiene asociado un modelo 3D, el cual será renderizado en el visualizador por el motor gráfico. En tiempo de edición el usuario puede elegir un modelo 3D de un catálogo que se mantiene a partir de una base de datos. Se permite utilizar aparte del catálogo genérico un catálogo propio del usuario, así por ejemplo, se puede tener un catálogo de objetos propio para cada escenario.

Los objetos son colocados por el usuario seteando la posición georeferenciada y son ubicados en altura sobre el MDE de manera automática por el sistema. Una característica destacable es que varios de los parámetros (posición, rotación, escala, etc.) pueden ser cambiados en el objeto, utilizando la visión 3D, observando como el objeto quedará en la escena tridimensional terminada. (Ver Fig. 2).

VI. EDICIÓN DE CLIMA Y EFECTOS ESPECIALES

El editor que presentamos en este trabajo tiene un módulo especial de edición de condiciones climáticas, el cual entre otras cosas permite configurar parámetros de las fases del día, las cuales posteriormente son interpretadas por el motor gráfico encargado de visualizar los escenarios creados.

Entre los parámetros que son editables para cada fase (amanecer, mediodía, atardecer y noche) se encuentran la hora de comienzo y fin de la fase, degrade de los colores del cielo, intensidad de color y cantidad de nubes, niebla, entre otros.

Internamente en el motor gráfico existe un Administrador de efectos de clima, quien se encarga de interpolar los valores entre una fase y otra, permitiendo así que las escenas de la simulación varíen dinámicamente de acuerdo a los valores establecidos.

La edición del clima se realiza en tiempo real, en la misma escena que se está creando, con lo que se pueden visualizar los cambios en el instante. (Ver Fig. 3).

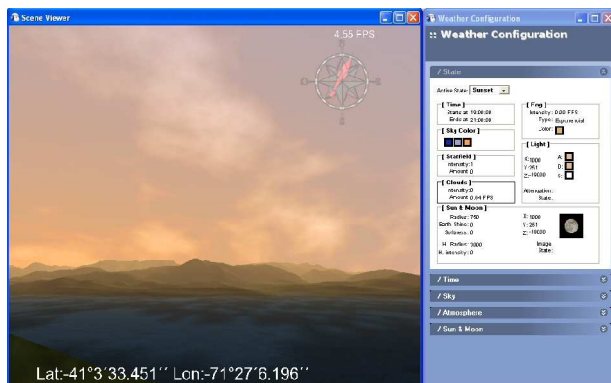


Fig. 3: Editor de efectos climáticos

VII. ESCENARIOS CONSTRUIDOS

Actualmente se está desarrollando el escenario de la ciudad de Bariloche. La creación está estructurada en fases y en grupos de personas. Las tareas más relevantes son obtener las posiciones georeferenciadas de los objetos estáticos representativos, determinar y construir el MDE a utilizar de la zona de interés, diseñar los modelos 3D a ser utilizados y configurar los parámetros para los módulos de llenado de zonas (en la creación de zonas de bosque y edificios) y simulación de clima. Una versión preliminar del escenario puede verse en la Fig. 4. En esta fase se ha trabajado con el MDE y la configuración de las fases del clima.



Fig. 4: Escenario de Bariloche en una primera fase de edición

VIII. CONCLUSIONES

Se consiguió desarrollar un editor de escenarios para ser utilizados en ambientes de realidad virtual. Los visualizadores existentes del sistema MELIPAL actualizados con dichos escenarios permiten entrenar navegantes de embarcaciones en condiciones inmersivas prácticamente idénticas a la realidad. Utilizando equipamiento de muy bajo costo y herramientas de software modernas se logró simular las condiciones climáticas de las escenas y se pudo crear escenarios con objetos localizados espacialmente de una manera muy similar a la real.

Estos desarrollos están siendo incorporados al sistema MELIPAL. El sistema MELIPAL-R se encuentra instalado en la Escuela Nacional de Náutica desde el año 2001, ha sido certificado por la OMI (Organización Marítima Internacional) y se utiliza regularmente como parte obligatoria de los cursos que allí se dictan. El módulo MELIPAL-P se encuentra instalado en la Escuela Nacional de Pesca, en la ciudad de Mar del Plata.

Actualmente se trabaja en la actualización de todo el sistema MELIPAL para permitir la utilización de las nuevas características incorporadas en los escenarios creados con el editor presentado en este trabajo.

REFERENCIAS

- [1] A.G. Gallangeher, "Virtual reality training in laparoscopic surgery: a preliminary assessment of minimally invasive surgical trainer virtual reality", en *(MIST VR), Endoscopy Mayo 1999*, pp. 310-313.
- [2] G. Boroni, M. Vénere, "Un simulador distribuido para entrenamiento de operarios", en *VIII Congreso Argentino de Ciencias de la Computación, CACIC 2002, Buenos Aires, 2002*.
- [3] G. Boroni, I. Otheguy, M. Solaro, M. Vénere, "Simulación en tiempo real de un radar de barrido horizontal", en *VIII South American Congress on Computational Mechanics, MECOM, Santa Fé, 2002*.
- [4] G. Boroni, M. Tossini, P. Vagliati, M. Vénere, "Simulación en tiempo real de un escenario marítimo mediante técnicas de realidad virtual", en *32° Jornadas Arg. de Informática e Investigación Operativa, JAIIO 2003, Buenos Aires, 2003*.
- [5] J. D' Amato, C. García Bauza, M. Vénere, "Simulación del entorno de una embarcación pesquera", en *33° Jornadas Argentinas de Informática e Investig. Operativa, JAIIO 2004, Córdoba, 2004*.
- [6] Faust, Hodges, Koller, Lindstorm, Ribarsky, Turner. Real-Time, Continuous Level of Detail Rendering of Height Fields, en *Proc. SIGGRAPH 96*, pp. 109-118, 1996.
- [7] Heidrich, Rottger, Slusallek, Seidel. "Real-Time Generation of Continuous Levels of Detail for Height Fields", en *Proc. WSCG '98*, pp. 315-322, 1998.
- [8] A. Clausse, V. Cifuentes, M. Vénere, "Un algoritmo para la simplificación poligonal de modelos topográficos digitales", en *33° Jornadas Arg. de Informática e Investig. Operativa, JAIIO 2004, Córdoba, 2004*.
- [9] P. Müller, Y. Parish, "Procedural Modeling of Cities", en *Proc. Of the 28th annual conf. on Computer graphics and interactive techniques*, pp: 301 - 308, 2001.