

Diseño e implantación de un sistema de inspección para la visualización de una bomba centrífuga en un ambiente virtual

Luis E. Vallés D, Antonio E. León S., Wasilio Koslow, Lucía Martino

Centro de Investigación de Materiales, Facultad de Ingeniería, Universidad de Carabobo, Valencia, Venezuela

Email: lapredim@uc.edu.ve

Resumen

Todo proceso industrial implica, por sí mismo, la utilización de equipos y herramientas para la consecución de sus objetivos. La utilización de los equipos y herramientas acarrea un proceso de deterioro, por el cual es necesario realizar mantenimiento, reparaciones y reemplazos de éstos. Las bombas centrífugas forman parte de muchos procesos productivos, en una amplia variedad tanto de pequeñas como de grandes industrias. Conocer cuándo deben realizarse reparaciones, mantenimientos o reemplazos sobre estas bombas es una labor que no siempre resulta sencilla o evidente, lo que implica la adquisición de equipos para la detección de fallas. La visualización del funcionamiento interno de bombas centrífugas a través de la utilización de sistemas computacionales, proporciona una herramienta para hacer la tarea de detección de fallas más sencilla, y a través de la simulación en ambientes virtuales es posible ver los efectos en los componentes que son susceptibles al fenómeno de desgaste. Los ambientes virtuales no inmersibles son una variedad de los ambientes de realidad virtual, y no requieren del hardware de inmersión, lo que proporciona una ventaja por su bajo costo. El Lenguaje de Modelación de Realidad Virtual (VRML) es un estándar para la creación de este tipo de ambientes virtuales no inmersibles, con éste se puede crear escenarios virtuales interactivos, controlados a través de un interfaz creada para tal fin, que bien resulta en una aplicación independiente o en una aplicación para la distribución en la Web (applet). A través de VRML se logra dar solución total o parcial a problemas para los cuales no era posible la utilización de las herramientas existentes, o bien dar un nuevo empuje a soluciones en donde el componente gráfico era pobre o inexistente.

Palabras clave: Visualización, realidad virtual, inspección, bomba centrífuga, buscador.

Design and implementation of an inspection system to visualize a centrifugal pump in a virtual reality environment

Abstract

All industrial process implies, by itself, the use of equipments and tools in order to achieve its goals, and also a process of deterioration in them. Thence, it is necessary a strategy of maintenance, repairment and replacement of the equipment and tools. Centrifugal pumps are part of many productive processes, in a wide variety, in small and big industries. Knowing the best time for the maintenance or replacement of these pumps is not an easy or simple task to accomplish. Thus, the acquisition of equipment for fault detection becomes very appropriate. The visualization of the internal operation of a centrifugal pump, using a computational system, provides a tool to make fault detection a simpler task. Using a virtual environment, it is possible to see the effects on the components that are susceptible to the wear phenomenon. A non-immersible virtual environment is a variety of the virtual reality environments that do not require immersion hardware and offers advantages due to the lower cost. The Virtual Reality Model Language (VRML) is a standard for the creation of the non-immersible virtual environment type. VRML allows the creation of interactive virtual stages controlled through an specially designed interface, resulting in an independent application or a Web distribution application (applet). VRML makes possible the partial or complete solution of problems for which the application of existent tools was not feasible, and a new perspective is given for solutions where the graphical component was poor or almost nonexistent.

Keywords: Visualization, virtual reality, inspection, centrifugal pump, browser.

1. INTRODUCCIÓN

El desarrollo de las nuevas tecnologías que avanzan a paso vertiginoso, ha creado para el ciudadano de hoy, la mentalidad de que la solución a cualquier problema, por complejo que sea, puede hacerse a través de la utilización de computadores. Si bien esto es cierto, también es cierto que a corto plazo no siempre resulta sencillo o práctico la utilización del computador para dar soluciones a un problema; pero puede hacerse en forma parcial, para después estructurar la solución global inicialmente planteada.

El proceso de dar soluciones parciales a problemas de gran envergadura o complejidad, se ha venido llevando a cabo en muchas áreas, por ejemplo, hace pocos años el concepto de realidad virtual, resultaba ciencia-ficción, y poco a poco su evolución ha dado el fruto requerido, primero con la aparición de los ambientes gráficos, y luego con la integración del hardware a estos ambientes. Esta evolución no siempre es lenta, y no siempre puede preverse con claridad hacia donde va, o que se tendrá luego de algún tiempo.

Sin embargo, esto no priva de la exploración en otras áreas, cada día podemos ver como nace una nueva área de estudio, en donde se utilizan recursos computacionales para su estudio, sin importar que tan complejo parezca ser o que tan imposible. Un claro ejemplo de esto está en el proyecto que llevan a cabo varios países a escala mundial en el área de estudio de genoma humano, hace 20 años era una tarea imposible, hace 10 se convierte en posible y hoy por hoy esta a punto de concluir.

Como en el caso del ejemplo anterior, el área de ingeniería de predicciones es compleja, y aún se encuentra en etapa de desarrollo, cada día cuenta con mayores adeptos en búsqueda de soluciones a problemas como la predicción del comportamiento de equipos en funcionamiento que sufren deterioro por causa de la interacción con agentes erosivos que producen cambios dimensionales en varios de sus componentes.

La industria, debido a que se ve directamente afectada por este fenómeno, realiza inversiones con el fin de solventar de alguna manera el problema que representa conocer con anticipación cuando sus

equipos deben ser reemplazados total o parcialmente, para así tener bajo control estos sucesos y realizar preventivamente las correcciones necesarias para minimizar el impacto sobre su proceso productivo.

Tal vez uno de los componentes industriales de mayor uso sea las bombas, y en particular las bombas centrífugas. Estas son utilizadas para transportar fluidos necesarios en el proceso productivo, sea cual fuese éste, y el correcto funcionamiento de las bombas es fundamental. El desgaste es uno de los fenómenos que se hace presente con mayor frecuencia en este tipo de equipos, y que ocasiona fallas en su funcionamiento, llegando incluso en los casos más severos a la paralización total de la bomba.

Para la visualización del desgaste es necesario desarmar la bomba, luego rearmarla. Esto debe hacerse cuando se tiene la sospecha, bien por el comportamiento de la bomba, o bien por recomendaciones del fabricante. Pero si pudiésemos ver en el interior de la bomba y observar el proceso de desgaste, se sabría con exactitud cuando realizar el desarme, y en otro caso, el reemplazo completo de ésta.

Los ambientes gráficos han sido utilizados para representar a través del computador el mundo real, de allí nace la nueva concepción de realidad virtual. Aunque esta revolución no ha terminado por completo, sus efectos pueden sentirse, pero la necesidad de equipos especiales limita el alcance de la tecnología de realidad virtual; es así que nace un nuevo concepto para realidades virtuales no inmersivas, y con ella nace VRML, que proporciona las mismas características de los ambientes virtuales, pero sin la capacidad de inmersión [1-2].

2. METODOLOGÍA

Antes de adentrarnos en la selección del ambiente virtual, debemos entender con claridad que son los ambientes virtuales no inmersivos y su relación con VRML.

Los ambientes de Realidad Virtual (VR) permiten la interacción del usuario con un ambiente gráfico tridimensional, y en donde este usuario se encuentra inmerso. Existe una variante de este modelo, los ambientes virtuales no inmersivos, en donde el usuario no puede "introducirse" en el ambiente

sino interactuar con éste, a través del teclado y ratón. VRML es un estándar para la creación de escenarios tridimensionales no inmersivos, pero por sí solo, no constituye el escenario, necesita del software que despliega el mundo virtual. Este software (Browser VRML) carga el archivo en VRML, los objetos, la iluminación, las texturas, sonidos, acciones, dejando preparado el ambiente para su interacción con el usuario [3-5].

Un mundo virtual o escenario virtual en VRML es un archivo de texto con la extensión WRL, que contiene la definición de los objetos y sus características para desplegarse en el Browser VRML.

El Browser VRML permite cierto nivel de control sobre sus objetos del mundo virtual, acciones como zoom in, zoom out, desplazamiento horizontal y vertical, rotación, son previsto por el software; pero no es posible eliminar o agregar objetos, cambiarles el color ni cualquier otra característica [6].

VRML permite la creación de mundos virtuales que manipulan los objetos contenidos en éste. La manipulación está limitada a algunas características de los objetos y su creación es compleja, además de ser rígida. Por ejemplo, es posible hacer rotar un objeto (por sí solo) pero para esto es necesario crear el objeto en primera instancia, luego crear un segundo objeto (invisible en el Browser) que controla la rotación, y un tercer objeto que inicia y detiene la rotación, además hay que crear vínculos entre cada uno de estos objetos, para instruir al Browser acerca de la rotación; Este proceso habría que hacerlo para cada objeto que se requiera rotar, y el proceso es similar para cada característica que se desee cambiar de un objeto.

En la especificación de VRML 1.0 este dinamismo era imposible pero para VRML 2.0 se incluyó, además de la posibilidad de manipulación de los objetos del Browser, la posibilidad de manipulación externa, es decir, a través de una interfaz ajena al ambiente virtual, utilizando una External Authoring Interface o EAI contenida en la especificación de VRML 2.0 y ampliada en VRML 97 [7].

La EAI permite manipulación del Browser y de los objetos contenidos en éste, permite agregar y eliminar objetos del Browser, cambiar el tamaño, posición, color, rotación, y casi cualquier otra

característica de los objetos. Para que sea posible la construcción de una EAI se necesita tener las especificaciones que proporciona el fabricante del Browser, además las bibliotecas necesarias, proporcionadas también por el fabricante.

Actualmente existe un gran número de Browser VRML adaptados para archivos de VRML 97, entre estos podemos mencionar a InterView y Cosmo Player 2.1, ambos con capacidad para la creación de una EAI y de manejo e instalación muy sencillos, además de que pueden ejecutarse en plataformas Windows 95/98/NT4.0. Para plataformas como Linux y Unix son escasos los ambientes virtuales que pueden encontrarse, y para el momento de la realización de esta investigación, no existe ningún Browser VRML que permita la construcción de una EAI para plataforma Unix o Linux [8-9].

La adquisición de un Browser VRML no es suficiente para hacer posible la visualización de un archivo VRML, éste se debe a que el Browser no es un programa completo, es un plug-in para navegadores Web.

Principalmente la diferencia entre InterView y Cosmo Player se debe a que el Cosmo Player fue el primero en hacer su aparición, permite mayor funcionalidad, la documentación es más extensa y puede instalarse sobre casi cualquier navegador Web, además de ser el más utilizado. Por estas razones y para fines de esta investigación se eligió la utilización del plug-in Cosmo Player 2.1 desarrollado por Computer Associates International Inc. Utilizando el modelo de applet y con programación en Java. Se utiliza la librería NPCOSMOP21.ZIP, proporcionada por Cosmo Software, que contiene las clases que permiten la interacción de la EAI con el ambiente virtual [10].

3. DESARROLLO DE LA INTERFAZ (EAI)

Para poder dar una buena explicación de este paso, es necesario establecer abreviaturas para los diferentes elementos que se utilizarán, éstas se mencionan a continuación:

- MV (Mundo Virtual): Archivo de texto con extensión WRL que contiene información en VRML de uno o más EMV.

- EMV (Elemento del Mundo Virtual): Objeto o pieza que forma parte del mundo virtual. Cada EMV puede o no desplegarse en el ambiente virtual.
- BVRML (Browser VRML): Donde se visualizan los objetos del MV.
- EBV (Elemento del Browser VRML): Todo objeto que pertenece al ambiente y sobre el cual se tiene control.
- EEAI (Elemento de Interfaz): Todo objeto creado en la interfaz y que tiene equivalente en el Browser VRML.

Ahora bien, los objetos que se muestran en el BVRML pueden manipularse a través de la interfaz. Para esto es necesario crear un nuevo objeto (EEAI) que posea las características que se desean controlar. Este objeto no pertenece al BVRML ni al mundo virtual, es un objeto de la interfaz, pero es a través de éste que se controla un objeto del BVRML [11-13].

Si se quiere controlar o manipular todos los elementos del ambiente (EBV), deben crearse tanto elementos de la interfaz (EEAI) como elementos (EBV) que se quieran controlar. Este control se lleva a cabo a través de la recepción y envío de eventos desde y hacia el BVRML. Cada elemento de la interfaz debe estar equipado con las directrices de eventos que se quieren controlar, además debe construirse el MV adecuado para la recepción de eventos. Por ejemplo, si se desea controlar la rotación de un elemento desde la interfaz, digamos que queremos hacer un cubo rotativo, entonces en primera instancia se debe crear en el MV la definición para recibir un elemento desde el exterior, escribiéndose la siguiente línea en el archivo.

4. DEF nomNodo Group { }

Donde nomNodo es cualquier nombre que se le quiere dar a este nodo. Esta línea crea una definición vacía, sin elementos, ni tamaño, ni color, ni forma. Este nodo vacío es la ventana para el ingreso de unos o de varios EEAI. En la interfaz se captura este nodo a través del método getNode, esto crea un vínculo entre el MV y la EAI. Posteriormente se crean los objetos que se desean controlar, esto se logra utilizando la función de la librería createVrmlFromString, la cual solo necesita una cadena (String) que contenga la misma estructura de VRML para la creación de objetos [14].

Para crear los EEAI es necesario tener la codificación en VRML de los objetos que se desean, y la IRV adquiere estos objetos desde los archivos de la bomba. Para tal fin se proporciona la capacidad de especificar el archivo que se desea cargar, luego se analiza extrayendo de cada archivo las piezas de la bomba de la siguiente manera:

- Luego de abrir el archivo, se extrae todo su contenido en una variable tipo String, y se cierra el archivo.
- Con esta variable se analiza el texto del archivo, en busca de nodos "IndexedFaceSet", campos "material", nombre de las piezas, indicadores de rotación y el tiempo de vida útil.
- Con cada nodo "IndexedFaceSet", "material", nombre e indicadores de rotación se construye un EEAI, que está dotado de capacidad para rotar, trasladarse, cambiar de color y cambiar de posición. Todos los EEAI están agrupados en una estructura de vector para su manipulación.
- Luego de creados los EEAI en el vector, se recorren uno a uno, buscando el indicador de rotación, para crear una copia de todas las piezas de la bomba que puedan rotar, y se almacena en otro vector. Cabe destacar que cada pieza de rotación crea un hilo de ejecución independiente para realizar su tarea, esto permite la sincronización en los movimientos de rotación de todo el conjunto.
- Posteriormente se procede a cargar los archivos de desgaste que se encuentren en el directorio destinado para esto y que el usuario especifica. Cada archivo de desgaste debe estar asociado a una pieza. Esta asociación se logra a través del nombre de pieza, el cual debe estar especificado tanto en el archivo de desgaste como en el archivo de la bomba. Únicamente se extrae de cada archivo de desgaste el nodo "IndexedFaceSet", el nombre asociado y el tiempo de desgaste que representa a la pieza. Los elementos de desgaste se almacenan en una matriz, en donde las filas corresponden a los niveles o tiempo de desgaste y las columnas a las piezas desgastadas.

El proceso de carga y análisis de los archivos puede tomar de unos pocos segundos, hasta varios minutos, estos dependen del número de piezas y de su complejidad [15, 16].

5. CREACIÓN DE LOS ARCHIVOS DE LA BOMBA

5.1. Bomba sometida a Desgaste

Como ya se mencionó antes, todo elemento que se desee desplegar en un ambiente virtual a través de una EAI, debe estar codificado en VRML. La bomba fue construida originalmente en Auto CAD, en 3D, y cada pieza en un archivo diferente, y posteriormente se realizó el ensamble en un único archivo.

Luego de tener todas las piezas de la bomba en un único archivo, a partir de éste se realizó la conversión a VRML, esto se logró con la utilización de una herramienta proporcionada por Autodesk C.A., la cual permite convertir un sólido 3D a formato VRML, la conversión se realizó para cada pieza de la bomba individualmente. Es necesario tener en cuenta en la conversión la escala a la cual se realiza, es decir, la conversión debe realizarse a la escala adecuada para la correcta visualización en el ambiente, ya que si se hace a una escala muy pequeña o muy grande, el movimiento a través del ambiente para visualizar la bomba desde diferentes puntos de vistas, puede ser engorroso. Para el ambiente el tamaño no importa, pero para quien visualiza sí. Por ejemplo, si se crea un objeto de 3 unidades (de longitud), el ambiente coloca el punto de vista 2 ó 3 unidades, de modo que para acercarse a éste sólo hace falta recorrer 1 ó 2 unidades y quedar lo suficientemente cerca para observar los detalles. Pero si en cambio el objeto es de 3 ó 4 mil unidades, el ambiente debe colocarse a 2 ó 3 mil unidades de distancia para abarcarlo en su totalidad. Recorrer 3 ó 4 mil unidades en el ambiente para apreciar los detalles de cada pieza toma mucho tiempo y la velocidad de visualización se degrada sensiblemente [17].

Una vez que se han generado los archivos para cada una de las piezas de la bomba, se agrupan en un único archivo, éste será el que se leerá en la IRV para su visualización.

La IRV requiere además, que se especifiquen algunas características de cada pieza y de la bomba, esto es: nombre de la pieza y un indicador de si ésta tiene movimiento o no, y el tiempo de vida útil de la bomba. Esto se indica agregando dos líneas por cada pieza, una que indica el nombre y otra que indica si la pieza está rota o no. Puede escribirse antes del inicio de la codificación de cada pieza, o en cualquier otra

parte del archivo, pero siguiendo los parámetros siguientes:

- Para el nombre se escribe: # Nombre = nombre - de- la- pieza;
- Para la rotación se escribe: #Rota = indicador-booleano;
- Para el tiempo de vida útil: # Vida Útil =vida-útil- en- horas.

Es importante mantener el formato, los indicadores (#Nombre,#Rota= y #Vida útil=) deben escribirse tal y como se muestran, con el signo # que los precede y la primera letra en mayúsculas, de lo contrario el sistema no podrá reconocer los indicadores, luego del indicador se escribe el valor correspondiente a ese indicador y se termina la línea con un punto y coma. El orden en que aparecen las piezas debe ser el mismo orden que los nombres, de lo contrario no habrá concordancia. Si se olvida alguno de estos parámetros la IRV tomará los siguientes valores por defectos:

- Si no existen nombres, cada pieza tendrá el siguiente nombre dado por la IRV: "Pieza # ", donde # es el número de la pieza, empezando por 1 hasta el número de piezas que contenga el archivo. Si se encuentran más piezas que nombres, se identifican las piezas hasta la cantidad de nombres que se encuentren, es decir, si se encuentran 10 piezas y 7 nombres, las 3 últimas piezas se nombrarán "Pieza 8", "Pieza 9" y "Pieza 10". Si existen más nombres que piezas, se ignoran los últimos nombres.
- Si no existe identificación de rotación, ninguna pieza rotará, si existe más piezas que indicadores, las últimas piezas se rotarán, si existen más indicadores que piezas, se ignora los últimos indicadores.
- Si no existe indicador de vida útil, se asume 50.000 horas.

5.2. Archivos de Desgaste de la Bomba

Para simular el desgaste de la bomba, fue necesario crear los archivos de las piezas desgastadas. Para cada pieza se generaron varias piezas, cada una con un grado de desgaste mayor, hasta alcanzar el nivel de desgaste deseado. Luego de generar cada archivo de desgaste, se agruparon todos los archivos

correspondientes a cada una de las piezas en un único archivo, identificado con el nombre de la pieza al cual pertenece, y con los tiempos de desgaste [18].

Debido a la poca información que se tiene en cuanto a desgaste en piezas mecánicas de este tipo, los archivos generados solo pretenden dar al usuario una aproximación al comportamiento real del desgaste en bombas centrífugas.

Los archivos de las bombas se crearon utilizando AutoCAD, a partir de allí se generaron a VRML utilizando el convertidor para VRML que Autodesk proporciona. La ventaja de este convertidor, además de ser rápido y eficiente, es que permite que la conversión sea tal y como se muestra en el ambiente de diseño de AutoCAD. Pero tiene la desventaja que no es posible agregar detalles de calidad como definición de bordes, por lo que algunas de las piezas generadas carecen de estética y pueden observarse con bordes filosos, y ángulos pronunciados.

6. RESULTADOS Y DISCUSIÓN

A continuación se muestra una ejecución típica de la IRV, la secuencia de acciones y las ventanas que permiten el control sobre el ambiente virtual, debe aclararse que para la ejecución se utilizó como Navegador Web el Explorer 5.0 de Microsoft por recomendaciones del fabricante del Browser VRML quien indica que el plug-in Cosmo Player 2.1 se ejecuta con mayor eficiencia en este Navegador Web. El hardware utilizado para la ejecución fue un equipo Pentium 233 MMX. Con 64 Mb en memoria RAM, disco de 2 Gb y tarjeta gráfica de 4 Mb, sin acelerador gráfico, en plataforma Windows 95/98/NT4.0, se eligió este equipo por estar dentro de los requerimientos mínimos exigidos para la ejecución de Cosmo Player 2.1, y aunque la plataforma de desarrollo fue Windows NT4.0, pero en Windows 2000 fue imposible realizar la ejecución debido a incompatibilidades del Browser VRML con esta plataforma. El tiempo total de ejecución fue 8.5 minutos en Windows NT4.0, 9.5 minutos en Windows 95 y 8.7 minutos en Windows 98.

6.1. Carga de los Archivos de la Bomba y Desgaste

El primer paso a realizar es cargar el archivo HTML que contiene el applet de la IRV y la

referencia al escenario virtual IRV.WRL en el Navegador Web, luego que se realiza esta acción se presenta la interfaz que se muestra en la Figura 1.

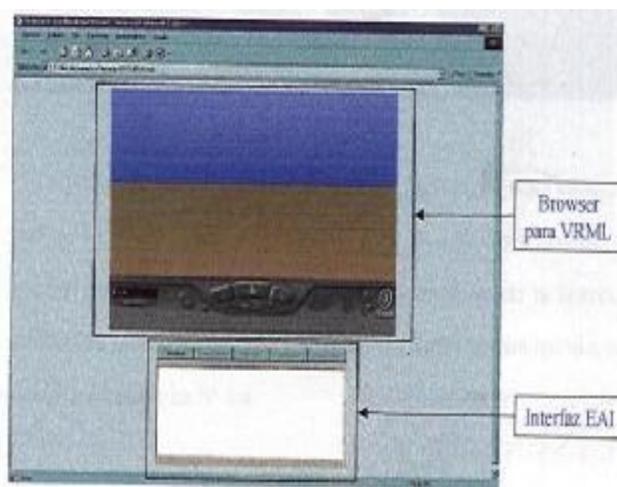


Figura 1. Carga del archivo IRV.HTML.

En esta primera vista se puede apreciar en la parte superior el escenario virtual que se encuentra codificado en el archivo IRV.HTML, éste actualmente está vacío y solo puede apreciarse, en la parte inferior el applet IRV.CLASS que lo controla. En el escenario se puede identificar en la zona inferior de éste, los controles que proporciona el fabricante, es decir, Cosmo Player, estos controles permiten realizar varias acciones, tales como: zoom in, zoom out, rotaciones, desplazamientos horizontales y verticales, etc.

En el applet se aprecia que solo un botón está disponible, el correspondiente a "Archivo" (Ver Figura 2). Este permite al usuario especificar las ubicaciones de los archivos de la bomba, cuando se pulsa se muestra el cuadro de diálogo que se observa en la Figura 3.

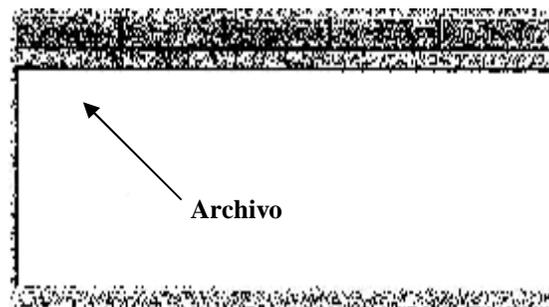


Figura 2. Botones de la interfaz.

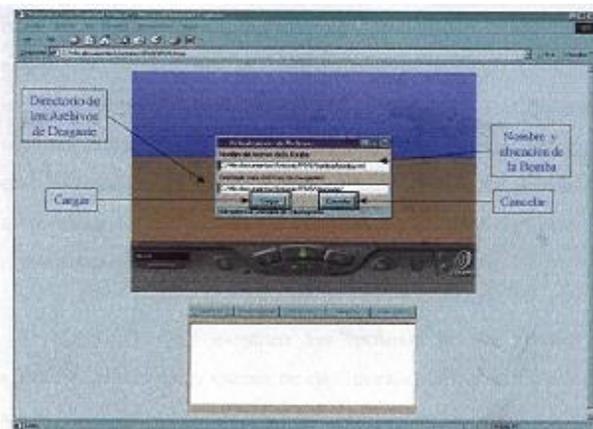


Figura 3. Petición de archivos.

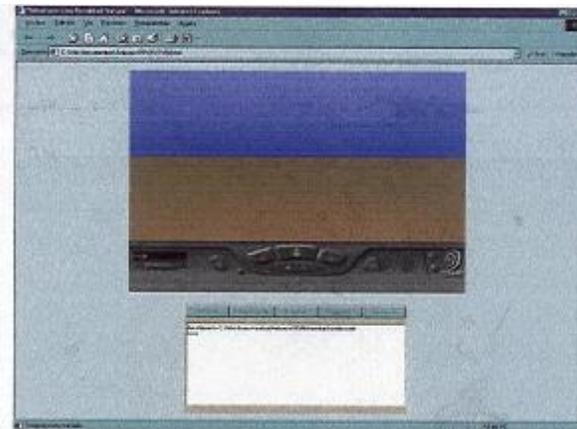


Figura 4. Carga de los archivos de la bomba.

En este cuadro de diálogo se pide la dirección y nombre del archivo de la bomba original, en primera instancia (primer cuadro de texto) y el directorio que contiene los archivos de la bomba desgastada (segundo cuadro de texto). Una vez identificado el archivo de la bomba y el directorio de desgaste, se pulsa el botón "Cargar". Hecho esto, la IRV verifica la existencia tanto del archivo como del directorio en cuestión. Si uno de estos no existe, la IRV proporciona a través del área de texto bajo los botones la información del error producido. El cuadro de diálogo no se cierra hasta localizar los archivos adecuados o hasta que se le indique con el botón "Cancelar". Es importante saber que debido a que se está trabajando con un applet, la localización de los archivos debe estar en un directorio local, el applet no podrá acceder a directorios remotos, ni en la red de área local, ni a través de la Web, es por esto que los archivos deben guardarse en directorios locales, respetando la condición de que los archivos de desgaste y el original de la bomba deben encontrarse en directorios diferentes.

Luego de que se localizan los archivos, la IRV procede a cargarlos, analizarlos y extraer de estos la información necesaria para la ejecución, hasta no completar este proceso no se habilitan el resto de los botones de la interfaz. Este proceso puede tardar de 2 a 4 minutos, dependiendo de la cantidad de archivos, de su complejidad y de la velocidad del equipo en que se encuentre ejecutando la aplicación. Para el caso de esta prueba, la carga del archivo original y 4 archivos de desgaste se realizó en 2 minutos con 40 segundos. En las Figuras 4 y 5 puede verse la secuencia de este proceso.

Primero se busca y se carga el archivo de la bomba sin desgaste, una vez concluido el análisis, se despliegan todas las piezas de la misma, mostrándose gradualmente pieza a pieza hasta completar el armado en el escenario virtual, como se observa en la Figura 5.

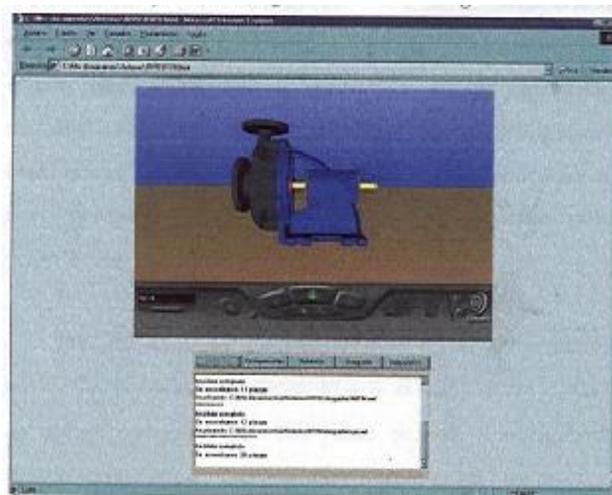


Figura 5. Carga de los archivos de desgaste.

A medida que se cargan los archivos, se muestra la información referente a estos: nombre del archivo, ubicación y cantidad de piezas que contiene. Una vez concluido este proceso se habilitan el resto de los botones (a excepción del botón Archivos") y la IRV esta lista para comenzar a trabajar (Ver Figura 6).



Figura 6. Botones activos luego de la carga de archivos.

6.2. Adición y Eliminación de Componentes de la Bomba

Una vez concluida la carga, el botón de "Archivo" queda deshabilitado, debido a que no está permitido cargar una nueva bomba, a menos que se vuelva a cargar el applet. Esta restricción se debe a que el escenario virtual mantiene la información de los objetos que se agregan o eliminan de éste. La información permanece en el escenario, sin poder ser controlada por el usuario, hasta que el applet es destruido o reinicializado. Para cargar una nueva bomba es necesario cerrar la ventana del Navegador Web y volver a cargar la página o actualizar.

Para agregar o eliminar componentes a voluntad se pulsa el botón "Componentes" (Ver Figura 6), el cual muestra el cuadro de diálogo correspondiente (Ver Figura 7). Este cuadro de diálogo contiene tres listas de componentes: los "Visibles", los "Disponibles" y los "Rotando".

Los "Visibles" son los componentes que se están mostrando actualmente en el escenario virtual, inicialmente todos los componentes de la bomba están visibles.



Figura 7. Cuadro de diálogo para componentes.

La lista "Disponibles", indica cuales son los componentes que se encontraron en el archivo de la bomba sin desgaste.

La lista "Rotando", contiene los componentes a los cuales se les localizó el indicador "# Rota =Si". Esta lista debe manejarse con cuidado y de preferencia nunca modificarse, esto es porque si se agrega o elimina una pieza incorrectamente, los resultados pueden ser impredecibles. Por ejemplo, si se agrega a esta lista la voluta, entonces la rotación de esta pieza haría parecer que existe un error en cuanto al análisis de los archivos.

Ahora veremos como puede agregarse y/o eliminarse piezas del ambiente. Para este ejemplo eliminaremos las piezas voluta, base y eje como puede verse en la figura 8, para ello se selecciona de la lista el componente voluta, luego se pulsa el botón ">", inmediatamente el botón "Aplicar" se activa detectando un cambio en el contenido de las listas. Ahora sólo es necesario presionar "Aceptar" o "Aplicar" para que los cambios tengan efecto, si se realiza un doble-click en el nombre de algún componente de la lista "Visibles" o "Rotando", este se elimina de la lista, teniendo el mismo efecto que seleccionar el nombre y pulsar el botón ">" para los "Visibles" o el botón "<" para los "Rotando".

Al "Aplicar" los cambios puede verse el resultado en el escenario (Ver Figura 9), y rectificar, ya que el cuadro de diálogo no se cierra. Si presionamos "Aceptar" se realizan los cambios en el escenario y se cierra el cuadro de diálogo. Si no queremos realizar ningún cambio al escenario, simplemente presionamos "cancelar". Los cambios hechos con el botón "Aplicar", es decir, si se aplican los cambios, no se puede revertir el efecto con el botón "Cancelar".



Figura 8. Eliminando la voluta, base y eje.

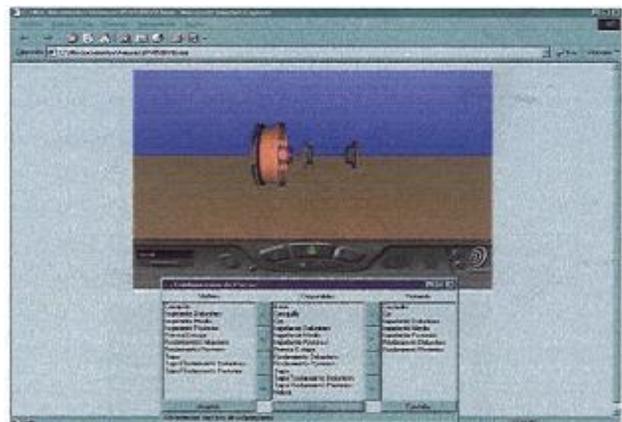


Figura 9. Aplicando los cambios.

El botón ">" eliminan uno a uno los componentes, mientras que el ">>" los elimina todos al mismo tiempo de la lista "Visibles". En cambio "<" y "<<" agregan, uno a uno y todos, respectivamente, a la lista "Visibles". La funcionalidad de los botones es similar en la lista de "Rotando".

6.3. Rotación

Para ver la bomba en funcionamiento, basta con pulsar el botón "Rotación" (Ver Figura 6), este inicia inmediatamente un hilo de ejecución para cada pieza que se encuentre en la lista de rotaciones del cuadro de diálogo de "Componentes", como se ve en la Figura 10.

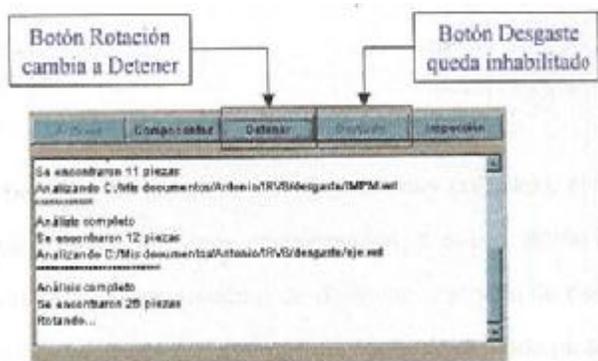


Figura 10. Rotando.

Debido a que esta es una imagen estática no puede apreciarse el movimiento, pero en la interfaz puede verse como el botón de "Desgastes" queda inhabilitado por la rotación. Esto se debe a que como el proceso mismo de rotación, es un hilo de ejecución para cada pieza (condición necesaria para que la rotación sea síncrona) y los desgastes se requieren realizar también en un hilo de ejecución para cada pieza que se desgasta, esto hace imposible realizar modificaciones a las piezas de rotación, por que al intentar detener el hilo de rotación, se genera una excepción proveniente del Monitor de hilos que la Máquina Virtual de Java controla. Esto no impide ver la rotación de las piezas desgastadas.

Cuando se pulsa el botón de rotación, este cambia su estado a "Detener", de manera que para detener la rotación solo es necesario pulsar de nuevo en este mismo botón.

6.4. Desgaste

El proceso de desgaste es un poco más complejo.

El cuadro de diálogo consta de varios componentes, a saber: Botón Iniciar, botón cancelar, tiempo máximo de desgaste, intervalo de desgaste, lista de piezas a desgastar y tiempo de desgaste de cada pieza. Para ver el proceso de desgaste, se escoge en primera instancia el tiempo máximo que se simulará el desgaste, luego se elige el intervalo de tiempo a considerar y por último se presiona el botón iniciar. En ese mismo momento cuando comienza el proceso, al mismo tiempo aparecen indicadores de tiempo de desgaste para cada pieza, hasta alcanzar el tiempo máximo de desgaste (Ver Figura 11).



Figura 11. Diálogo desgaste.

Puede verse aquí los componentes antes mencionados, y también se observa que los botones de componentes, el botón de Rotación se encuentra deshabilitado, hasta terminar la ejecución del desgaste.

Puede observarse el progreso de desgaste en la interfaz, y es de mencionar que no puede detenerse el proceso, hasta que éste alcanza el valor de tiempo máximo de animación. Una vez concluida esta fase, se puede observar los efectos del desgaste en las piezas de la bomba. La Figura 10 muestra estos efectos en el eje de la bomba.



Figura 12. Desgaste en ejecución.

Si se quiere se puede iniciar de nuevo el proceso de desgaste, cambiando los parámetros, y la IRV comenzará de nuevo desde el principio a realizar el desgaste, así mismo en las figuras 15 y 16 para los álabes de la bomba.

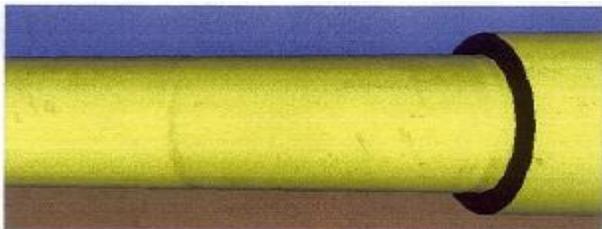


Figura 13. Eje sin desgaste.

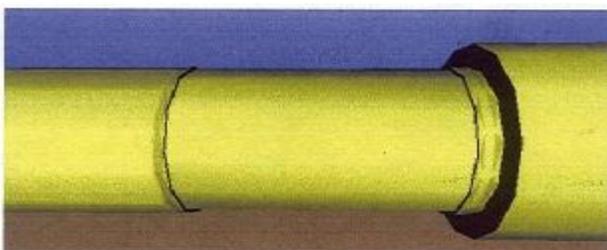


Figura 14. Eje desgastado.



Figura 15. Alabes sin desgaste.



Figura 16. Alabes desgastados.

6.5. Inspección

Una vez concluido el proceso de desgaste es interesante ver cómo se quedan los componentes de ésta, es por ello que existen rutas de inspección predefinidas que muestran los principales aspectos del impelente, eje y rodamientos.

Para inspeccionar una pieza, solo es necesario elegir la ruta apropiada. Una ruta es una secuencia de puntos en el espacio que llevan al usuario a visualizar con detalle aspectos de interés en la bomba. Si alguna de las piezas a inspeccionar no está visible, la IRV la coloca visible temporalmente y una vez terminada la ruta de inspección, la IRV vuelve el escenario al estado antes del inicio de la inspección. Es posible controlar la velocidad de inspección modificando el valor que contiene la lista de "Retardo". Está expresada en milisegundos cuando se tarda la IRV entre dos puntos de la ruta de inspección, el retardo por defecto es cero.

En la Figura 17 puede verse el cuadro de diálogo para las rutas. Cabe destacar que una vez iniciada la ruta, esta puede pausarse para detallar una zona en específico, o puede detenerse por completo para volver al estado inicial. Las rutas se ejecutan en una secuencia fija, no pueden modificarse pero son los bastantes detalladas como para visualizar completamente la bomba.

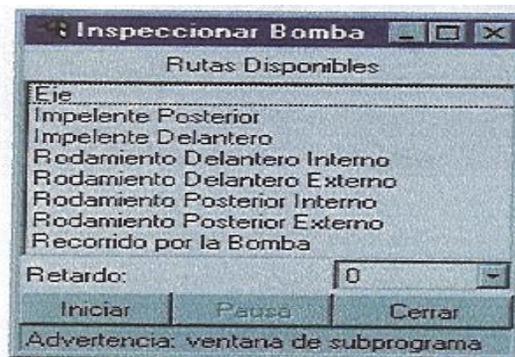


Figura 17. Diálogo de rutas de inspección.

El retardo puede configurarse desde 0 milisegundos hasta 2000 milisegundos, es decir, de 0 a 2 segundos, con intervalos de 100 milisegundos (Ver Figura 18).



Figura 18. Distribución de los retardos.

A continuación se muestra una secuencia de Figuras desde la 20 hasta la 25 que permiten apreciar algunas etapas del proceso de inspección del eje.



Figura 19. Inspección eje 01.

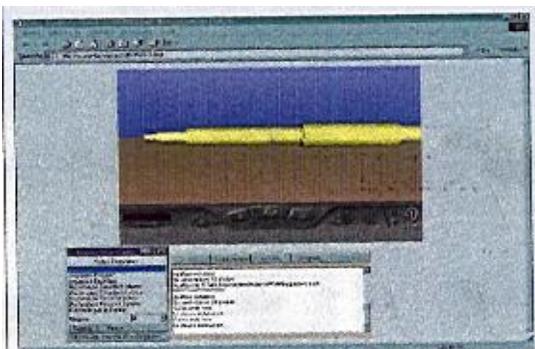


Figura 20. Inspección eje 02.

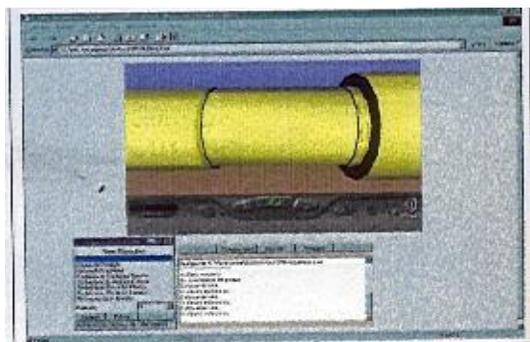


Figura 21. Inspección eje 03.

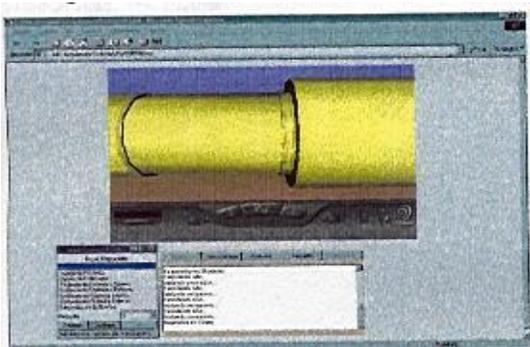


Figura 22. Inspección eje 04.

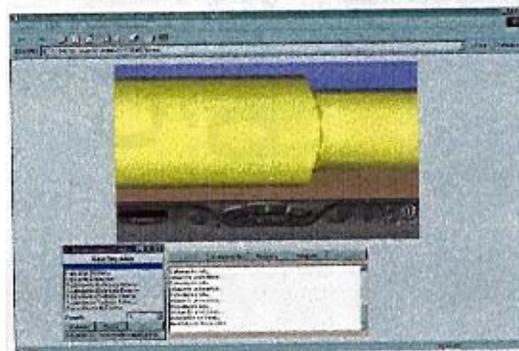


Figura 23. Inspección eje 05.

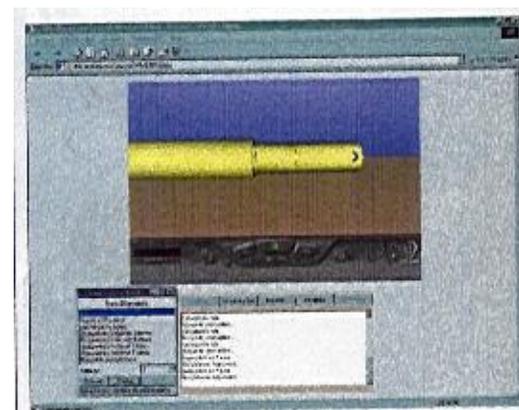


Figura 24. Inspección eje 06.

7. CONCLUSIONES

Los sistemas de simulación se basan en el comportamiento matemático del fenómeno a simular, pero el desgaste en bombas centrífugas no obedece a ningún comportamiento matemático establecido, sin embargo puede hacerse una aproximación, utilizando para ello los patrones de desgaste visibles en estos equipos.

- La capacidad de VRML para la simulación de comportamientos reales, sin la obligada adquisición de equipos (como lo es en realidad virtual) lo convierte en una herramienta variable y rentable, pero con la salvedad de que este nunca podrá superar las capacidades que ofrece la realidad virtual inmersiva.
- Las barreras en cuanto a las condiciones que deben establecerse para tener el control de los componentes, son evidentes en cuanto se intenta realizar acciones complejas. Sin embargo la flexibilidad que ofrece VRML para la anidación de nodos, permite encapsular aquellas áreas de gran complejidad con poco esfuerzo y tiempo.

- Así se logra el desarrollo de una solución diferente a problemas que anteriormente resultaban muy complejos o imposibles de realizar con las herramientas existentes, o bien dar nueva vida a soluciones existentes pero que carecían de la parte visual para mostrar sus resultados.

8. BIBLIOGRAFÍA

- [1] The VRML Repository. 1999. <http://www.webed.com.vrml>
- [2] VRML Java. 1999. "VRML Magazine". <http://www.vrmlsite.com>
- [3] Ames, A., Nadeau, D., Moreland, J. 1996. "THE VRML Sourcebook". Jhon Wiley & Sons, Inc.
- [4] Meier, A. 1998. "Herramientas para la creación de un museo Virtual". Memorias del IV Congreso Nacional sobre Multimedia y Videoconferencia.
- [5] Nadeau, D. 1999. "Introduction to vrml". <http://tecfa.unige.ch/guides/vrml>
- [6] Autodesk Web Site. 1999. <http://www.autodesk.com>
- [7] Carey, R., Bell, G. 2000. "The Annotated VRML 97 Reference", Book on Line. <http://www.best.com/~rickk/book>
- [8] Proposal for a VRML 2.0 Informative Annex. "External Authoring Interface". 1999. <http://www.vrml.org/specification/vrml97>
- [9] SGI Virtual Reality. 2000. <http://www.europe.sgi.com/virtual.reality/overview>
- [10] Cosmo Player 2.1. Cosmo Software. 2000. <http://www.cai.com/cosmo>
- [11] Nadeau, D. 1997. "VRML Feature Summary. DOCT Project Write Paper". <http://www.vrml.org>
- [12] Tutorial para Java. 2000. <http://tutorjava.8k.com>
- [13] VR-SYSTEM LABS Web Site. 1999. <http://vr-system.com>
- [14] Steed, A. "The VRML External Authoring Interface". 2000. University College London. <http://www.cs.ucl.ac.uk/staff/A.Steed/eai/>
- [15] San Diego Súper Computer. 1999. <http://www.sdsc.edu>
- [16] WEB3D Consortium. 1999. <http://www.vrml.org>
- [17] Ramírez, M., Hernández H., D. 1998. "Implantación para la Generación de Hipermapas 3D". Memorias del IV Congreso Nacional sobre Multimedia y Videoconferencia.
- [18] Zubicaray, M. 1997. "Bombas. Teoría, Diseño y Aplicaciones". Segunda Edición. Editorial Limusa. México.