

UNIVERSIDAD POLITÉCNICA DE VALENCIA

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y
COMPUTACIÓN



**SISTEMA DE AYUDA A LA ENSEÑANZA
DE GEOMETRÍA BASADO EN REALIDAD
AUMENTADA**

Máster en
Inteligencia Artificial Reconocimiento de Formas e Imagen Digital

Gustavo Alberto Roveló Ruiz

Dirigido por
Emilio Camahort Gurrea
Francisco J. Abad Cerdá

Valencia, España. Julio 2009

Agradecimientos

Agradezco a mis padres y hermana por su apoyo incondicional. Por animarme junto con mis amigos a seguir adelante a pesar de las adversidades.

Agradezco también al Consejo Nacional de Ciencia y Tecnología de México por financiar mis estudios de Máster. También hago patente mi reconocimiento al grupo de Investigación de Informática Gráfica del Departamento de Sistemas Informáticos y Computación por el apoyo académico y las facilidades otorgadas durante los dos años de mi postgrado, en especial a los profesores Emilio Camahort y Francisco Abad por sus enseñanzas y su paciencia.

Resumen

Uno de los principales objetivos de las instituciones educativas ha sido buscar nuevos métodos para lograr que los alumnos asimilen los conocimientos de manera eficiente. Es por ello que a lo largo de la historia se han adoptado nuevas herramientas tecnológicas para su utilización como apoyo didáctico en las aulas a todos los niveles educativos; aunque por cuestiones económicas no siempre estén al alcance de todos.

El presente trabajo presenta un nuevo sistema de Realidad Aumentada para la enseñanza de conceptos abstractos en Geometría. El sistema muestra escenas 3D interactivas que pueden ser exploradas y manipuladas por los estudiantes usando como herramientas marcadores físicos. También permite que el alumno resuelva ejercicios preparados por el profesor de manera autónoma, aunque asistido por el sistema.

El sistema puede ser utilizado en diferentes escenarios sin necesidad de hardware especial: en el aula como una herramienta para el profesor, o en casa como una herramienta de estudio para el alumno. El profesor puede crear nuevos ejercicios mediante una herramienta interactiva, que genera las descripciones de las escenas usando un lenguaje basado en XML.

En este trabajo se presentan las características y el diseño de la primera versión del sistema, así como los resultados obtenidos después de pasar encuestas a un grupo de alumnos que ha probado el sistema.

Palabras clave: *Intuición espacial, Realidad Aumentada, Enseñanza de Matemáticas, Geometría.*

Índice general

Agradececimientos	II
Resumen	III
Índice de Tablas	VII
Índice de Figuras	VIII
1. Introducción	1
1.1. Estructura del trabajo.	3
2. Estado del Arte	5
2.1. Marco teórico conceptual.	5
2.1.1. Realidad Aumentada.	5
2.1.2. ARToolKit.	8
2.1.3. Limitaciones del ARToolKit.	9
2.1.4. VRML (Virtual Reality Modeling Language)	10
2.2. Informática en la educación.	11
2.3. Realidad Aumentada en la educación.	13
2.3.1. Clase magistral.	14
2.3.2. Sistemas colaborativos.	16
2.3.3. Autoaprendizaje.	18
2.3.4. Aprendizaje lúdico.	20
3. Aportaciones	24
3.1. Planteamiento del problema.	24
3.2. Justificación.	26

3.3. Objetivo general.	26
3.4. Objetivos específicos.	26
3.5. Problemas a resolver.	27
3.6. Alcances y limitaciones.	28
3.6.1. Alcances.	28
3.6.2. Limitaciones.	28
3.7. El prototipo de una solución.	29
4. Implementación	30
4.1. Descripción general.	30
4.2. Módulos del sistema.	31
4.2.1. OpenGL y ARToolKit.	32
4.2.2. Primitivas.	32
4.2.3. Entrada / Salida.	33
4.2.4. Ejercicios guiados y ejercicios de autoevaluación.	33
4.2.5. Herramienta de descripción de escenas.	34
4.3. Tipos de ejercicios soportados en el sistema.	37
4.4. Conjunto de marcadores.	41
4.5. Archivos de configuración.	43
4.6. Otras características del sistema.	45
4.7. Escenarios de la aplicación.	46
4.8. Ejemplo de un ejercicio.	47
4.9. Aspectos de diseño.	53
4.9.1. Casos de uso.	53
4.9.2. Examinar modelos.	54
4.9.3. Examinar modelos versión extendida.	55
4.9.4. Dibujar primitivas.	56
4.9.5. Dibujar primitivas versión extendida.	57
4.9.6. Eliminar primitivas.	58
4.9.7. Eliminar primitivas versión extendida.	59
4.9.8. Resolver ejercicios.	60
4.9.9. Resolver ejercicios versión extendida.	61
4.9.10. Guardar escena.	62
4.9.11. Guardar escena versión extendida.	63

4.9.12. Escribir escenas.	64
4.9.13. Escribir escena versión extendida.	65
4.9.14. Diagramas de clase.	66
4.9.15. Clase cColor.	67
4.9.16. Clase cPrimitive.	68
4.9.17. Clase cPoint.	69
4.9.18. Clase cVector.	70
4.9.19. Clase cSphere.	71
4.9.20. Clase cCube.	72
4.9.21. Clase cCone.	73
4.9.22. Clase cCylinder.	74
4.9.23. Clase cVRMLObject.	75
4.9.24. Clase cLabel.	76
4.9.25. Clase cCoordinateAxis.	77
4.9.26. Clase cScene.	78
4.9.27. Clase cMultiChEx	79
4.9.28. Clase cDrawingEx.	80
4.9.29. Clase cTransEx.	81
4.9.30. Clase cSceneStore.	82
4.9.31. Clase cGroups.	83
4.9.32. Clase cFileHandling.	84
4.9.33. Estrategia de robustez ante cambios de iluminación.	84
5. Resultados	87
6. Conclusiones	91
7. Apéndice	97

Índice de tablas

2.1. Rango de seguimiento ARToolKit	10
4.1. Descripción de las transiciones del grafo de estados.	86
5.1. Resultados de la encuesta.	88

Índice de figuras

2.1. Continuo de Milgram.	6
2.2. Algoritmo de ARToolKit.	9
2.3. MARIE	15
2.4. Trabajo de Kondo y otros.	15
2.5. RA y Web3D.	16
2.6. Construct3D	17
2.7. ARiSE	18
2.8. CiberMath	19
2.9. Trabajo de Nuñez y otros.	20
2.10. Química aumentada.	20
2.11. Interior del cuerpo humano.	21
2.12. Historias lúdico educativas.	22
2.13. Historias lúdico educativas.	23
4.1. Espacio de trabajo recomendado.	31
4.2. Módulos del sistema.	32
4.3. Herramienta de descripción de escenas.	36
4.4. Ejercicios de opción múltiple.	37
4.5. Ejercicios de dibujo.	38
4.6. Ejercicio de transformaciones.	40
4.7. Herramientas del sistema.	41
4.8. Comparación de materiales.	42
4.9. Cubriendo los marcadores.	43
4.10. Sistema de archivos.	44
4.11. Menú del sistema.	46

4.12. Ejemplo de uso del sistema - Secuencia1.	51
4.13. Ejemplo de uso del sistema - Secuencia2.	52
4.14. Diagrama de casos de uso del sistema.	53
4.15. Diagrama de clases del sistema.	66
4.16. Grafo de estados.	85
5.1. Comparación de características.	90

Capítulo 1

Introducción

A lo largo de su historia el hombre ha buscado maneras eficientes de transmitir el conocimiento valiéndose de las herramientas a su alcance. De esta búsqueda han surgido gracias al ingenio del hombre, por mencionar algunos ejemplos, los códices en las culturas prehispánicas de América o los pergaminos en Asia y, en épocas más recientes la imprenta, que en su momento significó una revolución cultural ya que simbolizó un cambio en la manera de transmitir y almacenar el conocimiento con la impresión de libros a gran escala.

El crecimiento actual de las ciencias computacionales ha permitido desarrollar nuevas herramientas para su utilización en la enseñanza. En las aulas es común contar con proyectores de transparencias para retroproyectar el material que cada profesor ha preparado como ayuda para su clase. De la misma forma cada día son más las aulas que están equipadas con proyectores digitales y ordenadores que permiten el uso de software especializado en presentaciones.

Sin embargo, aún cuando este tipo de tecnología está cambiando la manera de impartir clase, en algunos casos sigue siendo insuficiente para facilitar la comprensión de los conceptos más abstractos.

Específicamente, la enseñanza de las Matemáticas ha representado siempre un reto para los profesionales de la educación. Convertir un concepto abstracto en palabras y ejemplos claros que permitan al alumno asimilarlo no es una tarea fácil, y no siempre se logra. Además, se debe tener en cuenta que no todos los alumnos tienen la misma capacidad para comprender las explicaciones que los

profesores dan de cada tema.

Además de los factores humanos en el proceso de enseñanza, la manera tradicional de impartir las clases de Matemáticas ha sido usando únicamente las pizarras y los modelos 2D dibujados en ellas. Dichos modelos a menudo intentan representar objetos tridimensionales, por ejemplo, vectores en el espacio o volúmenes de sólidos, o conceptos como la suma de dos vectores. En algunos casos se pide al alumno que construya maquetas de los conceptos estudiados en clase. Esto facilita la comprensión del tema, sin embargo exige una inversión de tiempo que se podría utilizar para estudiar conceptos más avanzados.

Es por ello que se deben proponer soluciones que permitan a los profesores contar con las herramientas para ayudar a que los alumnos asimilen mejor los conceptos que por su naturaleza son difíciles de comprender. Dichas soluciones también deberían permitir a los alumnos interactuar con los modelos presentados para generar sus propias conclusiones a partir de la experiencia. En otras palabras, el objetivo de los sistemas de ayuda a la docencia es dotar al alumno de las herramientas para convertirse en un elemento activo en el proceso de aprendizaje.

La Realidad Aumentada (RA) es la rama de la Informática que combina visión por ordenador, reconocimiento de patrones y síntesis de imágenes. Propone una nueva metodología de trabajo que permite gozar de algunos de los beneficios de los entornos virtuales a un coste económico significativamente menor y ganando en movilidad. El objetivo principal de la RA es superponer información útil sobre la percepción que el usuario tiene de la realidad. Dicha información viene codificada típicamente en forma de imágenes generadas por ordenador y permite enriquecer la experiencia del usuario en un entorno real.

Este tipo de sistemas suponen una herramienta novedosa en el proceso de enseñanza, y una ayuda importante para hacerlo más atractivo para el alumno. Un buen ejemplo de esto es el sistema de Shelton y Hedley [27] para la enseñanza de la relación entre la Tierra y el Sol.

Tanto cuentos infantiles [3] como complejos sistemas distribuidos pensados para la educación a distancia [16] utilizan las ventajas de la RA para proporcionar herramientas de trabajo novedosas para facilitar la enseñanza y captar la atención del alumno.

El presente trabajo propone el diseño e implementación de una herramienta que, usando técnicas de Realidad Aumentada, facilite la visualización y manipulación de modelos virtuales para el estudio de algunos conceptos geométricos en un curso de iniciación a la Informática Gráfica.

El sistema permitirá usar primitivas como sistemas de coordenadas tridimensionales, puntos y vectores en el espacio 3D, esferas, conos, cajas, planos, etc. También soportará diferentes operaciones en el espacio tridimensional, como las transformaciones de las primitivas anteriores u otras operaciones.

La idea principal es evitar una representación plana estática (el dibujo en la pizarra) de lo que realmente se encuentra definido en el espacio 3D. Así el alumno podrá ver en el espacio los modelos de las primitivas e interactuar con ellas para analizar el ejercicio desde distintos puntos de vista. Creemos que este proceso permitirá al alumno ganar percepción espacial.

El sistema se ha diseñado para usarse fundamentalmente con una webcam y un ordenador de escritorio o portátil, sin necesidad de instalaciones especiales. Esto permite su utilización tanto en el aula como material didáctico del profesor, como en casa como apoyo para el estudio personal del alumno.

De la misma manera que no hay requisitos de hardware ni instalaciones especiales, para el desarrollo de la herramienta propuesta se han usado librerías de software libre, por lo que la distribución de la herramienta sin fines de lucro se puede realizar sin ningún problema.

1.1. Estructura del trabajo.

La Tesis de Máster está organizada de la siguiente forma. En el capítulo 2 se presentan los antecedentes de este trabajo. Se establece el marco teórico conceptual en el que se sustenta. Se revisan los antecedentes de algunos estudios que se han realizado sobre la utilización de las herramientas informáticas en la educación; específicamente en la formación en Matemáticas. Finalmente se hace una revisión de los principales trabajos que usan técnicas de Realidad Aumentada para la educación.

En el siguiente capítulo, se plantea la problemática que se pretende resolver con el desarrollo de este trabajo. Para ello, se presentan los objetivos que se

persiguen y se plantean los alcances y limitaciones de la investigación.

En el capítulo 4 se describe la implementación de la primera versión del sistema desarrollado. Se describen los módulos que lo forman así como la funcionalidad del sistema y los escenarios en donde se plantea el uso de la herramienta. También se presentan aspectos de diseño de la aplicación: Diagramas de Casos de Uso y Diagramas de Clases.

Los primeros resultados obtenidos con usuarios del sistema se presentan en el capítulo 5. Se hace un análisis de la encuesta de usabilidad del sistema contestada por las personas que han utilizado el sistema. Se hace también una comparación del sistema desarrollado con los sistemas descritos en la sección de antecedentes.

Las conclusiones y planes futuros para el desarrollo de la investigación se presentan en el capítulo 6.

Además de lo anterior, se lista la bibliografía revisada y en la sección de apéndices se anexan las publicaciones fruto de este trabajo.

Capítulo 2

Estado del Arte

En esta sección se hace una revisión de cómo las tecnologías de la información han sido utilizadas en entornos educativos con el objetivo de mejorar el proceso de enseñanza aprendizaje.

Primero se establece el marco teórico conceptual en el que se fundamenta el desarrollo del sistema. Luego se hace una revisión de los aspectos generales de la informática como herramienta educativa y se mencionan dos estudios realizados en este sentido. Después se revisan algunas de las aplicaciones de Realidad Aumentada que persiguen el mismo objetivo.

2.1. Marco teórico conceptual.

2.1.1. Realidad Aumentada.

En los entornos virtuales, más conocidos como entornos de realidad virtual (RV), se introduce al usuario en una escena completamente sintética. Se le aísla de la realidad bloqueando aquellos canales sensoriales en los que se le esté estimulando. El sentido de la vista es el que más comúnmente se estimula en los sistemas actuales.

De acuerdo con Milgram [19], estos sistemas pueden ser clasificados tomando como referencia el grado de inmersión del usuario (ver figura 2.1). Desde los entornos completamente virtuales hasta llegar a aquellos sistemas en los que el único estímulo recibido por el usuario es el real. Al espectro entre un entorno

virtual y un entorno real se le conoce como Realidad Mixta.

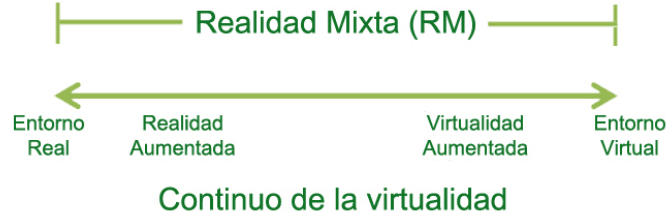


Figura 2.1: Continuo de Milgram.

Continuo de Milgram, una forma de clasificar los sistemas de acuerdo al grado de inmersión del usuario.

Es precisamente dentro de este espectro en donde podemos encontrar los sistemas de Realidad aumentada (RA). En estos sistemas el principal objetivo es añadir a la percepción que el usuario tiene de la realidad imágenes sintéticas. Estas imágenes generadas por ordenador enriquecen y añaden información útil para que el usuario desempeñe sus labores.

De acuerdo con Azuma [1] un sistema de Realidad Aumentada debe cumplir con las siguientes características:

- Combinar información real con información virtual.
- Debe permitir la interacción del usuario con el sistema en tiempo real.
- El proceso de registro de las imágenes sintéticas debe ser en 3D.

De acuerdo con la definición anterior el dispositivo de visualización que el sistema utilice para mostrar las imágenes no impide que sea clasificado dentro de la categoría de RA siempre y cuando se respeten las características antes mencionadas.

De esta manera, el sistema podría utilizar de acuerdo a los requisitos tecnológicos y económicos un dispositivo de los denominados Head Mounted Displays (HMD) o cascos de visualización, un monitor, proyectores digitales o cualquier otro instrumento de visualización.

Los sistemas que utilizan las técnicas de Realidad Aumentada son en su mayoría más económicos que los sistemas virtuales inmersivos. También pueden resultar portables, lo que representa una gran ventaja. Estas dos características

son las que han promovido en gran medida la utilización de las técnicas de RA en la educación.

Existen diferentes librerías que implementan algoritmos de visión por ordenador y reconocimiento de patrones orientados al desarrollo de aplicaciones de RA. En su mayoría toman como referencia patrones para calcular la posición y orientación de la cámara respecto a la escena real. Dichos patrones contienen marcas reconocidas por el sistema, que también permiten identificarlas.

Entre las librerías o herramientas para desarrollar aplicaciones de Realidad Aumentada podemos encontrar:

- DART (Designer's Augmented Reality Toolkit) [17], es un entorno de programación multimedia basado en Macromedia Director. Usa ARToolKit como plugin para llevar a cabo el proceso de localización de los marcadores.
- DWARF (Distributed Wearable Augmented Reality Framework) [18], utilizando XML permite construir sistemas de RA distribuidos. Es un entorno multiplataforma.
- Studierstube [24], es una herramienta basada en OpenInventor. Está pensada para facilitar la elaboración de entornos de RA colaborativos. Utiliza OpenTracker para el seguimiento de los usuarios dentro de la escena. Utiliza el Panel de Interacción Personal para facilitar la interacción con los sistemas de RA.
- ARTag [8], es una librería de RA aumentada. Utiliza patrones impresos para realizar el seguimiento del punto de vista del usuario. No incorpora un módulo nativo para la comunicación del sistema con el dispositivo de adquisición de video, por lo que utiliza los módulos correspondientes de ARToolKit como un plugin.

Es precisamente la librería ARToolKit la que se usará en el presente trabajo. Esta librería es de distribución libre bajo el esquema de "GNU General Public License". En la siguiente subsección se describirán las características principales de esta librería.

2.1.2. ARToolKit.

ARToolKit [13] es una librería de software libre diseñada para construir aplicaciones de Realidad Aumentada.

Utiliza un algoritmo de visión por ordenador basado en binarización para realizar la segmentación de las imágenes obtenidas por el dispositivo de captura de video, seguido de un proceso de detección de esquinas. Este algoritmo realiza el seguimiento de la posición de la cámara con respecto al marcador para determinar el punto de vista a utilizar para registrar los objetos sintéticos con la escena real.

Su funcionamiento se basa en el uso de patrones con marcos rectangulares de color negro. Con la información obtenida de estos marcadores, el algoritmo calcula la posición real de la cámara y su orientación relativa a los marcadores en tiempo real.

De acuerdo a la información proporcionada en la documentación de la misma, su funcionamiento se puede resumir en:

1. Se captura el video de la escena real con una cámara y se envía al ordenador.
2. El algoritmo revisa cada frame en busca de cualquier elemento rectangular.
3. Si se encuentra uno de los marcadores rectangulares, se realizan los cálculos para conocer la posición relativa de la cámara respecto a esos cuadros negros.
4. Se dibujan las primitivas gráficas en la posición que se ha encontrado.
Los gráficos se dibujan sobre el frame analizado, por lo que parecen estar pegados al marcador.
5. Se muestra el resultado en el dispositivo de visualización del sistema.

La figura 2.2 resume el funcionamiento de ARToolKit.

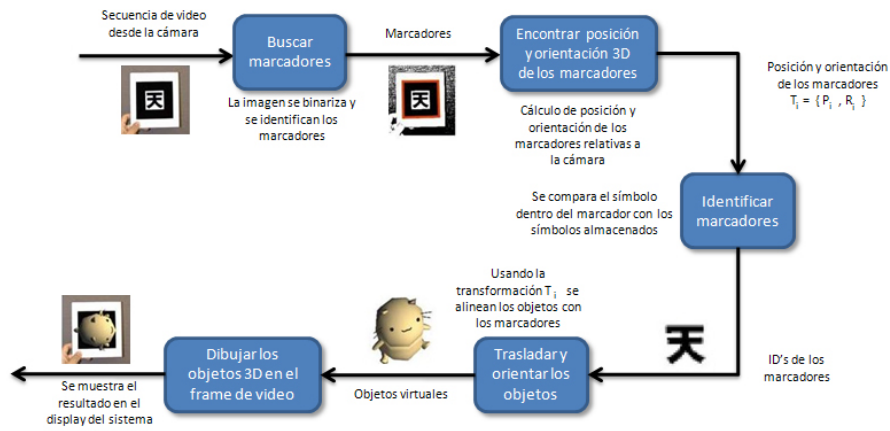


Figura 2.2: Algoritmo de ARToolKit.

Funcionamiento del algoritmo de trabajo de ARToolKit.

2.1.3. Limitaciones del ARToolKit.

Las limitaciones de esta librería son comunes a los sistemas de RA basados en algoritmos de visión por ordenador. Los modelos virtuales aparecerán en el dispositivo de visualización únicamente cuando los marcadores estén visibles. El algoritmo también se ve afectado si el usuario cubre de manera parcial los marcadores, lo que provoca normalmente que los objetos virtuales desaparezcan. Relacionado con el problema de la visibilidad de los marcadores, está el problema de los rangos de visión. Mientras más grande sea el patrón, más alejada podrá estar la cámara para detectarlo.

En la tabla 2.1 se muestra la distancia máxima a la que se ha probado la correcta identificación de los marcadores, para diferentes tamaños. Los resultados mostrados han sido obtenidos de la documentación del ARToolKit. Para llegar a ellos, los autores hicieron pruebas con marcadores de diferentes tamaños, colocándolos de manera perpendicular a la cámara y alejándolos de la misma hasta que los objetos virtuales desaparecían.

Este rango se ve afectado también por la complejidad del patrón a reconocer. Mientras más simples sean los patrones de los marcadores, mejor será la tasa de reconocimiento. Los marcadores con zonas blancas y negras bien definidas tienen mejores resultados.

Tabla 2.1: Rango de seguimiento ARToolKit

Tamaño del patrón (cm.)	Rango de usabilidad (cm.)
6.98	40.64
8.89	63.5
10.79	86.36
18.72	127

Por ejemplo, usando un marcador de 10.79 cm, de una complejidad considerable, la distancia se reduce de 86.36 cm. a solo 38.1 cm.

La orientación de los marcadores respecto a la cámara también afecta a la fiabilidad del algoritmo. Mientras más inclinados se encuentren los marcadores con respecto a la cámara y menos visible sea su centro, peor serán los resultados del proceso de reconocimiento.

Finalmente, las condiciones de iluminación en la escena real afectan de manera notable los resultados del seguimiento de los marcadores. Las luces pueden crear reflejos o brillos indeseados en los marcadores dependiendo de cómo se hayan construido. Los marcadores más habituales se construyen imprimiéndolos sobre papel, lo que dificulta su detección. Para reducir estos brillos, se puede utilizar materiales no reflectantes como el terciopelo.

2.1.4. VRML (Virtual Reality Modeling Language)

Es un estándar creado por el consorcio Web3D. Fue diseñado para la representación de gráficos 3D y su distribución en internet. Es un lenguaje de descripción de escenas y objetos 3D, que pueden ser interactivos y que se almacenan en un archivo de texto. En este archivo se definen los vértices de los objetos y sus propiedades (color, grosor de las líneas, posición, etc.) Con este lenguaje, también se pueden asociar direcciones web a los objetos de la escena, para que el usuario pueda acceder a ellas haciendo clic sobre los objetos.

Para editar los archivos con la descripción de las escenas no hace falta nada más que un editor de textos común, aunque hay muchas herramientas comerciales especializadas en la edición de los mundos virtuales. Para visualizar los modelos virtuales creados con este estándar, existen también un gran número

de navegadores VRML o bien plugins para los navegadores de internet tradicionales.

2.2. Informática en la educación.

Desde hace algunos años los autores se han interesado en evaluar el impacto que tiene el uso de herramientas informáticas para mejorar el proceso educativo. Se ha buscado cuál es la mejor forma de introducir el uso de las herramientas informáticas en el proceso de enseñanza aprendizaje. Se han propuesto diferentes herramientas como material didáctico extra en diferentes áreas de conocimiento, y se han realizado estudios experimentales para validar su aporte.

La idea de que el alumno mejore el proceso de adquisición de conocimiento a partir de su experiencia usando las tecnologías de la información tampoco es nueva. En [21] y en [7] se analiza la utilización de la informática desde el punto de vista constructivista para enseñar matemáticas. En ambos trabajos se destaca la importancia de la interacción del aprendiz con los objetos de estudio, y también con otros sujetos para lograr la construcción de los modelos mentales que generen conocimiento.

Ambos trabajos resaltan la importancia de convertir al alumno en un elemento activo en el proceso de enseñanza aprendizaje. Se habla también de las características ideales de las herramientas informáticas para contribuir eficientemente a la educación. Entre estas características están las siguientes:

- Proporcionar autonomía al alumno para investigar por su cuenta.
- Permitir un ambiente colaborativo entre compañeros.
- Estimular al alumno para que analice su manera de aprender.
- Trabajar en base a metas para que el alumno pueda evaluar su progreso.

Los autores coinciden también en la importancia de facilitar al alumno la relación entre lo conocido y lo que se está aprendiendo. Las herramientas deben ser explícitamente eso, una ayuda que permita al alumno concentrarse y mejorar su aprendizaje.

El objetivo que debe perseguirse de acuerdo a estos dos trabajos es crear un ambiente productivo para que el alumno genere sus propios modelos, de manera

que pueda después extenderlos para resolver nuevos problemas. El profesor debe convertirse entonces en el guía del aprendizaje, orientando el camino que el alumno sigue durante su experimentación.

Además del cambio de rol del profesor, la introducción de este tipo de herramientas como material didáctico significa un esfuerzo extra para ellos. Es necesario planear el tipo de ejemplos que serán utilizados, de manera que realmente contribuyan a mejorar la explicación de los conceptos. De la misma forma hay que cambiar la planificación del curso, para tener en cuenta el tiempo que requiere la utilización de estos sistemas.

En [28] Spalter y otros presentan el análisis de diferentes aproximaciones para introducir material didáctico interactivo a un curso escolar ya existente. De acuerdo con este trabajo, las aproximaciones que mejores resultados dieron son: usar el material didáctico en clase como apoyo para la explicación de algunos conceptos y dar al alumno el material didáctico como ejercicios a resolver en casa. En ambos casos el profesor debe redistribuir el trabajo de la materia para adaptarlo a las nuevas herramientas. También se resalta la importancia de proporcionar al alumno instrucciones sobre lo que se espera que haga con los ejercicios dados, explicaciones del concepto que está aprendiendo y también la manera de cómo este concepto se relaciona con lo ya visto.

Tomando en cuenta lo anterior se consigue despertar el interés de los alumnos por el material didáctico, y mejorar la comprensión de aquellos conceptos que se refuerzan. Sin embargo, es importante considerar que los fallos de hardware o software pueden provocar reacciones adversas y provocar en los alumnos la sensación de falta de control y falta de preparación de la clase y del material.

Otro estudio interesante sobre la contribución de la informática en la educación es el presentado en [5]. En este caso, la herramienta utilizada es una aplicación que permite interactuar con contenido tridimensional usando FogScreen como medio de visualización. La herramienta tiene dos módulos: uno para la generación del contenido en forma de tutoriales y otro para la interacción con este material. Usando un guante o un ratón 3D, el usuario puede manipular los objetos mostrados en la escena (vectores de fuerza, sistemas de coordenadas). El usuario puede manipular el punto de vista de la escena, la posición y orientación de los objetos, adelantar, retroceder o pausar el tutorial permitiendo al alumno

detenerse cuando sea necesario. El sistema también cuenta con diferentes maneras de mostrar al usuario las explicaciones sobre el contenido: texto en pantalla o bien una grabación de voz. El diseño se realizó pensando en que la herramienta se pudiera utilizar por personas sin conocimiento previo en entornos virtuales, lo que se ve reflejado en los resultados de las pruebas de usabilidad realizadas. Las pruebas con el sistema fueron realizadas usando un tutorial acerca de la “Dinámica de rotación de cuerpos rígidos” (Rotational Rigid Body Dynamics). En este tutorial los usuarios interactuaron moviendo los objetos presentados en la escena. A pesar de que algunos usuarios tuvieron dificultades con el uso de los dispositivos de interacción virtual, se adaptaron con éxito después de unos minutos usando el sistema.

Estos trabajos muestran el esfuerzo que se ha hecho por involucrar las nuevas tecnologías en la educación. En algunas ocasiones intentan cambiar de manera radical el concepto que actualmente se tiene del rol del profesor. Algunos de dichos trabajos se han apoyado en los resultados del área de la Realidad Aumentada, en la que se sustenta el presente trabajo.

2.3. Realidad Aumentada en la educación.

Esta sección revisa los trabajos más relevantes en el uso de la Realidad Aumentada aplicada al campo de la educación. Estas aplicaciones pueden ser clasificadas de acuerdo al área de interés en las que se aplican. Así mismo se pueden encontrar aplicaciones monousuario o multiusuario, que funcionan de manera local o distribuida.

Sin embargo, así como estos sistemas comparten ciertas características, existe una clara diferencia en el paradigma utilizado para enseñar. De esta manera se pueden clasificar en cuatro paradigmas diferentes: Clase magistral, Sistemas colaborativos, Autoaprendizaje y Aprendizaje lúdico.

Las Clases magistrales, son el paradigma educativo tradicional. En ellas es el profesor quien lleva el control de la clase. El alumno permanece como receptor pasivo de la información que se le proporciona. En los Sistemas colaborativos se hace énfasis en el trabajo en grupo. Se favorece la interacción entre compañeros, para resolver problemas complejos y generar conocimiento en base a la expe-

riencia colectiva. En el paradigma de Autoaprendizaje, se le da total libertad al alumno para que genere conocimiento en base a lo que investiga. El profesor se convierte en un guía de su aprendizaje, orientándolo. Finalmente, el Aprendizaje lúdico, como su propio nombre lo indica, es aprender jugando. Hacer que los conceptos que se estudian lleguen al alumno a través de actividades divertidas.

2.3.1. Clase magistral.

En este apartado se describen tres ejemplos de sistemas que siguen el paradigma de las Clases magistrales guiadas por un profesor.

El primero de ellos es el Multimedia Augmented Reality for E-Learning (MARIE) [16]. Este sistema fue diseñado como una herramienta para la educación a distancia de ingeniería (ver figura 2.3). El sistema permite al usuario seguir el guión con el contenido del curso, según el diseño del profesor. También permite a los usuarios interactuar con información tanto 2D como 3D y ver al mismo tiempo las explicaciones del profesor. Utilizando un HMD o un monitor de ordenador varios estudiantes pueden interactuar a la vez. También incorpora un módulo de sonido que puede mezclar dos fuentes: la explicación pregrabada y la voz del profesor. Uno de sus inconvenientes es que la interacción con el sistema se basa en la combinación de la manipulación de los marcadores y el uso del teclado y el ratón, lo que puede resultar incómodo para algunos usuarios.

El segundo sistema es el trabajo presentado en [14], que describe tres desarrollos aplicados a la educación que hacen uso de la RA. El primer ejemplo es un libro de texto enriquecido. En este sistema se superponen gráficos 3D sobre un libro de texto tradicional para mejorar la explicación del concepto. Para ello utilizan una cámara web para identificar el patrón impreso sobre la hoja del libro (ver figura 2.4(a)).

El segundo sistema se dedica a la enseñanza de Matemáticas en el entorno de un aula. El profesor da explicaciones sobre un modelo que los alumnos pueden ver en pantalla. Utilizando una cámara web y los marcadores impresos y colocados sobre su escritorio, el profesor puede interactuar con los modelos virtuales de sólidos como si se tratara de modelos reales (ver figura 2.4(b)). Sin embargo no está diseñado para que el alumno pueda usar el sistema de manera independiente ni incluye ejercicios que puedan ser resueltos.

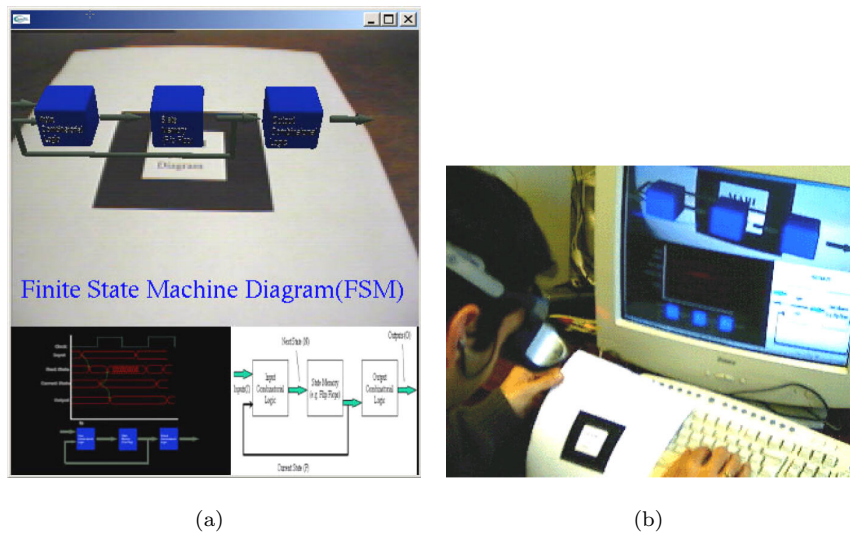


Figura 2.3: MARIE

Izquierda, método de interacción con el sistema; *Derecha*, interfaz de la aplicación.

El último sistema permite enriquecer el esqueleto de dinosaurios con información gráfica sobre cómo podría ser su cuerpo. Además, también permite a los usuarios experimentar con algunos aspectos que todavía están en estudio, como puede ser el color de la piel de algunas especies (ver figura 2.4(c)).

El artículo también presenta el desarrollo de un sistema de autor para preparar esta clase de contenido disminuyendo la necesidad de conocimientos previos sobre modelado 3D.

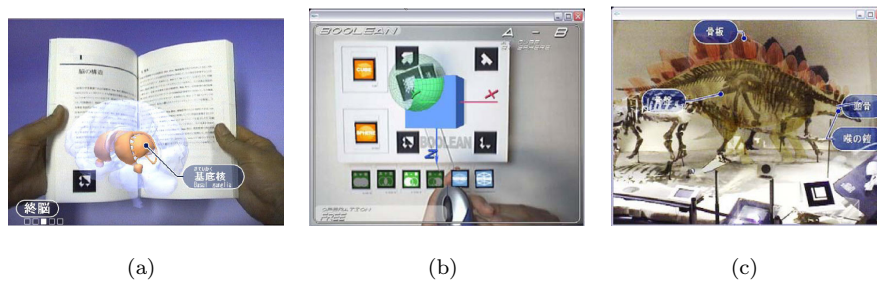


Figura 2.4: Trabajo de Kondo y otros.

RA aplicada a la educación, (a) un libro de texto aumentado, (b) aplicación para enseñar matemáticas en el aula y, (c) añadiendo información al esqueleto de un estegosaurio.

El sistema descrito en [15] combina Web3D y la RA para construir un sistema de ayuda a la enseñanza de conceptos de ingeniería mecánica (ver figura 2.5). Web3D es un conjunto de aplicaciones que implementan el estándar X3D para transmisión de escenas y de objetos 3D a través de internet usando XML. Normalmente se usan plugins instalados en un navegador web para visualizar estos modelos. Web3D permite al usuario interactuar con modelos 2D y 3D que pueden estar animados, y puede seleccionar qué contenido le resulta más interesante para seguir estudiando. El módulo de RA está embebido en el navegador y funciona como una aplicación estándar de RA y hace uso de la biblioteca Microsoft Foundation Class (MFC) para crear su interfaz de usuario. El sistema permite usar modelos almacenados en formatos estándar. Este sistema se puede utilizar a través de Internet o de manera local durante una clase o en el laboratorio.

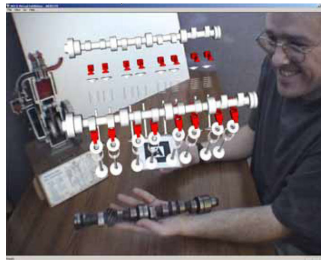


Figura 2.5: RA y Web3D.

Ejemplo de uso de la aplicación desarrollada combinando Web3D y Realidad Aumentada.

2.3.2. Sistemas colaborativos.

En esta sección se presentan dos sistemas que utilizan un entorno multiusuario colaborativo para que los alumnos desarrollen diferentes tareas. El primero de estos sistemas y quizás uno de los más importantes es Construct3D [26], un sistema de Realidad Aumentada construido sobre la plataforma colaborativa Studierstube [25]. Este sistema permite a varios usuarios provistos de cascos HMD trabajar concurrentemente sobre la misma tarea (ver figura 2.6). Según los autores, Construct3D no es una herramienta de diseño 3D profesional. El objetivo del trabajo era crear un interfaz de usuario amigable utilizando el Per-

sonal Interaction Panel (PIP) [29]. Este panel permite al usuario interactuar con objetos virtuales a través de una interfaz tangible y de un menú en el que puede seleccionar opciones y ejecutar acciones. Esta plataforma se diseñó originalmente para la enseñanza de geometría. Construct3D necesita un espacio dedicado para instalar el equipo, y localiza el origen del sistema de coordenadas virtual en el centro de la habitación. El sistema puede trabajar con líneas, puntos, planos y algunos volúmenes sólidos como esferas y conos. También es posible guardar las escenas en ficheros y tiene un subsistema de audio que asiste al usuario en el desarrollo del ejercicio. Por medio de encuestas, los autores encontraron que algunos usuarios preferían los sistemas CAD de escritorio tradicionales. Sin embargo, los mismos usuarios valoraban la facilidad de uso de Construct3D y lo encontraban útil como herramienta de aprendizaje.

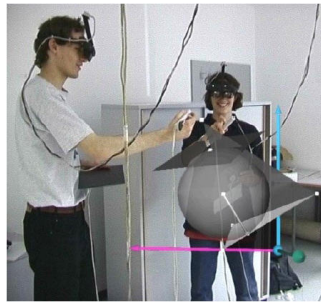


Figura 2.6: Construct3D

Construct3D, un sistema de realidad aumentada que favorece el entorno colaborativo entre usuarios.

ARiSE (Augmented Reality in School Environments) [4] es una plataforma construida con el esfuerzo conjunto de varias universidades europeas, y su objetivo es facilitar la transmisión de conocimientos al estudiante. El sistema permite a varios estudiantes interactuar y trabajar en equipo. También incluye las herramientas necesarias para que un usuario no experto en RA pueda usarlo y generar contenido fácilmente. Los resultados de las pruebas de usabilidad fueron descritos en [2]. El objetivo de dichas pruebas fue evaluar la usabilidad del sistema y la calidad pedagógica del material presentado por el sistema. Entendemos por usabilidad de un sistema lo fácil que resulta de usar y de aprender a usarlo. Por calidad pedagógica entendemos en qué medida el

sistema ayuda a mejorar el proceso de aprendizaje. De forma parecida a otros estudios de este tipo, se encontró que la novedad de esta tecnología estimula al estudiante a utilizar la herramienta, encontrándola divertida y motivante. Sin embargo, dichas pruebas revelaron que los usuarios esperaban algo más del sistema, especialmente con respecto a la retroalimentación cuando se ejecuta una acción. El usuario esperaba que de alguna manera el sistema le informara si la acción desarrollada cumplió su objetivo o no. Los tests de usabilidad mostraron también la necesidad de mejorar la interacción con el usuario, ya que hubo problemas de precisión usando la herramienta. Cabe mencionar que ninguno de los alumnos que participó en las encuestas tenía conocimientos previos de RA.

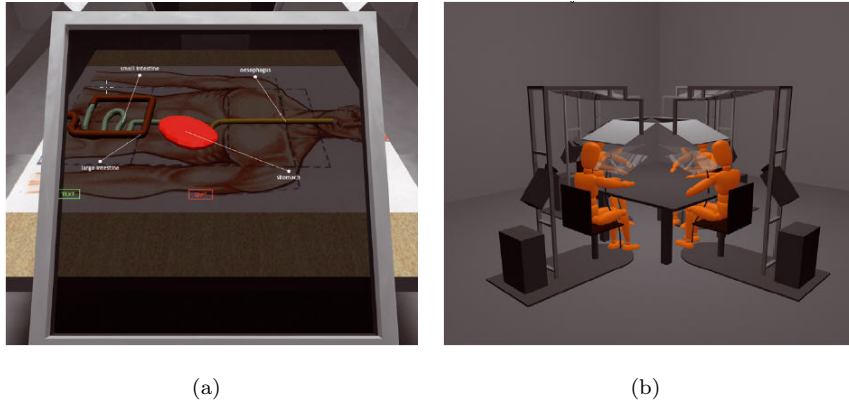


Figura 2.7: ARiSE

(a) uso del sistema en una clase de anatomía, (b), prototipo del dispositivo que usará la aplicación como medio de visualización.

2.3.3. Autoaprendizaje.

De acuerdo con los autores, CyberMath [30] soporta diferentes paradigmas educativos. Sin embargo, su diseño se inspiró en un museo, en el que cada sala presenta un tema distinto. Así, el alumno puede repasar cada tema sin la ayuda de un profesor. Es un entorno virtual basado en la plataforma DIVE [6], que se puede usar en un entorno inmersivo tipo CAVE o con un HMD. Permite trabajar a varios usuarios a la vez replicando el contenido en cada máquina cliente, conectadas a través de Internet. También dispone de aulas, en las que avatares simulan la presencia de los usuarios. El uso de gráficos realistas permite

que la aplicación sea más atractiva para el usuario (ver figura 2.8). El sistema permite importar modelos o animaciones construidos en Mathematica, lo cual le da flexibilidad de uso. Como resultado de las pruebas de usabilidad del sistema, se observó que en algunas ocasiones el uso de los avatares obstruía la visibilidad del usuario.

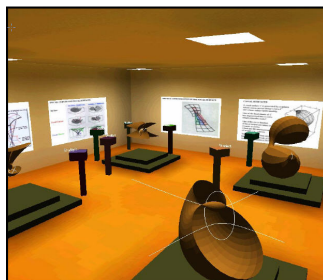


Figura 2.8: CiberMath

CiberMath, una aplicación con gran calidad en su interfaz gráfica, diseñada siguiendo el patrón de un museo.

Otra aplicación que hace uso de técnicas de RA en la educación con un enfoque de autoaprendizaje es la presentada por Núñez y otros [20]. En este sistema se presentan modelos tridimensionales de compuestos inorgánicos descritos en VRML. El sistema usa un PC normal con seis cámaras USB conectadas mediante un concentrador para permitir el trabajo simultáneo de varios usuarios, de manera individual o por pequeños grupos (ver figura 2.9). La secuencia de video obtenida por cada cámara se reproduce en una ventana independiente, para permitir que todos los alumnos puedan observar el trabajo de sus compañeros. Para ello se utiliza el proyector del aula. Los resultados obtenidos tras varias pruebas en la Universidad Jaume I de Castellón, España, muestran que los alumnos encuentran el sistema muy interesante y permite captar su atención e incrementar su motivación.

En [9] se presenta la evaluación de dos versiones de un sistema de “Química aumentada”. Se ha diseñado para interactuar y componer modelos tridimensionales de moléculas orgánicas a partir de modelos almacenados en una base de datos. El sistema permite crear moléculas usando herramientas representadas mediante marcadores 2D, imitando el modelo de bolas y varillas usado tradicionalmente en la enseñanza de Química (ver figura 2.10). El sistema se probó con

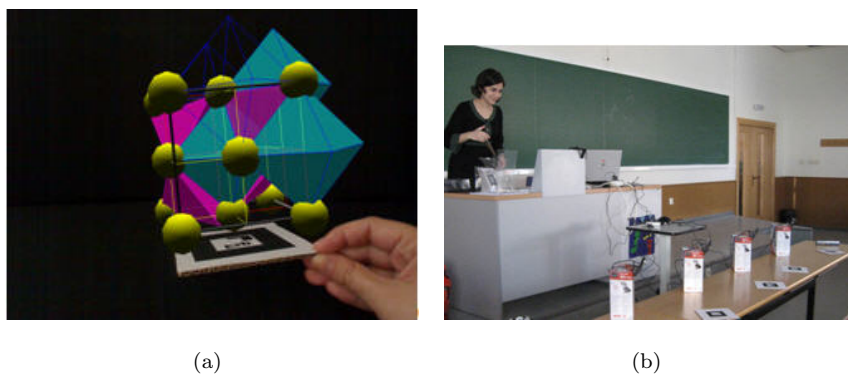


Figura 2.9: Trabajo de Nuñez y otros.

(a), Ejemplo de uno de los compuestos inorgánicos mostrados con la aplicación; (b), Propuesta de configuración propuesta del espacio de trabajo.

estudiantes de secundaria, que en su mayoría lo encontraron atractivo y fácil de usar a pesar de que interactuar con el sistema resultaba más complejo que con el modelo de bolas y varillas tradicional.

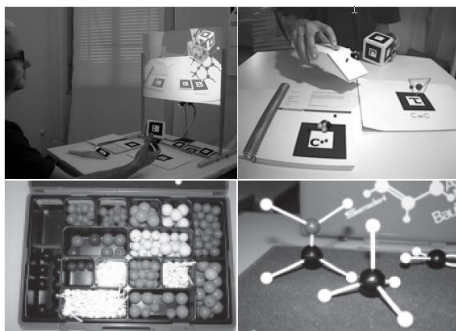


Figura 2.10: Química aumentada.

Augmented Chemistry, comparación del modelo BSB y la analogía usando RA en el sistema.

2.3.4. Aprendizaje lúdico.

Otros sistemas educativos que utilizan la RA son los que se han desarrollado dentro de nuestro grupo de investigación. Uno de ellos es el Sistema de Realidad Aumentada para conocer el interior del cuerpo humano [10]. El sistema utiliza interfaces tangibles para que el usuario pueda aprender la estructura anatómica

del interior del cuerpo humano. El maniquí utilizado consiste en un cuadro de madera sobre el que se ubican los marcadores y una tela a la que se le hicieron cortes que pueden ser abiertos y cerrados con cremalleras (ver figura 2.11). De esta manera el usuario utilizando un HMD puede al abrir el cierre asomarse al interior del cuerpo, observando en la parte baja del abdomen la zona de los intestinos, o bien en la parte alta, la zona del estómago. En las pruebas realizadas durante la Escuela de Verano de la UPV, participaron 40 niños entre 8 y 10 años, que encontraron el sistema fácil de usar y estimulante para su aprendizaje, y demostraron interés por utilizarlo en otras áreas de conocimiento.

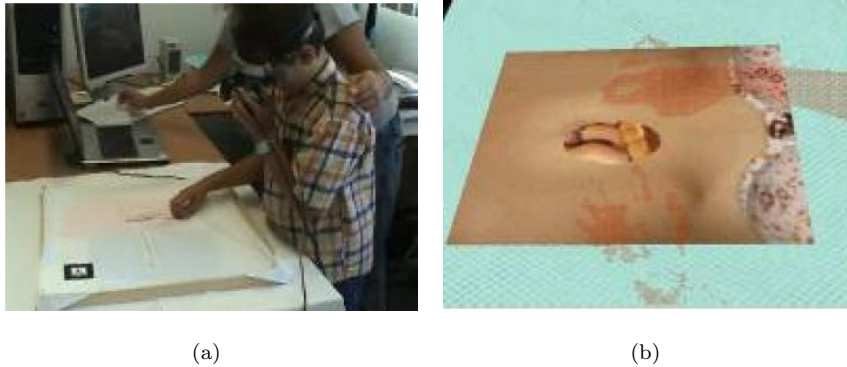


Figura 2.11: Interior del cuerpo humano.

(a) interfaz física del sistema, (b), la imagen que el usuario puede ver del cuerpo humano.

Otros sistemas son los presentados en [11]. En este trabajo se proponen aplicaciones de RA para niños en las que se cuentan historias lúdico-educativas. Estas aplicaciones utilizan como interfaz un cubo sobre el cual se proyectan los videos (ver figura 2.12). En el primer sistema, la historia se proyecta sobre cualquiera de las seis caras del cubo, y el usuario puede decidir el progreso de la historia usando otros dos cubos independientes que identifican las dos posibles opciones. En el segundo sistema, la historia se proyecta sobre una de las caras del cubo principal, y las herramientas de selección están unidas a él, de manera que el usuario tiene que levantar cualquiera de ellas para elegir cómo se desarrolla la historia. En la evaluación de este sistema participaron 44 niños de entre 6 y 8 años durante la Escuela de verano de la UPV. Los resultados de dicha evaluación muestran que el uso de un dispositivo HMD o de un monitor

como medio de visualización del sistema, no influye en el impacto positivo que los sistemas tienen sobre los niños.

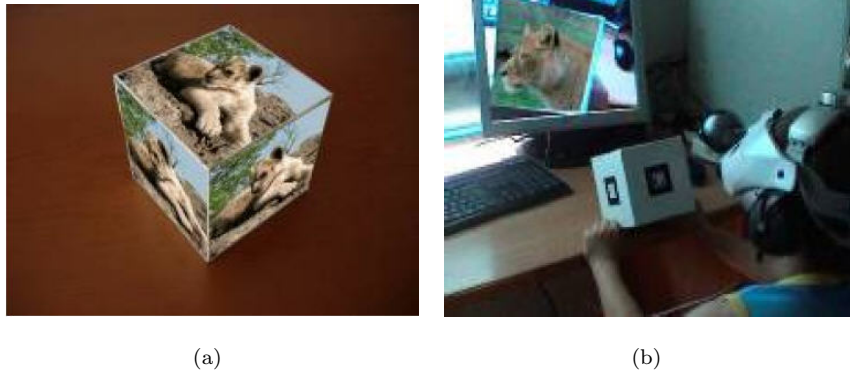


Figura 2.12: Historias lúdico educativas.

(a) la historia aparece proyectada en todas las caras del cubo, (b), usando un HMD, el usuario puede ver sobre el cubo que sostiene, la historia que se le está contando.

Finalmente, en [12] se presentan tres juegos de RA enfocados a la concienciación de los niños en temas de conservación del medio ambiente. En las pruebas de estos tres juegos participaron 160 niños de la Escuela de verano de la UPV. En el primer juego Magnetic Cubes el usuario debe encontrar alguno de los animales en peligro de extinción que contempla el sistema, y si acierta, puede elegir si desea ver un video con información sobre dicho animal (ver figura 2.13(a)). Usa como interfaz tres cubos que se mantienen unidos uno a otro magnéticamente. A través de la combinación de los marcadores colocados en las caras visibles de los cubos el usuario realiza las acciones necesarias para interactuar con el sistema. El siguiente juego denominado Finding pairs, es el clásico juego de memoria, en donde el usuario debe encontrar la tarjeta que completa el par con el animal en peligro de extinción mostrado. Si el usuario acierta, tiene la posibilidad de ver un video con información relevante sobre la especie que ha encontrado (ver figura 2.13(b)). El tercer juego, fue desarrollado para funcionar en un teléfono móvil, y en él el usuario debe identificar la imagen que aparece sobre el marcador, para de esa manera colocarlo en el contenedor que corresponda a la categoría de reciclado adecuada (ver figura 2.13(c)). En total se usan 5 marcadores diferentes, uno para cada contenedor de basura, y

uno sobre el que se muestra el objeto a clasificar.

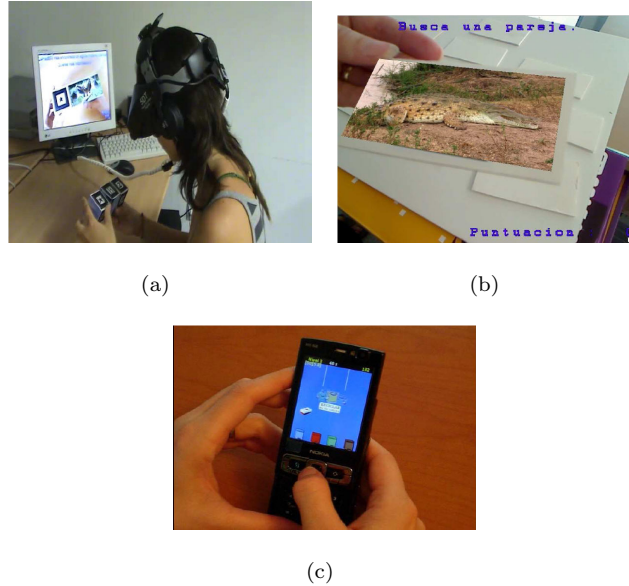


Figura 2.13: Historias lúdico educativas.

En esta figura se pueden ver los tres sistemas presentados en el trabajo de [12]; (a) Magnetic Cubes, (b) Finding Pairs, (c) Recycling.

Como comentario final a esta sección se puede decir que todos los sistemas presentados son muy completos. Sin embargo, en algunos casos, el coste de la infraestructura del sistema los hace inalcanzables para un presupuesto reducido. Otro problema importante es que en algunos de estos sistemas el alumno permanece como espectador en el proceso de aprendizaje. Con el diseño de la aplicación que presentamos queremos afrontar estos problemas. Damos al alumno buena parte del protagonismo en el proceso del aprendizaje. Por otra parte, el profesor dispone de una herramienta versátil para mejorar la calidad de sus explicaciones y evaluar al alumno. Y sobre todo, pensamos en un diseño sencillo que permita implantar el sistema sin invertir una fuerte cantidad de dinero.

Capítulo 3

Aportaciones

En la sección anterior se hizo una revisión del estado del arte en la aplicación de las tecnologías de la información a la enseñanza. También se revisaron los sistemas que utilizan las técnicas de Realidad Aumentada con objetivos pedagógicos en diferentes áreas del conocimiento.

En esta sección se describen los objetivos que se persiguen en este trabajo, se hace una descripción de la problemática que se intenta resolver. Y se definen los alcances y limitaciones del proyecto. También se hará una breve descripción de las características generales del sistema que se ha desarrollado, resaltando como se han alcanzado los objetivos planteados.

3.1. Planteamiento del problema.

Brindar una educación de calidad es uno de los principales objetivos que persigue toda institución educativa a cualquier nivel del que hablemos; para ello, se debe contar con el personal capacitado y las mejores herramientas disponibles para lograr que los alumnos obtengan una mejor formación.

Va siendo cada vez más común, que de acuerdo al presupuesto con el que cuenta la escuela, haya más aulas equipadas con proyectores digitales y ordenadores. Estas herramientas buscan brindar tanto a los alumnos como a los profesores una herramienta versátil para mejorar la forma de impartir las clases.

Sin embargo este tipo de herramientas generalmente se utilizan para presen-

taciones de diapositivas, siendo aún poco el material interactivo disponible en algunas de las asignaturas como Geografía, Física, Química o Matemáticas.

Si hablamos específicamente de la enseñanza de las Matemáticas, esto ha representado siempre un gran reto para los profesionales de la educación, puesto que es difícil traducir en palabras claras y sencillas un concepto abstracto para que la explicación sea asimilada por la mayor cantidad de alumnos que sea posible.

Muy relacionado con problema de “simplificar” las explicaciones está la diversidad de capacidades que existe entre los individuos que forman parte de un grupo. Por ello hay que buscar el compromiso entre la complejidad de los temas vistos y el tiempo que se le dedica, ya que cuando un estudiante no entiende un concepto importante a la larga se convierte en una herramienta que no podrá utilizar.

Otro de los problemas es que, por lo general, los alumnos sienten aversión por las Matemáticas. Así que otro de los retos a vencer al tratar de desarrollar una herramienta educativa para la enseñanza de las Matemáticas es lograr crearla de manera que sea “atractiva” y fácil de usar para el usuario final y que motive al alumno en su aprendizaje.

Las herramientas con las que actualmente en la mayoría de las instituciones educativas se imparte Matemáticas no van más allá de la pizarra, reglas, escuadras, transportadores, compás, calculadora, etc. Dichos recursos pueden resultar limitados cuando en una pizarra plana se hacen las proyecciones correspondientes, por ejemplo, para tratar de dibujar un vector en el espacio tridimensional. Existe una parte del alumnado al que le cuesta visualizar espacialmente dichas primitivas, y aún más entender dentro del espacio algunas operaciones como la suma de vectores.

Existen también herramientas computacionales que son en su mayoría aplicaciones en las que los gráficos son en 2D, por ejemplo hechas en Java para que se ejecuten en un entorno web [28]. En mucha menor medida, hay otras aplicaciones que se desarrollan en un entorno 3D [26], pero que pueden resultar prohibitivas para el presupuesto asignado a la mayoría de las escuelas. Este tipo de herramientas necesitan hardware especializado y espacios específicamente diseñados para su instalación, por lo que solo pueden ser usados dentro de la

escuela y no al mismo tiempo por varios grupos con temas diferentes.

Sin embargo, aún cuando este tipo de tecnología está cambiando la manera de dar clase, en algunos casos sigue siendo insuficiente para facilitar la comprensión de los conceptos más abstractos, por ejemplo, las transformaciones geométricas. En otros casos, el coste de la inversión necesaria para implementar el sistema es muy caro, o bien, se sigue dejando al alumno como un espectador durante el proceso de enseñanza - aprendizaje.

3.2. Justificación.

Para resolver los problemas expuestos, surge la necesidad de desarrollar una herramienta de bajo coste, en la que el alumno pueda experimentar para aprender. Dicha herramienta debe resultar atractiva y estimulante para el estudiante, y ser flexible y fácil de usar, igual para profesores y alumnos. Y sobre todo que pueda ser usada tanto en el aula como en el lugar de estudio de los alumnos: bibliotecas, laboratorios o en el hogar.

3.3. Objetivo general.

El objetivo principal de este trabajo es desarrollar una aplicación de Realidad Aumentada orientada a la enseñanza de conceptos abstractos en Geometría, que permita mostrar al alumno escenas 3D interactivas que puedan ser exploradas desde multitud de puntos de vista y que le ayuden a mejorar su intuición espacial.

3.4. Objetivos específicos.

A continuación se enumeran más detalladamente los objetivos del trabajo:

- Lograr un sistema de RA estable que pueda ser utilizado en diferentes escenarios: en clase como herramienta didáctica para el profesor y en casa o en el laboratorio por parte del alumno como herramienta de estudio.
- Mantener los requisitos de software y hardware al mínimo, para que la introducción del sistema no represente una inversión económica fuerte para

las instituciones educativas.

- Desarrollar las primitivas de dibujo necesarias para que en el sistema puedan crearse modelos 3D interactivos.
- Desarrollar los módulos necesarios para permitir al usuario interactuar con las primitivas.
- Desarrollar los módulos necesarios para que el sistema reconozca la descripción de las escenas dadas por el usuario en un archivo.
- Desarrollar el módulo de autoevaluación para el alumno con ejercicios dados por el profesor.
- Usar ARToolKit como librería base para el proceso de registro de los objetos virtuales en la escena real con respecto a la posición de los marcadores.

3.5. Problemas a resolver.

Los problemas que se pretende resolver con los objetivos anteriores son los siguientes:

- La dificultad de los alumnos para obtener una buena percepción espacial de los conceptos geométricos tridimensionales con las herramientas didácticas tradicionales.
- La elevada inversión económica de los sistemas virtuales tradicionales es prohibitiva para la mayoría de los presupuestos escolares.
- La mayoría de los entornos virtuales completos no pueden ser llevados al aula, ni mucho menos al lugar de estudio del alumno.
- El alumno permanece como espectador durante su aprendizaje, por lo que no puede construir los conceptos a partir de su experiencia.

3.6. Alcances y limitaciones.

3.6.1. Alcances.

En la primera versión del sistema se han planteado los siguientes hitos:

- Se desarrollará un prototipo completo y funcional del sistema.
- Se contemplarán las siguientes primitivas de dibujo: Sistemas de coordenadas tridimensionales, puntos, vectores, esferas, cubos, cilindros, conos, etiquetas y modelos VRML.
- Se contemplarán las siguientes acciones dentro del sistema: dibujar primitivas en la escena, borrar primitivas de la escena, reiniciar la escena a su estado original, guardar la escena en un archivo, sumar vectores y aplicar las transformaciones geométricas básicas (rotación, traslación y escalado).
- Se desarrollará el módulo para convertir desde una descripción dada por el usuario de las escenas en un formato similar a XML al grafo de escena de la aplicación y viceversa.
- Se desarrollará un módulo básico de autoevaluación en donde el profesor podrá definir tres tipos de ejercicios para valorar el avance de los alumnos: Ejercicios de dibujo; Ejercicios de opción múltiple y Ejercicios de Transformaciones (véase la sección 4.3).

3.6.2. Limitaciones.

El sistema en su diseño actual tendrá las siguientes limitaciones:

- No se modificará el algoritmo de seguimiento del ARToolKit, por lo que se heredarán las limitaciones de esta librería.
- No se realizarán pruebas de usabilidad exhaustivas del sistema en esta etapa de la investigación.

Estas limitaciones podrán ser resueltas en futuras versiones del sistema, a medida que avance la investigación y el desarrollo del mismo.

3.7. El prototipo de una solución.

La solución que se presenta en este trabajo es un sistema de Realidad Aumentada para enseñar geometría en el espacio 3D. Utilizando las técnicas de RA, se pretende lograr que el alumno pueda obtener una mejor intuición espacial de los conceptos explicados. Se plantea el uso de modelos tridimensionales vistos a través de un monitor común, en lugar de modelos 3D dibujados sobre una pizarra plana.

Además, por el diseño que se ha seguido, la herramienta desarrollada es de bajo coste, y con requisitos de hardware reducidos: únicamente requiere un ordenador actual y una webcam. El resto de los componentes del sistema pueden ser fácilmente contruidos con materiales comunes.

Los marcadores del sistema pueden ser impresos y las varitas utilizadas para su manejo pueden ser contruidas con cartón o con listones de madera. Para el profesor, el sistema se convierte en una herramienta que le permite, demostrar algunos conceptos de manera más eficiente, puesto que en lugar de dedicar tiempo dibujando modelos 3D proyectados en la pizarra plana, ahora puede dedicarse a clarificarlos mejor, utilizando como referencia los modelos del sistema. Para ello el sistema puede ser utilizado con un video proyector. El siguiente capítulo presenta los detalles de implementación del sistema, así como una descripción completa de sus características.

Capítulo 4

Implementación

El sistema de Realidad Aumentada de bajo coste que aquí se detalla tiene como principal objetivo facilitar el entrenamiento de la percepción espacial de los estudiantes. En este sistema, el alumno se convierte en un sujeto activo durante su aprendizaje.

4.1. Descripción general.

El sistema únicamente necesita un ordenador de tipo PC y una webcam. La webcam permite capturar el vídeo sobre el que se mostrará la escena, que se dibujará alineada con los marcadores utilizados como fondo (véase por ejemplo, la figura 4.1). Para resolver el tracking de la cámara se ha usado la librería ARToolKit [13].

La figura 4.1 muestra la configuración recomendada para el entorno de trabajo del sistema: un PC, una webcam y varios marcadores que pueden ser contruidos fácilmente por los alumnos, a partir de materiales comunes.

El sistema permite construir escenas usando cualquiera de las siguientes primitivas: sistemas de coordenadas tridimensionales, puntos, vectores, esferas, cubos, conos, cilindros y objetos complejos descritos con el estándar VRML. El sistema también permite definir grupos de primitivas sobre los que aplicar acciones.

Dichas acciones pueden ser: situar un objeto en la escena editando el archivo de configuración de la escena, dibujar primitivas usando la herramienta de di-

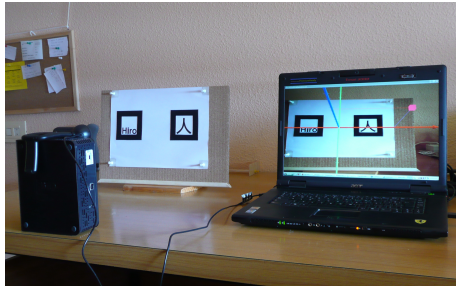


Figura 4.1: Espacio de trabajo recomendado.

Configuración recomendada del espacio de trabajo para nuestro sistema

bujo del sistema, aplicar las transformaciones básicas (traslaciones, rotaciones y escalados), seleccionar entre varias opciones para responder las preguntas planteadas, reiniciar el trabajo con la escena eliminando las primitivas dibujadas y guardar el trabajo realizado para utilizarlo en una sesión posterior. Cada una de las acciones se puede asociar a un marcador distinto, que el usuario deberá seleccionar y mover dentro de la escena. Por supuesto, el usuario podrá siempre cambiar el punto de vista de la escena moviendo la cámara.

4.2. Módulos del sistema.

El sistema de Realidad Aumentada está compuesto por siete módulos independientes (ver figura 4.2). Estos módulos son: la librería de ARToolKit, el módulo de definición de primitivas, el módulo de entrada/salida, los ejercicios guiados, los ejercicios de autoevaluación y la herramienta de descripción de escenas. Toda la salida gráfica del sistema se basa en el módulo correspondiente a OpenGL.

En la figura 4.2, cada una de las flechas que unen los módulos del sistema representa una relación de uso. Por ejemplo, la flecha que une el módulo de la Herramienta de descripción de escenas con los de Ejercicios guiados y Autoevaluación, indica que el primero usa a los otros dos. A continuación se presenta la descripción de cada uno de los módulos que forman parte del sistema.

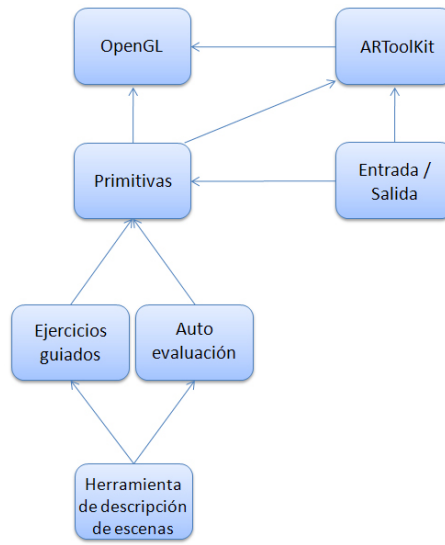


Figura 4.2: Módulos del sistema.

4.2.1. OpenGL y ARToolKit.

Estos dos módulos forman los cimientos de la aplicación desarrollada. Las primitivas de dibujo así como la interfaz gráfica han sido desarrolladas usando OpenGL.

La librería ARToolKit, ha servido como la base para las funciones de Realidad Aumentada del sistema. Esta librería proporciona las herramientas necesarias para generar los marcadores que se usarán en el sistema. Implementa además el algoritmo de seguimiento de estos marcadores, calculando una matriz de transformación con la posición y orientación relativa de esos marcadores con respecto a la cámara. ARToolKit también proporciona una interfaz transparente para comunicar el sistema con la cámara y las clases necesarias para cargar modelos 3D almacenados con el formato del estándar VRML.

4.2.2. Primitivas.

Este módulo es el encargado de administrar la generación del grafo de escena, con todas sus acciones válidas. En este módulo se encuentran las clases que definen las diferentes primitivas soportadas por el sistema.

Estas primitivas son: Sistemas de Coordenadas, Etiquetas, Puntos, Vectores,

Esferas, Cubos, Cilindros, Conos y objetos complejos almacenados en formato VRML (de ahora en adelante se hará referencia a ellos como objetos VRML). Dentro del grafo de escena se pueden definir grupos de primitivas, a los que se les puede aplicar las acciones del sistema.

Las acciones soportadas por la versión actual del sistema son: dibujar primitivas, borrar primitivas, reiniciar la escena a su estado original, sumar vectores, aplicar transformaciones afines (rotación, traslación y escalado) y guardar el estado actual de una escena o ejercicio en un archivo.

Además, el profesor podrá elegir en el momento de crear la descripción de la escena si el resultado de aplicar una transformación sobre las primitivas se muestra con una animación. La animación mostrará cuadro a cuadro el cambio de posición de la primitiva. Así el alumno podrá entender mejor el funcionamiento de las transformaciones.

4.2.3. Entrada / Salida.

Este módulo es el encargado de cargar las escenas almacenadas en disco y construir el grafo de escena correspondiente. El formato del archivo de escenas es similar a XML. Este módulo también realiza el proceso inverso codificando de nuevo el contenido del grafo de escena para poder almacenarlo en un archivo de texto.

Dentro de los archivos de descripción de escenas el usuario puede codificar uno o más ejercicios, definiendo la forma de agrupar las primitivas, y también las acciones permitidas.

4.2.4. Ejercicios guiados y ejercicios de autoevaluación.

En estos módulos se encuentran las clases que implementan los métodos necesarios para manipular el grafo de escena. Se implementa el algoritmo de selección de escenas, que decide que grafo se muestra en pantalla, dependiendo del par de marcadores que se esté mostrando ante la cámara.

En estos módulos se implementa también un algoritmo para dar estabilidad a los modelos 3D, tratando de reducir al mínimo la influencia de los cambios de iluminación sobre la escena. Este algoritmo se basa en la premisa de que mientras el usuario interactúa con el sistema, modificando las propiedades de la escena,

no cambiará el punto de vista. Así el algoritmo fija la matriz de transformación del ARToolKit que indica la posición y orientación relativa de los marcadores respecto a la cámara, consiguiendo estabilizar por completo la escena.

4.2.5. Herramienta de descripción de escenas.

Las escenas se almacenan en archivos de texto y están descritas en un lenguaje similar a XML. Para facilitar la creación de estos archivos, se ha creado un módulo independiente que implementa una herramienta de construcción de escenas con una interfaz gráfica amigable, evitando que el usuario tenga que aprender el lenguaje de descripción de escenas (ver la figura 4.3).

La herramienta permite la descripción de las escenas, la configuración de los grupos de primitivas y las luces de las escenas. Las propiedades de estos elementos, como el color de la luz, también se pueden configurar fácilmente. Las acciones permitidas por cada grupo de primitivas se definen también con esta herramienta.

Entre las propiedades de las escenas que se configuran con la herramienta está la asociación de éstas con los marcadores que la identificarán. Finalmente, en un mismo archivo puede haber descritos varios ejercicios guiados y ejercicios de autoevaluación.

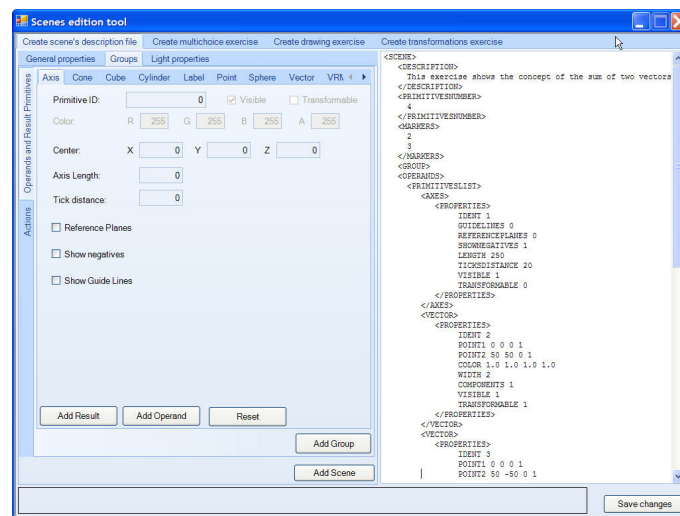
Los ejercicios guiados, son aquellos ejercicios en los que el profesor dará una explicación frente al grupo. En este tipo de ejercicios, el sistema no ofrece la posibilidad de resolver los problemas planteados. Por ejemplo, si en la escena se describe un problema de suma de vectores, el profesor deberá completar el ejercicio correctamente con las herramientas de dibujo, o bien, deberá describir el ejercicio completo en el archivo correspondiente.

Los ejercicios de autoevaluación están pensados para que el profesor los deje como actividades a los alumnos. Cada uno de estos ejercicios debe llevar en su descripción las instrucciones de las actividades que se han de realizar. Para este tipo de ejercicios el sistema ofrece la posibilidad de resolver los problemas planteados (ver sección 4.3 para una descripción de los ejercicios soportados por el sistema). Estos ejercicios incluyen también una descripción de la respuesta correcta, que será mostrada al alumno cuando le solicite al sistema mostrar la solución. La explicación de la respuesta también aparecerá cuando el alumno

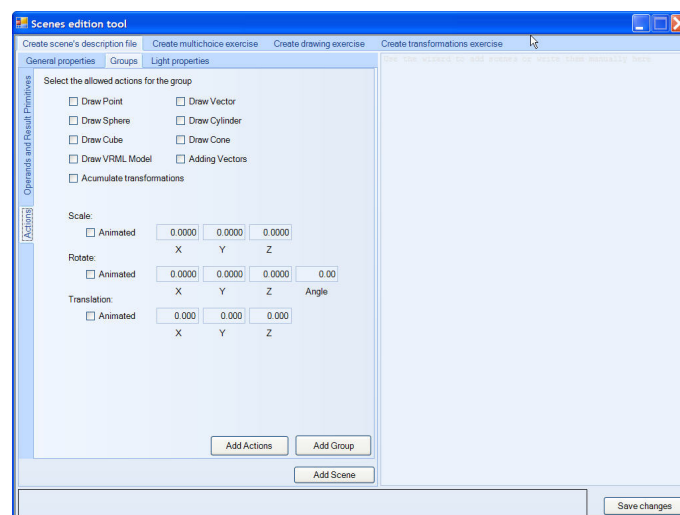
complete satisfactoriamente el ejercicio, a manera de confirmación del concepto.

Cada uno de los tipos de ejercicios que pueden ser implementados con el sistema tienen una sección separada en la herramienta de descripción de escenas (ver figura 4.3). Cada una de las escenas que el usuario describe aparece en el cuadro de texto ubicado en la parte derecha de la ventana de la interfaz. La retroalimentación de la herramienta hacia el usuario cuando intenta realizar una acción no permitida aparece en la barra de estado de la parte inferior de la ventana. En esta misma barra aparecen los mensajes de confirmación cuando una acción (por ejemplo añadir una primitiva a un grupo) se ha realizado con éxito.

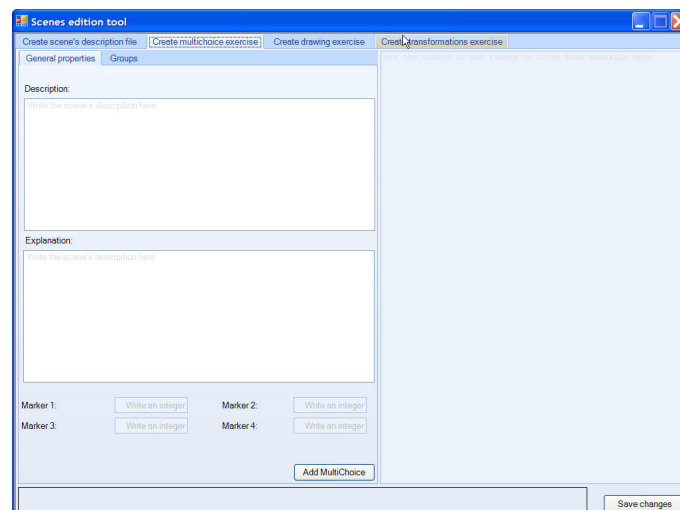
En la figura 4.3 se puede observar la interfaz gráfica de la herramienta de descripción de escenas. En la figura 4.3(a), se ven los controles para añadir una primitiva a un grupo. En 4.3(b), se muestra la ventana de configuración de las acciones permitidas en la escena. Y en 4.3(c), se muestran los controles para la configuración de las características generales de un ejercicio de opción múltiple.



(a)



(b)



(c)

Figura 4.3: Herramienta de descripción de escenas.

4.3. Tipos de ejercicios soportados en el sistema.

Con el conjunto de primitivas y acciones descrito, el sistema permite que el profesor prepare ejercicios demostrativos o para la evaluación de los alumnos. Los ejercicios pueden ser evaluados de manera inmediata para dar una retroalimentación interactiva al usuario.

El sistema, en el estado actual, soporta tres tipos de ejercicios:

- opción múltiple,
- ejercicios para completar dibujando y
- ejercicios de transformaciones.

En los ejercicios de *opción múltiple*, el profesor realiza una pregunta teórica y el alumno deberá escoger entre una de las respuestas propuestas. El sistema verifica si la opción seleccionada es correcta, y si no, describe la respuesta correcta (ver figura 4.4).

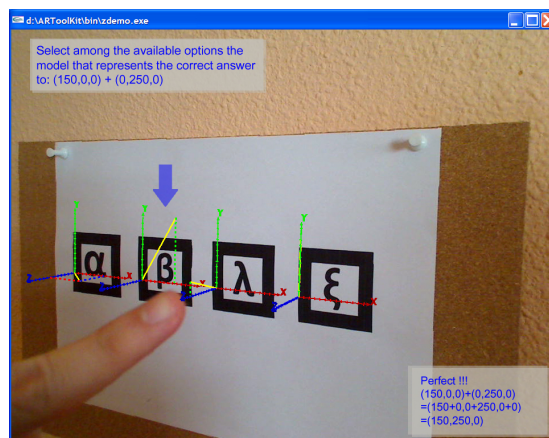


Figura 4.4: Ejercicios de opción múltiple.

los ejes han sido dibujados en rojo *eje-x*, verde *eje-y*, y azul *eje-z*, usando un sistema de coordenadas dextrógiro. Las posibles respuestas a la cuestión; se muestran en color amarillo. Una flecha en color azul indica cual es la opción que ha sido seleccionada.

Para seleccionar uno de los modelos presentados, el alumno debe cubrir parcial o totalmente el marcador deseado, durante un periodo de tiempo. El alumno

puede usar sus dedos para ello, tal como se muestra en la figura 17. Cuando un marcador es cubierto por unos segundos, el sistema interpreta la acción, como la selección del usuario, colocando una flecha sobre el modelo correspondiente, proporcionando así retroalimentación sobre la opción seleccionada. La flecha se dibuja en color azul si la respuesta dada por el alumno es correcta y en color rojo en caso contrario. El alumno puede seleccionar como máximo dos de las cuatro respuestas propuestas. Si después de la segunda selección el alumno no acierta, el sistema mostrará entonces la solución correcta al problema y mostrará la explicación correspondiente (dada por el profesor en el archivo de descripción).

En los *ejercicios de dibujo*, el estudiante debe dibujar las primitivas con la posición y orientación adecuadas. Por ejemplo, para sumar dos vectores, el alumno deberá usar la herramienta de dibujo para trazar el vector resultante en el espacio tridimensional.

El problema con este tipo de ejercicios es la precisión del dibujo a mano alzada en un entorno 3D. Para ayudar al usuario en esta tarea, el sistema muestra las proyecciones del cursor 3D sobre los ejes de coordenadas. Además, las coordenadas del mismo aparecen en todo momento en la esquina inferior derecha de la pantalla (ver figura 4.5). De manera adicional, el sistema permite configurar el grado de tolerancia que habrá de considerarse para determinar si el alumno ha dado una respuesta correcta.

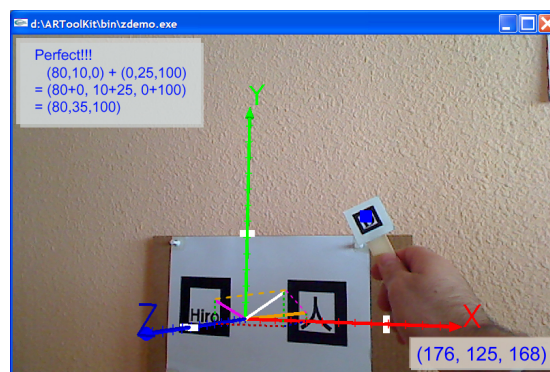


Figura 4.5: Ejercicios de dibujo.

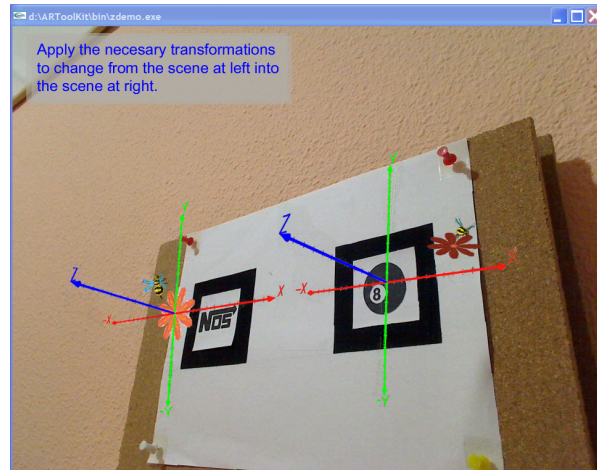
el alumno debe dibujar el vector resultante.

La composición matricial de las transformaciones afines y su utilización en informática gráfica es un tema que muchas veces es difícil de conceptualizar.

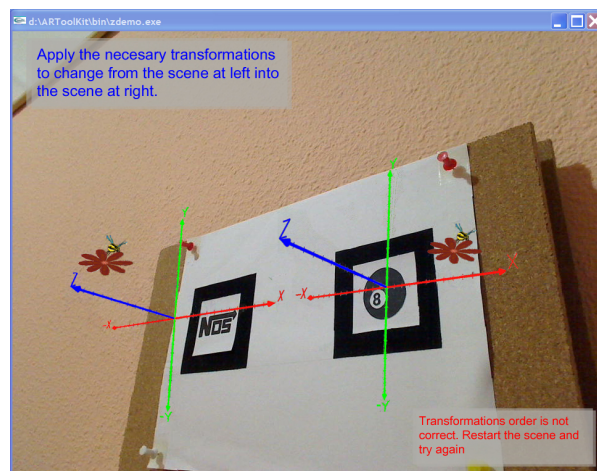
Imaginar gráficamente el impacto que la multiplicación de las matrices de transformación por los puntos que forman los vértices de las primitivas en cada escena no es algo trivial para todos los alumnos. Otro problema con este tipo de operaciones, es que el orden en el que se multipliquen las matrices de transformación para construir la matriz de transformación final define el resultado que se ha de obtener.

Junto con la explicación dada por el profesor, el tercer tipo de ejercicios, *los de transformaciones*, se pretende dar al alumno una herramienta para que vea y practique gráficamente el impacto de la construcción de las transformaciones afines en forma matricial sobre los puntos de la geometría de las primitivas en una escena.

En estos ejercicios se proporciona el estado inicial y final de un objeto. El alumno debe dar la secuencia de transformaciones para llevar el objeto del estado inicial al estado final. Si el usuario no aplica las transformaciones en el orden adecuado, el sistema le indicará el mensaje de error. Un ejemplo sencillo es cuando sólo son necesarias una rotación y una traslación. Cada transformación está asociada a un marcador diferente y el estudiante debe seleccionarlos en el orden apropiado. El orden de aplicación sobre la escena viene dado por el orden en que los marcadores aparecen ante la cámara (ver la figura 4.6).



(a)



(b)

Figura 4.6: Ejercicio de transformaciones.

El estudiante debe aplicar las transformaciones para llevar la flor y la abeja de su estado inicial (a la izquierda), a la posición y orientación deseadas (derecha) de la figura (a). Las transformaciones deben ser aplicadas en el orden correcto.

En la figura (b) el usuario ha utilizado las transformaciones en el orden incorrecto, por lo que el sistema le indica el error.

4.4. Conjunto de marcadores.

Cada acción permitida dentro de la escena está asociada a un marcador. Dichos marcadores, que son más pequeños que los usados como fondo, vienen fijados a un trozo de madera a modo de varita, para facilitar su manejo y orientación dentro de la escena (véase la figura 4.7).

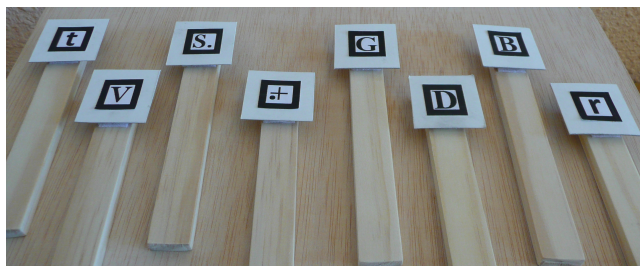


Figura 4.7: Herramientas del sistema.

Distintos marcadores usados para ejecutar las acciones en el sistema.

La figura 4.1 muestra la configuración típica del entorno de trabajo para la utilización del sistema. En esta configuración el usuario puede colocarse cómodamente del lado de la mano con la que escriba.

Uno de los problemas principales de las aplicaciones que usan ARToolKit es que el algoritmo implementado para la detección de los marcadores es muy sensible a las condiciones de iluminación. Los reflejos producidos por las luces de la habitación sobre la tinta negra de los marcadores reducen la fiabilidad del tracking. Esto provoca frustración en los usuarios, puesto que, aparte del propio ejercicio, deben preocuparse de la posición relativa entre la cámara, el marcador y las luces. Para buscar una solución a este problema, se probaron cuatro materiales distintos para construir los marcadores: tinta de impresora láser, tinta de impresora de inyección, cartulina negra y papel de terciopelo.

Los mejores resultados se obtuvieron con el papel de terciopelo. Al ser un material mate, se reducen los brillos y se facilita el proceso de segmentación de las imágenes capturadas por la cámara (véase la figura 4.8). Para facilitar la construcción de los marcadores, la parte interior de los mismos, que sirve para identificarlos, se sigue imprimiendo con tinta en lugar de usar papel de terciopelo.

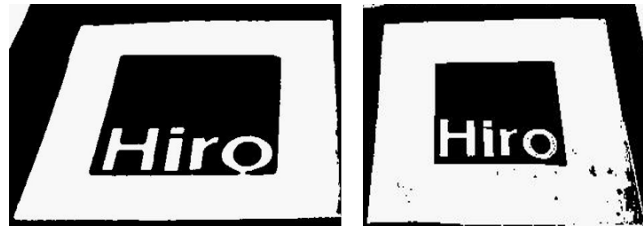


Figura 4.8: Comparación de materiales.

Resultados de la segmentación de un marcador hecho con papel terciopelo (izquierda) y tinta de impresora de inyección (derecha).

Cada ejercicio se representa en el sistema como un grafo de escena que contiene las primitivas gráficas y las acciones que se pueden ejecutar sobre el grafo. Cada escena viene definida por dos marcadores que cumplen dos objetivos: identificar la escena y definir la posición y orientación del sistema de coordenadas donde está definida. En las pruebas realizadas, dichos marcadores están contruidos sobre una hoja de papel pegada a un tablero de corcho. El usuario puede cambiar en cualquier momento el punto de vista desde donde se visualiza la escena moviendo la cámara con respecto a dichos marcadores.

El utilizar dos marcadores para identificar una escena permite reducir los problemas que se producen cuando el usuario tapa parcial o totalmente uno de los marcadores. La figura 4.9 muestra un ejemplo de este problema. Cuando el sistema es capaz de reconocer ambos marcadores, combina la información extraída de cada uno de ellos para obtener un resultado más estable.

La estrategia consiste en utilizar la posición relativa del sistema de coordenadas virtual respecto a los dos marcadores, dada por el usuario en el archivo en donde se listan los marcadores disponibles para cada escena. Cuando ambos marcadores están visibles, se combina la orientación y posición de ambos para calcular la orientación y posición de la escena respecto con a la cámara. Cuando solo uno de los marcadores está visible, solamente se usa la información de su posición relativa respecto al centro del origen de coordenadas para calcular posición y orientación de la escena completa.

Cuando ambos marcadores desaparecen mientras el usuario está trabajando con la escena, interviene el algoritmo explicado en la sección 4.9.3. Por otro lado, cuando se dispone de la información de uno o los dos marcadores; se

realiza un promedio de las últimas 3 matrices de transformación. Para realizar este promedio de matrices de transformación, se extraen la información de la rotación y la traslación de la matriz que calcula ARToolKit. Luego, utilizando cuaterniones, se calcula el promedio de la rotación en los últimos 3 frames, y por separado el promedio de la traslación.

Con la información del promedio de orientación y posición de los marcadores con respecto a la cámara, se construye de nuevo la matriz de transformación que se utiliza como la matriz Vista-Modelo de OpenGL para dibujar la escena. De esta forma, se consigue estabilizar un poco la escena teniendo una matriz de transformación sin tantas variaciones entre un frame y otro.

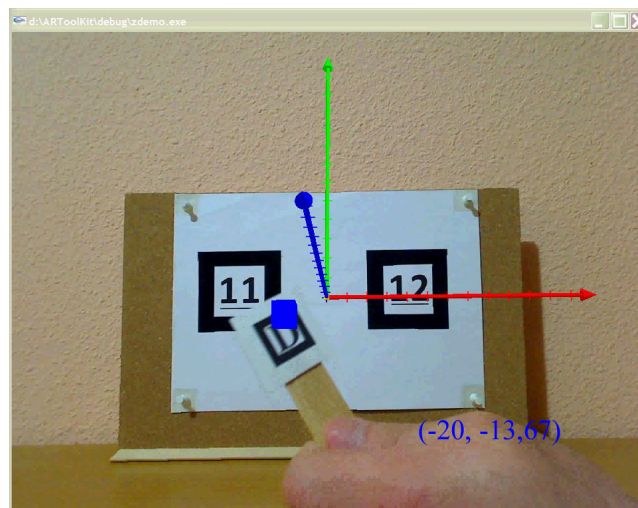


Figura 4.9: Cubriendo los marcadores.

Usar dos marcadores permite al sistema funcionar incluso cuando uno de ellos no es completamente visible.

4.5. Archivos de configuración.

En esta sección se hace una descripción de la estructura de archivos de configuración utilizados en el sistema. Se describirá la utilidad y la organización de cada uno de ellos.

El sistema trabaja con varios tipos de archivos:

- los propios marcadores,

- una lista con los marcadores disponibles en el sistema,
- la lista con las herramientas que se pueden utilizar,
- los marcadores que definen el fondo de cada ejercicio, y
- el archivo de sesión.

Para facilitar la creación, edición y lectura de estos archivos se utiliza un formato basado en XML. Los marcadores se describen mediante el formato de ARToolKit.



Figura 4.10: Sistema de archivos.

Estructura del sistema de archivos utilizado por el sistema.

En la figura 4.10, se puede apreciar la estructura de archivos que el sistema utiliza para su funcionamiento. El archivo que contiene la lista de marcadores es único en el sistema y contiene la ruta del archivo donde se encuentra cada marcador. De esta forma, cada marcador está identificado de forma unívoca en el sistema, desacoplando cada ejercicio de la ruta donde se encuentran los archivos de marcadores. Esto permite reutilizar marcadores en distintos ejercicios y sustituir, añadir o eliminar marcadores con facilidad.

La lista de herramientas disponibles relaciona acciones con marcadores. Cada ejercicio podrá especificar las acciones permitidas.

Para establecer el sistema de coordenadas de cada ejercicio, en el archivo de definición de fondos se define la posición del origen con respecto a cada

marcador. Por defecto, dicho sistema de coordenadas está situado en el centro de los dos marcadores de fondo de cada ejercicio.

Cada archivo de sesión puede contener la descripción de varios ejercicios. Cada ejercicio tiene la siguiente información:

- una descripción donde el profesor explica lo que se espera que aprenda el alumno con el ejercicio,
- los identificadores de los dos marcadores usados como fondo,
- la lista de primitivas que se podrán dibujar en la escena, aunque éstas no sean visibles inicialmente,
- la lista de acciones que se pueden realizar en la escena,
- el máximo número de primitivas que podrá insertar el usuario en la escena,
- uno o más grupos de primitivas, y
- la explicación de la respuesta correcta en caso de que se trate de un ejercicio de autoevaluación.

Los grupos de primitivas especifican qué primitivas del ejercicio se pueden manipular. A cada grupo de primitivas se le pueden asignar una o más acciones que afectarán a las primitivas que estén dentro del grupo. Las primitivas que no están incluidas en ningún grupo no pueden ser manipuladas por el usuario.

4.6. Otras características del sistema.

Cada ejercicio está identificado por un par de marcadores de fondo distinto. De esa forma, cambiar de un ejercicio a otro es tan sencillo como cambiar los marcadores de fondo. De esta forma, cada estudiante podría tener su propio conjunto de fondos a modo de libro que permitiera cambiar fácilmente de un ejercicio a otro.

La interacción con el sistema se basa fundamentalmente en el uso de los marcadores y del menú. Se ha buscado minimizar al máximo la necesidad de usar el teclado. El menú se habilita con un marcador especial. Cuando el menú aparece en la pantalla se interrumpe la interacción con la escena. Las opciones del

menú son: reiniciar la escena, guardar la escena, resolver el ejercicio, mostrar/ocultar la descripción de la escena, borrar primitivas y salir (véase la figura 4.11). Para los usuarios avanzados se han definido una serie de atajos de teclado para lanzar acciones sin necesidad de interactuar con el menú.

El sistema incluye además la opción de configurar el color y la posición de la luz que iluminará la escena. Esta característica permite mejorar la percepción de los objetos tridimensionales mostrados. La posición puede ser fijada en la escena o bien puede seguir la posición de la cámara, apuntando a los marcadores de fondo.

En caso de que el usuario cometa algún error, existe un marcador que simula la acción de un borrador, y que reinicia la escena a su estado inicial (como fue cargada desde el archivo). También se puede eliminar solamente una primitiva de la escena. Para ello se puede usar la opción del menú correspondiente y seleccionar la primitiva a borrar en la lista desplegable (ver figura 4.11).

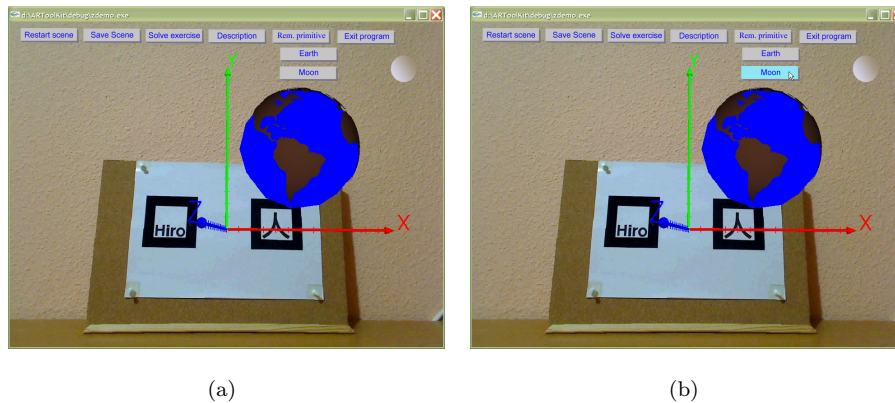


Figura 4.11: Menú del sistema.

para eliminar una primitiva el usuario la selecciona mediante el cursor desde la lista que aparece en el menú. *Derecha*, el menú desplegado, *Izquierda*, el usuario seleccionando una de las opciones.

4.7. Escenarios de la aplicación.

Considerando las características descritas, se plantea el uso de la aplicación en dos escenarios distintos: durante una clase, como material de apoyo a la

explicación del profesor, y en el laboratorio o en casa, donde el alumno resuelve ejercicios planteados por el profesor.

En el primer escenario, el profesor es quien dispone de los marcadores que se van a utilizar en la sesión. Cuando el profesor desee reforzar algún concepto, tan sólo deberá seleccionar los marcadores de fondo correspondientes, y apuntar la cámara. Una vez que el sistema detecta los marcadores, el profesor puede manipular la escena según las acciones previstas en su definición. En cualquier momento es posible mover la cámara para cambiar el punto de vista y observar el modelo desde otra posición.

En el segundo escenario la principal diferencia es que las instrucciones para la realización del ejercicio están codificadas en el archivo de escena. En ambos casos, el usuario puede guardar el estado actual de la escena y continuar en otra sesión. La ventaja de usar archivos para almacenar la descripción de las escenas es que el profesor puede definir dichos archivos y puede distribuirlos a los alumnos mediante correo electrónico o mediante una página web. También puede crear distintos archivos y enviarlos por separado a distintos alumnos o grupos con el propósito de evaluar sus conocimientos.

4.8. Ejemplo de un ejercicio.

Esta sección describe un ejemplo de uso del sistema. El objetivo del ejemplo mostrado en las figuras 4.12 y 4.13 es estudiar la suma vectorial. Para ello, el profesor debe preparar una escena con los elementos necesarios. A continuación se muestra una posible implementación del ejercicio:

```
<SCENE>
  <DESCRIPTION>
    This exercise shows the concept of the sum of two vectors
  </DESCRIPTION>
  <PRIMITIVESNUMBER>
    4
  </PRIMITIVESNUMBER>
  <MARKERS>
    2
    3
  </MARKERS>
```

```

<GROUP>
  <OPERANDS>
    <PRIMITIVESLIST>
      <AXES>
        <PROPERTIES>
          IDENT 1
          SHOWNEGATIVES 1
          LENGTH 250
          TICKSDISTANCE 20
          VISIBLE 1
        </PROPERTIES>
      </AXES>
      <VECTOR>
        <PROPERTIES>
          IDENT 2
          COLOR 1.0 1.0 1.0 1.0
          WIDTH 2
          VISIBLE 0
        </PROPERTIES>
      </VECTOR>
      <VECTOR>
        <PROPERTIES>
          IDENT 3
          COLOR 1.0 0.0 1.0 1.0
          WIDTH 2
          VISIBLE 0
        </PROPERTIES>
      </VECTOR>
    </PRIMITIVESLIST>
  </OPERANDS>
  <RESULT>
    <PRIMITIVESLIST>
      <VECTOR>
        <PROPERTIES>
          IDENT 4
          POINT1 0 0 0 1
          POINT2 0 0 0 1
          COLOR 1.0 0.647 0.0 1.0
          WIDTH 4
          COMPONENTS 1
          VISIBLE 0
        </PROPERTIES>

```

```

        </VECTOR>
    </PRIMITIVESLIST>
</RESULT>
<ACTIONLIST>
    DRAWVECTOR 1
    SUMVECTOR 1
</ACTIONLIST>
</GROUP>
<LIGHT>
    FOLLOWCAM 0
    POSITION 250 250 250
    COLOR 1.0 1.0 0.0
</LIGHT>
</SCENE>

```

Al inicio del código se puede observar cómo se configuran las propiedades generales de la escena: La descripción de la misma y los índices de los marcadores que la representan en la lista de marcadores del sistema.

En este caso, se ha definido un grupo de primitivas en la escena. Este grupo está formado por los dos vectores que habrán de sumarse (los operandos) y un vector como resultado. Además de los vectores se define el sistema de coordenadas sobre el que se trabajará en la escena. Para todas las primitivas se definen solo las propiedades que han de tomar un valor diferente al predeterminado. Entre las propiedades más importantes está el ID de las primitivas, el nombre con el que aparecerá en la lista desplegable de la opción “Remover Primitiva” del menú.

Se puede observar que los tres vectores que han sido declarados en la descripción de la escena aparecen con la propiedad “Visible” con el valor 0, lo cual indica que serán dibujados después. Para ello, al grupo se le ha definido la opción de “Dibujo de vectores”. También se ha habilitado la acción “Sumar vectores” para este grupo de primitivas, lo cual indica que cuando se hayan dibujado los dos vectores, el usuario podrá indicar al sistema que calcule el resultado, mostrando el marcador adecuado.

Como se puede observar en la descripción de las propiedades del vector resultante, cada vector está definido por una par de puntos, POINT1 y POINT2. Estas dos propiedades no hace falta definir las si el estado inicial del vector es invisible, puesto que el inicio y el fin del vector lo determinará el usuario usando

la herramienta de dibujo. En este caso han sido dejadas a propósito para ilustrar la manera en cómo se definen los vectores en el sistema.

Finalmente se configuran las características de la luz de la escena, que en este caso estará fija en la posición $(250, 250, 250)$ y será de color amarillo.

Inicialmente, al apuntar la cámara a los marcadores de fondo, se muestra únicamente el sistema de coordenadas. Para dibujar los vectores a sumar, el profesor debe seleccionar la herramienta de dibujo adecuada e introducirla en el campo de visión de la cámara. Una vez que el sistema detecta el nuevo marcador, inmediatamente entra en modo de dibujo, añadiendo ayudas visuales sobre los ejes para posicionar más fácilmente el vector en el espacio.

Una vez que se han dibujado los operandos, el profesor deberá introducir en la escena el marcador asignado a la acción de suma vectorial. El sistema mostrará el resultado de sumar los vectores automáticamente. Una vez que se ha completado el ejemplo, el profesor podrá mover la cámara para observar la escena desde otro punto de vista.

El usuario puede guardar la escena en un archivo para su uso en otra sesión en cualquier momento. Para completar el proceso de aprendizaje, el profesor puede dar a los alumnos un ejercicio para resolver, evaluarlo y asignar una nota.

En las figuras 4.12 y 4.13 se pueden observar las secuencias de frames con ejemplos del sistema. *Secuencia 1*: se calcula la suma de dos vectores que inician en el origen (pasos 1 - 6); luego el usuario elige la opción “Reiniciar la escena” (paso 7), la escena regresa a su estado original. *Secuencia 2*: Se dibujan y calcula la suma de una secuencia de dos vectores que inicia en el origen de coordenadas (pasos (1 - 3); luego usando el marcador de borrado, se reinicia la escena (pasos 4 y 5).

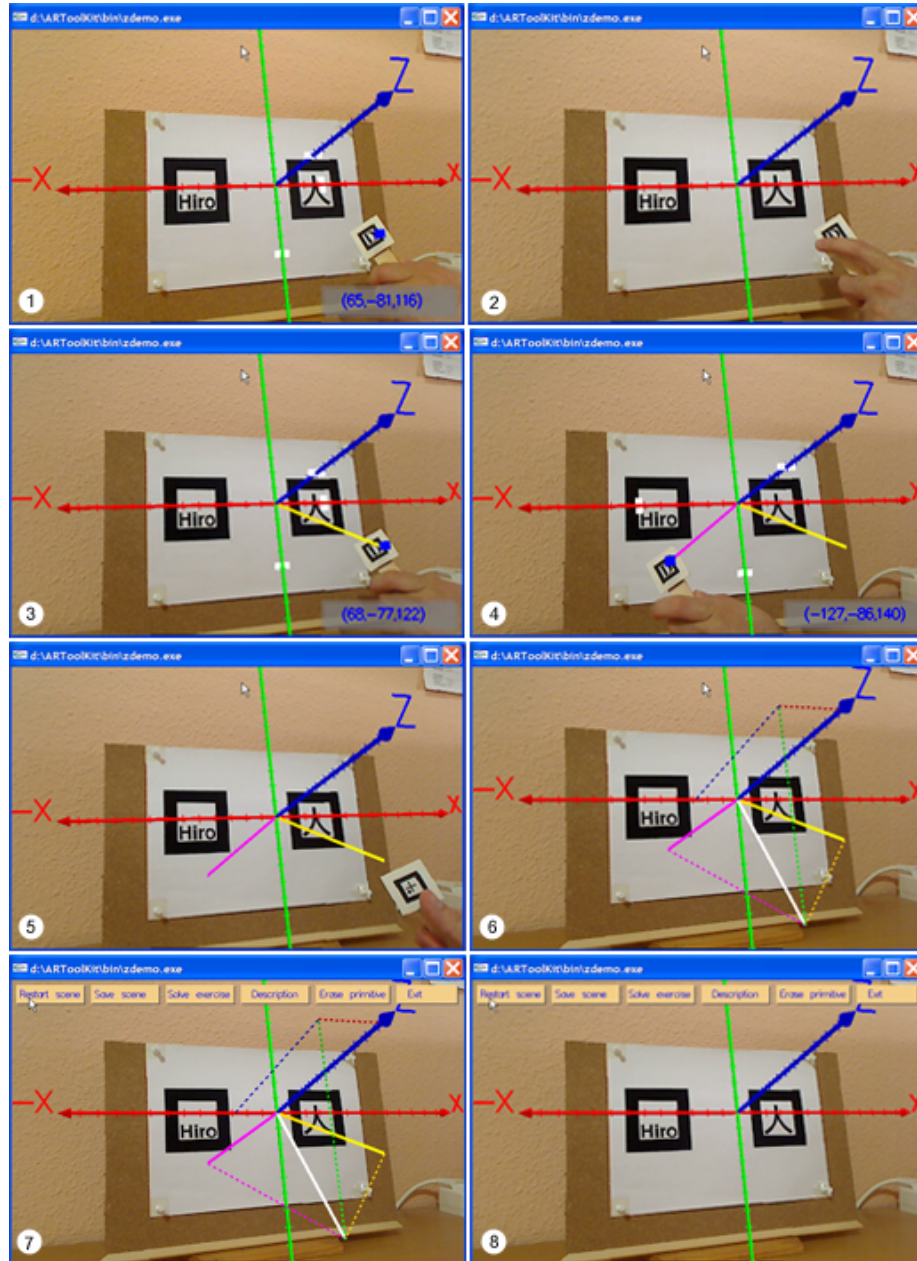


Figura 4.12: Ejemplo de uso del sistema - Secuencia1.

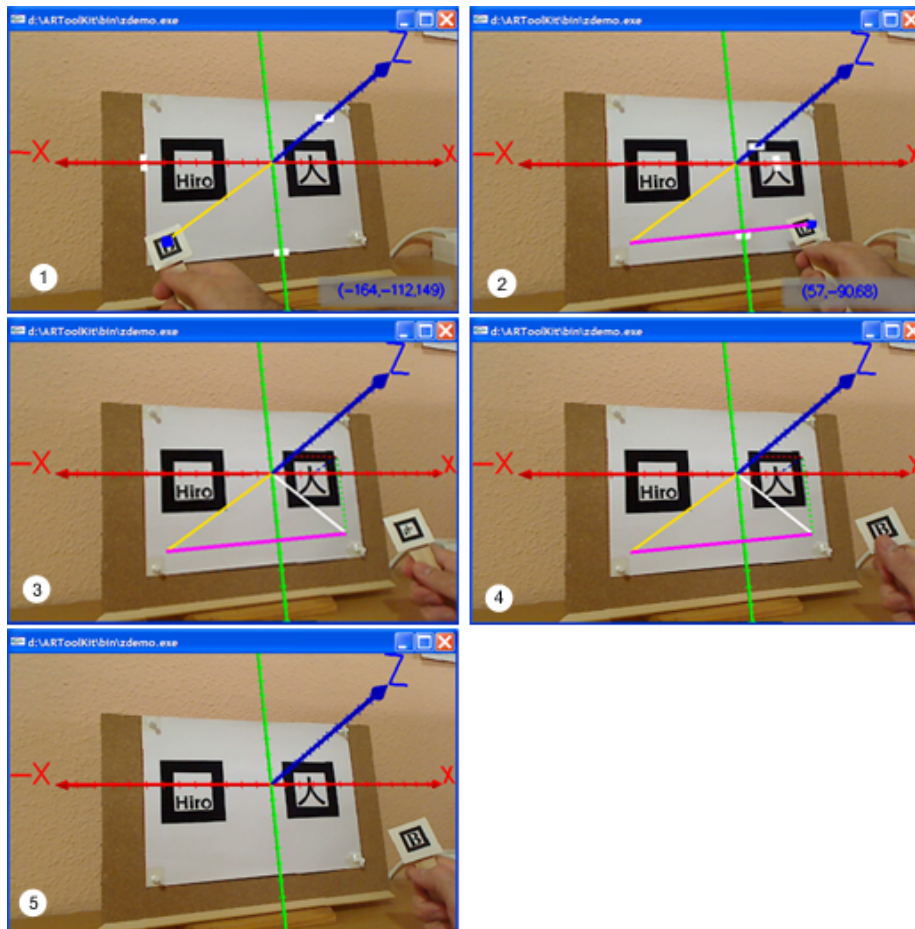


Figura 4.13: Ejemplo de uso del sistema - Secuencia2.

4.9. Aspectos de diseño.

A continuación se presenta el análisis y diseño en el que se basa el desarrollo del sistema que se ha descrito en este trabajo. Primero se presenta la lista de casos de uso que generalizan las funciones de la aplicación. Se muestra el diagrama general de casos de uso y la descripción corta y extendida de cada uno. Después se listan las clases que se han codificado y que son componentes vitales de la herramienta. Finalmente se hace la descripción del algoritmo que se ha implementado para mejorar la robustez de la herramienta frente a los cambios de iluminación.

4.9.1. Casos de uso.

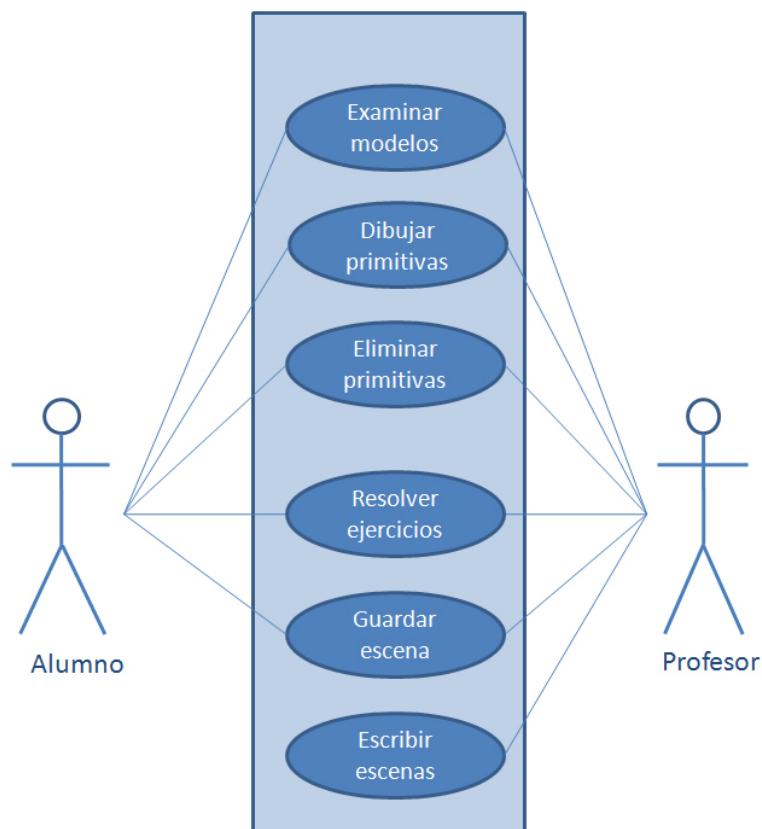


Figura 4.14: Diagrama de casos de uso del sistema.

4.9.2. Examinar modelos.

Nombre	Examinar modelos
Autor, Fecha	Gustavo Roveló, 4 de Agosto de 2008
Descripción	Permite al usuario observar el modelo 3D correspondiente al marcador que capte con la cámara Web en una monitor de PC convencional.
Actores	Profesor, Alumno
Precondiciones	<ul style="list-style-type: none"> ■ Que el usuario haya iniciado la ejecución del sistema. ■ Que el usuario tenga disponible el manual con los ejercicios y marcadores adecuados.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario enfoca la cámara web o dirige su mirada con el HMD 2. El sistema reconoce el patrón observado y despliega en la pantalla del ordenador o en el lente del HMD el modelo correspondiente
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario inicia la interacción con el modelo que ve en el dispositivo de visualización utilizado 2. El usuario activa el modo HOLD de la aplicación y la escena no cambia a pesar de que el usuario modifique el punto de vista.
Poscondiciones	-

4.9.3. Examinar modelos versión extendida.

Número	1
Nombre	Examinar modelos
Autor, Fecha	Gustavo Rovelo, 4 de Agosto de 2008
Modificación, Fecha	Gustavo Rovelo, 23 de Junio de 2009
Disparador	El sistema detecta que un marcador válido está siendo observado por el usuario (actor)
Actores	Profesor, Alumno
Descripción	Permite al usuario observar el modelo 3D correspondiente al marcador que capte con la cámara Web en un monitor de PC convencional.
Precondiciones	<ul style="list-style-type: none"> ▪ Que el usuario haya iniciado la ejecución del sistema. ▪ Que el usuario tenga disponible el manual con los ejercicios y marcadores adecuados.
Poscondiciones	-
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario enfoca la cámara web o dirige su mirada con el HMD 2. El sistema reconoce el patrón observado y despliega en la pantalla de la computadora o en el lente del HMD el modelo correspondiente
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario inicia la interacción con el modelo que ve en el dispositivo de visualización utilizado 2. El usuario activa el modo HOLD de la aplicación y la escena no cambia a pesar de que el usuario modifique el punto de vista.
Excepciones	Si se bloquea parcialmente el marcador y no se puede capturar por la cámara, el sistema se basará en la historia inmediata para seguir mostrando el modelo 3D. Si después de transcurrido cierto tiempo el sistema sigue sin capturar el marcador completamente, el sistema dejará de mostrar el modelo.
Prioridad	Máxima
Incluye	-
Frecuencia de uso	Frecuentemente
Req. Especiales	Requisitos mínimos para ARToolKit
Suposiciones	-
Notas	-

4.9.4. Dibujar primitivas.

Nombre	Dibujar primitivas
Autor, Fecha	Gustavo Rovelo, 4 de Agosto de 2008
Descripción	Permite al usuario situar las primitivas del sistema dentro de una escena, definiendo su posición usando la herramienta de dibujo del sistema.
Actores	Profesor, Alumno
Precondiciones	<ul style="list-style-type: none"> ■ El sistema debe haber reconocido un patrón de algún ejercicio y se debe estar visualizando el modelo 3D correspondiente.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce la herramienta de dibujo en la escena 2. El usuario mueve la herramienta de dibujo hasta situarla en el lugar deseado y simula un clic sobre ella 3. El sistema ubica la primitiva tomando como centro el punto dado
Flujo alternativo	-
Poscondiciones	El modelo se verá afectado por las acciones del usuario y cambiará de manera dinámica

4.9.5. Dibujar primitivas versión extendida.

Número	2
Nombre	Dibujar primitivas
Autor, Fecha	Gustavo Rovelo, 4 de Agosto de 2008
Modificación, Fecha	Gustavo Rovelo, 23 de Junio de 2009
Disparador	El usuario introduce en la escena el marcador correspondiente a la herramienta de dibujo.
Actores	Profesor, Alumno
Descripción	Permite al usuario situar las primitivas del sistema dentro de una escena, definiendo su posición usando la herramienta de dibujo del sistema.
Precondiciones	<ul style="list-style-type: none"> ■ El sistema debe haber reconocido un patrón de algún ejercicio y se debe estar visualizando el modelo 3D correspondiente. ■ Debe haber suficientes primitivas disponibles para recibir las coordenadas dadas por el usuario.
Poscondiciones	El modelo se verá afectado por las acciones del usuario y cambiará de manera dinámica
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce la herramienta de dibujo en la escena 2. El usuario mueve la herramienta de dibujo hasta situarla en el lugar deseado y simula un clic sobre ella 3. El sistema ubica la primitiva tomando como centro el punto dado
Flujo alternativo	-
Excepciones	Si no hay más primitivas declaradas en la escena que puedan actuar como receptoras de las coordenadas el sistema mostrará un mensaje al usuario indicando el error. El usuario cubre los marcadores de fondo, por lo que el sistema activa el modo HOLD automáticamente y el usuario no verá reflejado el cambio del punto de vista hasta que los marcadores vuelvan a ser visibles.
Prioridad	Segundo en el proceso de desarrollo
Incluye	-
Frecuencia de uso	Frecuentemente
Req. Especiales	Requisitos mínimos para ARToolKit
Suposiciones	-
Notas	-

4.9.6. Eliminar primitivas.

Nombre	Eliminar primitivas
Autor, Fecha	Gustavo Roveló, 4 de Agosto de 2008
Descripción	Permite al usuario eliminar las primitivas de una escena.
Actores	Profesor, Alumno
Precondiciones	<ul style="list-style-type: none"> ■ Debe haber una escena visible en la pantalla.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce la herramienta de borrado en la escena 2. El usuario simula un clic sobre ella 3. El sistema elimina las primitivas de la escena
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario activa el menú del sistema 2. El usuario elige la opción correspondiente a Remover primitivas 3. El usuario selecciona la primitiva que desea borrar de la escena
Poscondiciones	El modelo se verá afectado por las acciones del usuario y cambiará de manera dinámica

4.9.7. Eliminar primitivas versión extendida.

Número	3
Nombre	Eliminar primitivas
Autor, Fecha	Gustavo Roveló, 4 de Agosto de 2008
Modificación, Fecha	Gustavo Roveló, 23 de Junio de 2009
Disparador	El usuario introduce en la escena el marcador correspondiente a la herramienta de borrado. El usuario selecciona del menú la opción correspondiente a Remove Primitives.
Actores	Profesor, Alumno
Descripción	Permite al usuario eliminar primitivas de una escena
Precondiciones	<ul style="list-style-type: none"> ■ Debe de haber una escena visible en la pantalla
Poscondiciones	El modelo se verá afectado por las acciones del usuario y cambiará de manera dinámica
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce la herramienta de borrado en la escena 2. El usuario simula un clic sobre ella 3. El sistema elimina las primitivas de la escena
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario usuario activa el menú del sistema 2. El usuario elige la opción correspondiente a Borrar primitivas. 3. El usuario selecciona la primitiva que desea borrar de la escena 4. El sistema elimina las primitivas de la escena
Excepciones	-
Prioridad	Tercero en el proceso de desarrollo
Incluye	-
Frecuencia de uso	Frecuentemente
Req. Especiales	Requisitos mínimos para ARToolKit
Suposiciones	-
Notas	-

4.9.8. Resolver ejercicios.

Nombre	Resolver ejercicios
Autor, Fecha	Gustavo Roveló, 4 de Agosto de 2008
Descripción	El usuario podrá interactuar con los modelos 3D utilizando el señalador, las opciones del menú para modificar la escena y resolver los ejercicios que se le plantean. Poner puntos en la escena, dibujar vectores, definir el orden de aplicación de las transformaciones afines.
Actores	Profesor, Alumno
Precondiciones	<ul style="list-style-type: none"> ■ Los marcadores correspondientes a cualquiera de los tres tipos de ejercicios del sistema deben estar visibles.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario dirige su punto de vista hacia el marcador del ejercicio 2. El usuario realiza las modificaciones pertinentes en la escena para completar el ejercicio indicado 3. El sistema le indicará al usuario si la solución es correcta o no 4. El usuario decide reiniciar el ejercicio o cambiar a otro
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario puede guardar el resultado de su trabajo para continuar con él en otra ocasión 2. El usuario puede decidir utilizar la opción de Mostrar solución del sistema para ver el resultado del ejercicio
Poscondiciones	El ejercicio queda solucionado correctamente

4.9.9. Resolver ejercicios versión extendida.

Número	4
Nombre	Resolver ejercicios
Autor, Fecha	Gustavo Roveló, 4 de Agosto de 2008
Modificación, Fecha	Gustavo Roveló, 23 de Junio de 2009
Disparador	El sistema detecta que el usuario está observando un marcador que corresponde a un ejercicio
Actores	Profesor, Alumno
Descripción	El usuario podrá interactuar con los modelos 3D utilizando el señalador, las opciones del menú para modificar la escena y resolver los ejercicios que se le plantean. Poner puntos en la escena, dibujar vectores, definir el orden de aplicación de las transformaciones afines.
Precondiciones	<ul style="list-style-type: none"> Los marcadores correspondientes a cualquiera de los tres tipos de ejercicios del sistema deben estar visibles.
Poscondiciones	El ejercicio queda solucionado correctamente
Flujo Normal	<ol style="list-style-type: none"> El usuario dirige su punto de vista hacia el marcador del ejercicio El usuario realiza las modificaciones pertinentes en la escena para completar el ejercicio indicado El sistema le indicará al usuario si la solución es correcta o no El usuario decide reiniciar el ejercicio o cambiar a otro
Flujo alternativo	<ol style="list-style-type: none"> El usuario puede guardar el resultado de su trabajo para continuar con él en otra ocasión El usuario puede decidir utilizar la opción de Mostrar solución del sistema para ver el resultado del ejercicio
Excepciones	El usuario cubre los marcadores de fondo, por lo que el sistema activa el modo HOLD automáticamente y el usuario no verá reflejado el cambio del punto de vista hasta que los marcadores vuelvan a ser visibles.
Prioridad	Cuarto módulo a desarrollar
Incluye	-
Frecuencia de uso	Frecuentemente
Req. Especiales	Requisitos mínimos para ARToolKit
Suposiciones	-
Notas	-

4.9.10. Guardar escena.

Nombre	Guardar escena
Autor, Fecha	Gustavo Roveló, 4 de Agosto de 2008
Descripción	Permite al usuario guardar los cambios que haya realizado a la escena con la que esté trabajando.
Actores	Profesor, Alumno
Precondiciones	<ul style="list-style-type: none"> ■ El sistema debe haber reconocido un patrón de algún ejercicio y se debe estar visualizando el modelo 3D correspondiente.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario muestra el marcador correspondiente a la herramienta de guardar en la Escena 2. El usuario simula un clic sobre la herramienta. 3. El sistema despliega un mensaje con el estado de la acción: escena guardada o error en el proceso
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario utilizando las opciones disponibles del menú elige guardar la escena que está viendo asociada al marcador correspondiente 2. El usuario cancela la operación de guardar y el modelo queda tal cual estaba antes de iniciar las modificaciones.
Poscondiciones	Se genera un nuevo archivo en donde se almacena en el disco duro la escena con los cambios realizados por el usuario

4.9.11. Guardar escena versión extendida.

Número	5
Nombre	Guardar escena
Autor, Fecha	Gustavo Rovelo, 4 de Agosto de 2008
Modificación, Fecha	Gustavo Rovelo, 23 de Junio de 2009
Disparador	El usuario introduce en la escena el marcador correspondiente a la opción Guardar. El usuario selecciona la opción de guardar el trabajo que se encuentra en el menú del sistema.
Actores	Profesor, Alumno
Descripción	Permite al usuario guardar los cambios que haya realizado a la escena con la que esté trabajando.
Precondiciones	<ul style="list-style-type: none"> El sistema debe haber reconocido un patrón de algún ejercicio y se debe estar visualizando el modelo 3D correspondiente.
Poscondiciones	Se genera un nuevo archivo en donde se almacena en el disco duro la escena con los cambios realizados por el usuario.
Flujo Normal	<ol style="list-style-type: none"> El usuario muestra el marcador correspondiente a la herramienta de guardar en la Escena El usuario simula un clic sobre la herramienta. El sistema despliega un mensaje con el estado de la acción: escena guardada o error en el proceso
Flujo alternativo	<ol style="list-style-type: none"> El usuario utilizando las opciones disponibles del menú elige guardar la escena que está viendo asociada al marcador correspondiente El usuario cancela la operación de guardar y el modelo queda tal cual estaba antes de iniciar las modificaciones.
Excepciones	-
Prioridad	Quinto en el proceso de desarrollo
Incluye	-
Frecuencia de uso	Frecuentemente
Req. Especiales	Requisitos mínimos para ARToolKit
Suposiciones	El equipo de cómputo del usuario debe tener suficiente espacio en disco para el almacenamiento del modelo
Notas	-

4.9.12. Escribir escenas.

Nombre	Escribir escenas
Autor, Fecha	Gustavo Roveló, 4 de Agosto de 2008
Descripción	El profesor, haciendo uso de la herramienta de descripción de escenas podrá describir las características de una escena virtual, que asociará a un marcador para que posteriormente el alumno pueda resolver el ejercicio planteado
Actores	Profesor
Precondiciones	<ul style="list-style-type: none"> ■ Que el usuario haya iniciado la herramienta de descripción de escenas.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario define las características generales de la escena 2. El usuario crea los grupos con las primitivas que estarán disponibles en la escena. 3. El usuario configura las acciones que podrán ser realizadas sobre los grupos que ha incluido en la escena 4. El usuario configura las características de la luz que iluminará la escena 5. El usuario añade la escena al archivo 6. El usuario repite la operación tantas veces como escenas necesite
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario describe las escenas escribiendo directamente en un archivo la descripción de las escenas.
Poscondiciones	Se almacenará en el directorio de trabajo del sistema el nuevo archivo con las escenas descritas

4.9.13. Escribir escena versión extendida.

Número	6
Nombre	Escribir escena
Autor, Fecha	Gustavo Roveló, 4 de Agosto de 2008
Modificación	Gustavo Roveló
Fecha modificación	23 de Junio de 2009
Disparador	El usuario inicia la aplicación de descripción de escenas
Actores	Profesor, Alumno
Descripción	El profesor, haciendo uso de la herramienta de descripción de escenas podrá describir las características de una escena virtual, que asociará a un marcador para que posteriormente el alumno pueda resolver el ejercicio planteado.
Precondiciones	<ul style="list-style-type: none"> ▪ Que el usuario haya iniciado la herramienta de descripción de escenas.
Poscondiciones	Se almacenará en el directorio de trabajo del sistema el nuevo archivo con las escenas descritas
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario define las características generales de la escena 2. El usuario crea los grupos con las primitivas que estarán disponibles en la escena. 3. El usuario configura las acciones que podrán ser realizadas sobre los grupos que ha incluido en la escena 4. El usuario configura las características de la luz que iluminará la escena 5. El usuario añade la escena al archivo 6. El usuario repite la operación tantas veces como escenas necesite
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario describe las escenas escribiendo directamente en un archivo la descripción de las escenas.
Excepciones	-
Prioridad	Sexto módulo a desarrollar
Incluye	-
Frecuencia de uso	Frecuentemente
Req. Especiales	
Suposiciones	El equipo de cómputo del usuario debe tener suficiente espacio en disco para el almacenamiento del modelo
Notas	No habrá una diferenciación explícita del rol entre alumno y profesor, ya que a conveniencia un alumno puede plantearse sus propios ejercicios.

4.9.14. Diagramas de clase.

En esta sección se muestra el diagrama de clases del sistema y las relaciones entre ellas. Posteriormente se muestra la interfaz de cada clase, con sus miembros y funciones.

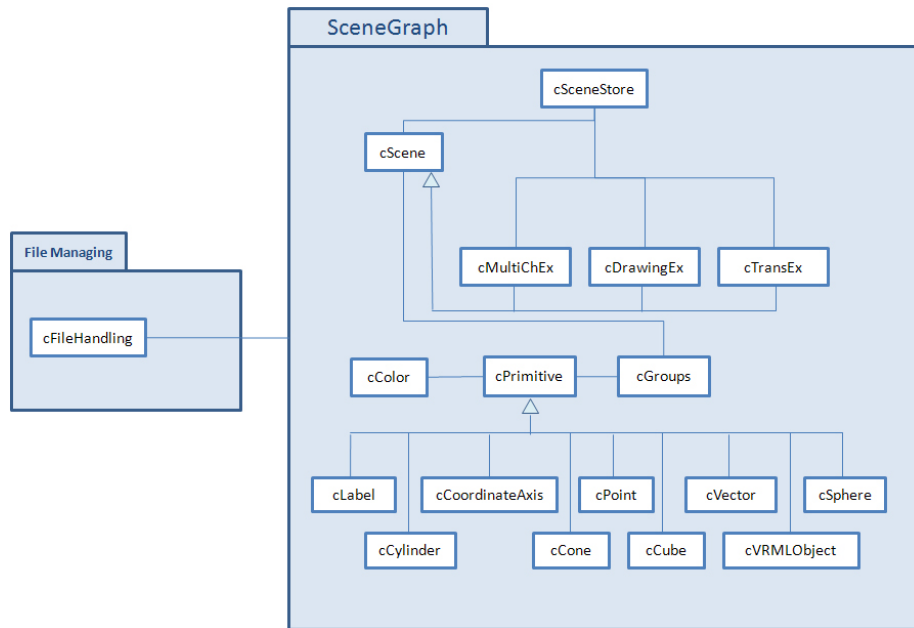


Figura 4.15: Diagrama de clases del sistema.

4.9.15. Clase cColor.

cColor
<pre>private: r : float; g : float; b : float; a : float;</pre>
<pre>public: cColor(); cColor(float r, float g, float b, float a); ~cColor(); void getColor(float *R, float *G, float *B, float *A); void setColor(float R, float G, float B, float A);</pre>

4.9.16. Clase cPrimitive.

cPrimitive
private: visible : bool transformable : bool primitiveColor : cColor name : string idPrimitive : int
public: cPrimitive(void); virtual ~cPrimitive(void); virtual void drawPrimitive(void); cColor getColor(); bool getVisible(); bool getTransformable(); int getID(); string getName(); void setColor(float R, float G, float B, float A); void setID(int pID); void setVisible(bool pVisible); void setTransformable(); void setName(string pName);

4.9.17. Clase cPoint.

cPoint: cPrimitive
<pre>private: centre : float*; size : int;</pre>
<pre>public: cPoint(); cPoint(int pX, int pY, int pZ, int pW); ~cPoint(); void convert2Vector(); void drawPrimitive(void); vector <float> getCoord(); int getSize(); void setCoord(vector <float> pCoord); void setSize(int pTam);</pre>

4.9.18. Clase cVector.

```
cVector: cPrimitive
```

```
private:
```

```
vectorP1 : float*;
```

```
vectorP2 : float*;
```

```
vectorN : float*;
```

```
module : float;
```

```
lon : float;
```

```
lat : float;
```

```
public:
```

```
cVector();
```

```
cVector(float *pCrdP1, float *pCrdP2);
```

```
~cVector();
```

```
void drawPrimitive(void);
```

```
vector <float> getCoord();
```

```
float getLatitud();
```

```
float getLongitud();
```

```
float getModule();
```

```
void normalize();
```

```
void setCoord(float *crdP1, float *crdP2);
```

```
void setModule();
```

```
void setLongitud();
```

```
void setLatitud();
```

4.9.19. Clase cSphere.

```
cSphere: cPrimitive
```

```
private:
```

```
centre : float*;
```

```
radius: float;
```

```
public:
```

```
Sphere();
```

```
Sphere(vector<float> pCentre, float pRadius);
```

```
~Sphere();
```

```
void  dibujaPrimitiva(void);
```

```
vector<float>  getCoord();
```

```
float  getRadius();
```

```
void    setCoord(vector<float> pCenter);
```

```
void    setRadius(float pRadius);
```

4.9.20. Clase cCube.

<code>cCube: cPrimitive</code>
private: <code>centre : float*;</code> <code>size: float;</code>
public: <code>Cube();</code> <code>Cube(vector<float> pCenter, float pSize);</code> <code>~Cube();</code> <code>void dibujaPrimitiva(void);</code> <code>vector<float> getCoord();</code> <code>bool getDrawComponents();</code> <code>float getSize();</code> <code>void setCoord(vector<float> pCenter);</code> <code>void setSize(float pSize);</code>

4.9.21. Clase cCone.

<code>cCone: cPrimitive</code>
<pre>private: centre : float*; radius: float; height : float;</pre>
<pre>public: Cone(); Cone(float *pCenter, float pRadius, float pHeight); ~Cone(); void dibujaPrimitiva(void); float getRadius(); float getHeight(); void setCoord(float *pCenter); void setRadius(float pRadius); void setHeight(float pHeight);</pre>

4.9.22. Clase cCylinder.

```
cCylinder: cPrimitive
```

```
private:
```

```
centre : float *;
```

```
radius: float;
```

```
height : float;
```

```
public:
```

```
Cylinder ();
```

```
Cylinder (float *Center , float Radius , float Height );
```

```
~Cylinder ();
```

```
void      dibujaPrimitiva (void);
```

```
float     getRadius ();
```

```
float     getHeight ();
```

```
void      setCoord (vector<float> pCenter);
```

```
void      setRadius (float pRadius);
```

```
void      setHeight (float pHeight);
```

4.9.23. Clase cVRMLObject.

cVRMLObject: cPrimitive
<pre> private: modelParth : string; Center : float*; vrml_id : int; vrml_id_orig : int; </pre>
<pre> public: VRMLObject(); ~VRMLObject(); void dibujaPrimitiva(void); vector <float> getCoord(); string getModelPath(); int getVRMLId(); int getVRMLIdOrig(); void setCoord(float *pCenter); void setModelPath(string pPath); void setVRMLId(int pId); void setVRMLIdOrig(int pIdOrig); </pre>

4.9.24. Clase cLabel.

<code>cLabel: cPrimitive</code>
<pre>private: centre : float* message : string</pre>
<pre>public: cLabel(); cLabel(vector <float> pCentre, string pMessage); ~cLabel(); void drawPrimitive(void); vector <float> getCoord(); string getMessage(); void setCoord(vector <float> pCentre); void setMessage(string pMessage);</pre>

4.9.25. Clase cCoordinateAxis.

```

cCoordinateAxis: cPrimitive

private:\\
flgLineasGuia : bool;
flgPlanosReferencia : bool;
flgNegativos: bool;
longitudEjes : int;
marcasIntermedias: int;

private:
void axis();
void planes();
void ticks();
public:
cCoordinateAxis();
cCoordinateAxis(int Tam,int Marc,bool pLin ,
                bool pPlan ,bool Neg);
~ cCoordinateAxis();
void drawPrimitive(void);
int getAxisLenght ();
bool getGuideLines();
bool getNegatives();
bool getReferencePlanes();
int getTicks();
void setGuideLines(bool pVisible);
void setLenght(int pLongitud);
void setNegatives(bool pVisible);
void setReferencePlanes(bool pVisible);
void setTicks(int pMarcas);

```

4.9.26. Clase cScene.

cScene
<pre> private: numPrimitivas : int; description : string; marker1 : int; marker2 : int; primitivesGroups : vector <cGroups*>; lightPosition : int*; lightColor : int*; </pre>
<pre> public: cScena(void); ~cScena(void); int addPrimitive(cPrimitive *pIn); void cleanScene(void); void drawScene(); string getDescription(); vector <int> getMarkers(); int getNumPrimitivas(); vector <int> getLightPos(); vector <int> getLightColor(); void rebootScene(); void setDescription(string pDescription); void setMaxPrimitivas(int pNum); void setMarkers(int pMarker1, int pMarker2); void setLightPos(vector <int> pPos); void setLightColor(vector <int> pColor); </pre>

4.9.27. Clase cMultiChEx

<code>cMultiChEx: cScene</code>
<pre>private: explanation : string; marker3 : int; marker4 : int; selectedAnswer : int; correct : bool;</pre>
<pre>public: cMultiChEx(void); ~cMultiChEx (void); void drawScene(); string getExplanation(); vector <int> getMarkers(); void setExplanation(string pExplanation); void setMarkers(vector <int> pMarkers); void validateAnswer(); void solveExercise();</pre>

4.9.28. Clase cDrawingEx.

```
cDrawingEx: cScene
```

```
private:
```

```
explanation : string;
```

```
correct : bool;
```

```
tolerance : int;
```

```
tipo : int;
```

```
public:
```

```
DrawingEx();
```

```
~DrawingEx();
```

```
void dibujarEscena();
```

```
bool getCorrect();
```

```
int getTolerance();
```

```
int getType();
```

```
string getExplanation();
```

```
void setCorrect(bool pCorrect);
```

```
void setExplanation(string pExplanation);
```

```
void setTolerance(int pTolerance);
```

```
void setType(int pType);
```

```
void validateAnswer();
```

```
void solveExercise();
```

4.9.29. Clase cTransEx.

cTransEx: cScene
<pre>private: explanation : string; correct : bool;</pre>
<pre>public: TransfEx(); ~TransfEx(); void dibujarEscena(); bool getCorrect(); string getExplanation(); void setCorrect(bool pCorrect); void setExplanation(string pExplanation); bool validateAnswer(); void solveExercise();</pre>

4.9.30. Clase cSceneStore.

cSceneStore
<pre> private: scenesNumber : int; multiChoiceNumber : int; drawingExNumber : int; transExNumber : int; scenesSet : vector<Escena*>; multiChoiceSet : vector <MultiChEx*>; drawingSet : vector <DrawingEx*>; transSet : vector <TransEx*>; indexScene : int; indexMulti : int; IndexDrawing : int; indexTrans : int; </pre>
<pre> public: cSceneStore (); ~cSceneStore (); int addScene (Escena *pIn); int addMulti (MultiChEx *pIn); int addDrawing (DrawingEx *pIn); int addTransformation (TransEx *pIn); void drawScene (); void drawDrawing (); void drawMultiCh (); void drawTrans (); </pre>

4.9.31. Clase cGroups.

cGroups
<pre>private: operands : vector<Primitiva*>; result : vector<Primitiva*>; Actions : bool*;</pre>
<pre>public: cGroups(); ~cGroups(); void groupCalculatePartialTransformations(); void groupCalculateVectorSum(); void groupCalculateTransformation(int *order); void groupCalculateRotation(); void groupCalculateTraslation(); void groupCalculateScalation(); bool groupAplyTrans (int *order, int i); void drawOperands(); void drawResults();</pre>

4.9.32. Clase cFileHandling.

cFileHandling
<pre> public : fileHandling (); ~fileHandling (); cScene* createScene(ifstream *scn ,int St ,int *Con,int pTN); cMultiChEx* createMulti(ifstream *scn ,int St ,int *Con,int pTN); cDrawingEx* createDraw(ifstream *scn ,int St ,int *Con,int pTN); cTransfEx* createTransf(ifstream *scn ,int St ,int *Con,int pTN); cCoordinateAxis* getAxis(ifstream *scn ,int St , int *Con); cLabel* getLabel(ifstream *scn , int St , int *Con); void getLight(ifstream *scn ,int St ,int *Con); cCube* getCube(ifstream *scn ,int St ,int *Con); cCone* getCone(ifstream *scn ,int St ,int *Con); cCylinder* getCylinder(ifstream *scn ,int St ,int *Con); cSphere* getSphere(ifstream *scn ,int St ,int *Con); cVRMLObject* getVRMLObject(ifstream *scn ,int St ,int *Con); int loadScene(char pFile[100] , cSceneStore *scns ,int pTNum); cPoint* getPoint(ifstream *scn ,int St ,int *Con); cVector* getVector(ifstream *scn ,int St ,int *Con); void getActions(ifstream *scn ,int St ,int *Con,Groups *Group); void getGroups(ifstream *scn ,int St ,int *Con,Escena *Scene); void saveScene(Escena *pScene); </pre>

4.9.33. Estrategia de robustez ante cambios de iluminación.

En esta sección se describe el algoritmo que se ha implementado en el sistema para aumentar la estabilidad de la visualización de la escena ante los cambios de iluminación. En la figura 28 se puede observar el diagrama de estados que representa este algoritmo. El sistema se encuentra siempre en uno de los siguientes cuatro estados:

- Estado 0.- Buscar marcadores.
- Estado 1.- Examinar la escena (Cambiar punto de vista).
- Estado 2.- Observar modelo.

- Estado 3.- Realizar acción.
- Estado 4.- No hay marcadores.

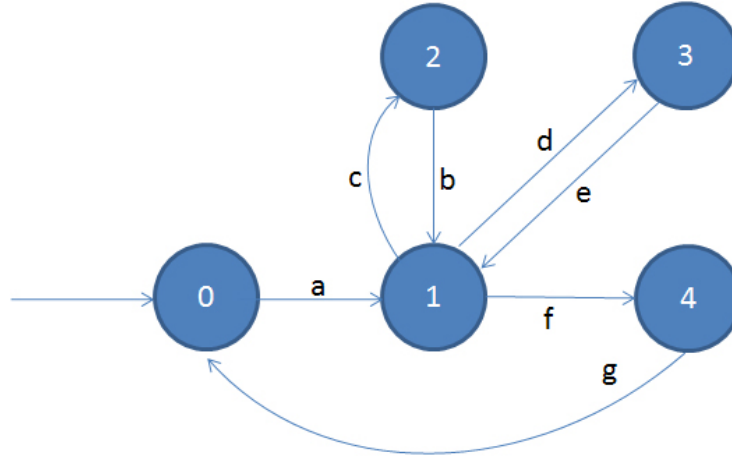


Figura 4.16: Grafo de estados.

El *estado 0* es el estado inicial, en este estado no hay marcadores capturados por la cámara y se emplea el algoritmo de detección de marcadores de ARToolKit.

En el *estado 1* se usa la matriz de transformación dada por el ARToolKit para configurarla como la Matriz de Vista - Modelo (ModelView Matrix) de OpenGL y dibujar los modelos correspondientes a la escena. Dicha matriz permite dibujar la escena alineada con respecto a los marcadores.

En el estado Observar el Modelo (*estado 2*), se fija la matriz de transformación, de manera que aunque el usuario cubra los marcadores, o estos desaparezcan por los cambios de iluminación, el modelo 3D permanecerá en la escena. El usuario no puede realizar modificaciones a la escena con ninguna de las herramientas. Mientras que el usuario no mueva la cámara, el punto de vista será correcto.

En el *estado 3*, Realizar acción, el usuario ha introducido un marcador correspondiente a las acciones permitidas en el sistema. El sistema fija automáticamente la matriz de transformación que se configura como la ModelView. El modelo 3D permanecerá visible mientras que el marcador de acción esté en la escena, sin importar que los marcadores de fondo no sean visibles.

El *estado 4* se activa cuando desaparecen los marcadores que estaban visibles en la escena. El modelo 3D permanece en la escena solo durante unos segundos. En la tabla 4.1 se describen las acciones que se toman en cada una de las transiciones entre estados.

Tabla 4.1: Descripción de las transiciones del grafo de estados.

Transición (es)	Disparador	Acción
<i>c</i>	El usuario activa el modo HOLD de la aplicación.	Se fija la matriz de transformación para que los cambios no afecten al modelo. Se toma la última matriz de transformación disponible.
<i>b, e</i>	<i>b</i> , el usuario desactiva el modo HOLD; <i>e</i> , el usuario retira de la escena la herramienta de trabajo.	Se utiliza la matriz de transformación que proporciona AR-ToolKit, se considera que el usuario está cambiando el punto de vista en la escena.
<i>d</i>	El usuario introduce una herramienta en la escena.	Se fija la matriz de transformación, el modelo permanecerá estático mientras haya una herramienta de trabajo dentro de la escena. Se trabaja bajo la premisa de que el usuario no cambiará el punto de vista mientras este modificando la escena.
<i>f</i>	Mientras el usuario examina la escena, los marcadores dejan de ser capturados por la cámara.	Como no hay marcadores para calcular la posición y orientación de la cámara, se fija la última matriz de transformación disponible. El modelo permanecerá en la escena por un determinado tiempo, después del cual desaparecerá.
<i>g</i>	Se sobrepasa el umbral de tiempo sin que la cámara capture un marcador.	Se aplica el algoritmo de detección de marcadores de ARToolKit. Si se detecta algún marcador en la escena, se utiliza la matriz de transformación.

Capítulo 5

Resultados

En esta sección se presentan los resultados de los primeros tests de usabilidad del sistema. Por usabilidad se entiende lo fácil que resulta usar el sistema y aprender a usarlo.

En estas pruebas participaron 12 personas con un promedio de edad de 26 años, que estudian en diferentes programas de posgrado de la Universidad Politécnica de Valencia. De los 12 usuarios encuestados, 7 son hombres y 5 son mujeres. El sistema se ha probado con usuarios no acostumbrados a trabajar con aplicaciones de RA. De los 12 usuarios 5 habían oído hablar de esta tecnología y 7 de ellos desconocían totalmente su existencia.

Antes de la prueba, los usuarios recibieron una charla de introducción de aproximadamente 10 minutos. En esta charla se les explicó el objetivo de las pruebas y el funcionamiento básico del sistema. Se les hizo una demostración de las funcionalidades de la herramienta y se les explicó las tareas a realizar.

El experimento consistió en una serie de 5 ejercicios en los que los usuarios debían alcanzar ciertas metas. Los ejercicios incluyeron las acciones básicas dentro del sistema: cambiar entre una escena y otra, dibujar primitivas, borrar primitivas, aplicar transformaciones y seleccionar elementos simulando el clic sobre los marcadores. Después de un promedio de 25 minutos trabajando con el sistema, se les pasó un cuestionario sobre su experiencia.

De acuerdo con las encuestas realizadas, a los usuarios les resultó más fácil manipular las herramientas del sistema cuando se encontraban detrás de la

cámara, preferentemente del lado de la mano con la que el usuario escribe.

Tabla 5.1: Resultados de la encuesta.

promedio de las calificaciones dadas por los usuarios.		
Concepto	Tamaño del patrón (cm)	Rango de usabilidad (cm)
Dibujar primitivas	3.76	4.23
Cambiar entre escenas	4.54	
Cambiar el punto de vista	3.91	
Resolver ejercicios	4.00	

Después de usar el sistema se les pidió que contestaran una encuesta para evaluar aspectos pedagógicos y de usabilidad. La escala de evaluación es de 1 al 5, siendo 1 lo más bajo y 5 la calificación más alta. La tabla 5.1 muestra los resultados. La mayoría de los resultados están entre el rango de 4 a 5. Sin embargo, esto puede ser mejorado.

Algunos alumnos, a pesar de la buena nota, manifestaron incomodidad por el hecho de tener que cambiar los marcadores para cambiar de escena. Ya se está trabajando en la búsqueda de una solución viable para este problema. El módulo de auto evaluación recibió buenas notas, ya que los usuarios manifestaron encontrarlo fácil de usar y sobre todo de mucha utilidad para reafirmar los conceptos presentados.

Con respecto a la componente educacional de la evaluación del sistema, también se obtuvieron buenos resultados, 4.23. Los usuarios expresaron su interés por extender el sistema a otras áreas de conocimiento. Consideran que la tecnología podría resultar útil para aprender estos y otros conceptos. Las conclusiones preliminares que se pueden extraer de dichos cuestionarios son que a la mayoría de los usuarios que probaron el sistema les pareció una idea novedosa. Así, dichos usuarios creen que el uso del sistema facilita el aprendizaje y la mayoría usaría el sistema si tuvieran que estudiar los conceptos tratados en él. También se ha llegado a la conclusión de que hay que mejorar la robustez del sistema frente a los cambios de iluminación.

Es importante mencionar que de acuerdo a los resultados obtenidos, la no-

vedad de la tecnología usada parece contribuir al objetivo planteado en este trabajo de motivar al alumno a usar la herramienta y por lo tanto a practicar los conceptos que en ella se presentan.

En la figura 5.1 se puede observar una comparación de las características del sistema que se propone en este trabajo y los sistemas que han sido revisados en el capítulo de antecedentes.

Además de comparar los tipos de paradigmas educativos utilizados, se comparan otras características relevantes. Se considera que en menor o mayor grado todos los sistemas son mono usuario, por lo que en la tabla se señala como característica la opción de trabajo de varios usuarios al mismo tiempo en el sistema. También se muestra si los sistemas pueden trabajar de manera distribuida, puesto que todos los sistemas son capaces de trabajar de manera local. Finalmente, se considera que un sistema requiere hardware y/o instalaciones especiales, si para su uso se requiere elementos como dispositivos de seguimiento especiales, dispositivos HMD, un área específica para su instalación o algún dispositivo físico de coste elevado.

Sistema	Clase magistral	Sistemas colaborativos	Autoaprendizaje	Aprendizaje lúdico	Multiusuario	Distribuido	Requiere Hardware y/o instalaciones especiales
MARIE	✓	x	x	x	✓	✓	x
Kondo06	✓	x	x	x	✓	x	x
Web3D-RA	✓	x	x	x	✓	✓	x
Construct3D	x	✓	x	x	✓	x	✓
ARiSE	x	✓	x	x	✓	x	✓
CiberMath	x	x	✓	x	✓	✓	x
Nuñez08	x	x	✓	x	✓	x	✓
Química aumentada	x	x	✓	x	x	x	x
Cuerpo humano	x	x	x	✓	x	x	x
Cubos magnéticos	x	x	x	✓	x	x	x
Encontrar parejas	x	x	x	✓	x	x	x
Reciclar	x	x	x	✓	x	x	✓
Rey Leon	x	x	x	✓	x	x	x
MI PROPUESTA	✓	x	✓	x	✓	x	x

Figura 5.1: Comparación de características.

Comparación de las características de los sistemas revisados.

Capítulo 6

Conclusiones

Se ha presentado en este trabajo un sistema educativo que utiliza las técnicas de Realidad Aumentada para enseñar conceptos geométricos. El sistema permite al profesor crear ejercicios que puede resolver frente a la clase o bien les puede entregar a los alumnos para que los resuelvan en casa o en el laboratorio.

Un grupo de estudiantes de diferentes programas de postgrado de la Universidad Politécnica de Valencia ha evaluado el sistema. Se han obtenido buenos resultados en las pruebas pedagógicas y de usabilidad. El sistema causó una buena impresión en los usuarios que lo encontraron motivante y útil para aprender geometría y algunas otras disciplinas.

Con respecto a las mejoras que se pretenden introducir a corto plazo, cabe mencionar la introducción del módulo de reconocimiento de comandos por voz para la simplificación de la interacción del usuario con el sistema. Este nuevo módulo contemplaría comandos simples para algunas tareas como: guardar una escena, borrar primitivas o resolver un ejercicio, por citar algunos ejemplos.

Para dotar al sistema de mayor flexibilidad y facilitar al usuario la generación de escenas se piensa desarrollar una herramienta de autor más intuitiva, que facilite al profesor la creación y visualización de escenas antes de cargarlas en el sistema. Por otro lado, también se hace necesaria la evaluación más profunda del sistema involucrando más alumnos y profesores. Se ha planeado usar la aplicación como herramienta didáctica en las asignaturas básicas de introducción a los gráficos por computador de la Escuela Técnica Superior de Ingeniería

Informática de la Universidad Politécnica de Valencia. Se evaluarán aspectos de usabilidad y diseño de la aplicación para validar si el sistema logra su propósito pedagógico. Se evaluará si la herramienta ayuda a los estudiantes a comprender mejor conceptos como las transformaciones afines, al mismo tiempo que se convierte en una herramienta accesible y fácil de usar para el profesor.

Bibliografía

- [1] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [2] A. Balog, C. Pribeanu, and D. Iordache. Augmented reality in schools: Preliminary evaluation results from a summer school. In *Proceedings of WASET International Conference on Technology and Education - ICTE 2007*, pages 114–117, Oct 2007.
- [3] M. Billinghurst, H. Kato, and I. Poupyrev. The magicbook - moving seamlessly between reality and virtuality. *Computer Graphics and Applications, IEEE*, 21(3):6–8, May/June 2001.
- [4] M. Bogen, J. Wind, and A. Giuliano. ARiSE - augmented reality in school environments. In *First European Conference on Technology Enhanced Learning, EC-TEL 2006*, pages 709–714, Crete, Greece, 2006.
- [5] M. Breisinger, T. Höllerer, J. Ford, and D. Folsom. Implementation and evaluation of a 3d multi modal learning environment. In Elaine Pearson and Paul Bohman, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006*, pages 2282–2289, Chesapeake, VA, June 2006. AACE.
- [6] C. Carlsson and O. Hagsand. Dive a multi-user virtual reality system. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 394–400, Sep 1993.
- [7] S. Castillo. Propuesta pedagógica basada en el constructivismo para el uso óptimo de las tic en la enseñanza y el aprendizaje de la matemática. In *RE-*

- LIME. Revista latinoamericana de investigación en matemática educativa*, volume 11, pages 171 – 194, 2008.
- [8] S. Cawood and M. Fiala. *Augmented Reality: A Practical Guide*. Pragmatic Bookshlef, 2008.
- [9] M. Fjeld, J. Fredriksson, M. Ejdestig, F. Duca, K. Býtschi, Benedikt M. Voegtli, and P. Juchli. Tangible user interface for chemistry education: comparative evaluation and re-design. In *Special Interest Group on Computer-Human Interaction (SIGCHI)*, pages 805–808, 2007.
- [10] M.C. Juan, F. Beatrice, and J. Cano. An augmented reality system for learning the interior of the human body. In *The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'08)*, pages 186–188, 2008.
- [11] M.C. Juan, R. Canu, and M. Giménez. Augmented reality interactive storytelling systems using tangible cubes for edutainment. In *The 8th IEEE International Conference on Advanced Learning Technologies (ICALT 08)*, pages 233–235, 2008.
- [12] M.C. Juan, D. Furió, R. Mollá, M. J. Vicent, R. Vivó, and M. Giménez. Edutainment games included as activities in the summer school of the technical university of valencia. In *Game-On*, pages 147–151, Valencia, España, 2008.
- [13] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality, conferencing system. In *2nd IEEE and ACM International Workshop on Augmented Reality (IWAR 99)*, 1999. <http://www.hitl.washington.edu/artoolkit>.
- [14] T Kondo. Augmented learning environment using mixed reality technology. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006*, pages 83–87, 2006.
- [15] F. Liarokapis, N. Mourkoussis, M. White, J. Darcy, M. Sifniotis, P. Petridis, A. Basu, and Paul F. Lister. Web3d and augmented reality to support

- engineering education. *World Transactions on Engineering and Technology Education*, 3(1):11–14, 2004.
- [16] F. Liarokapis, P. Petridis, Paul F. Lister, and M. White. Multimedia augmented reality interface for e-learning (MARIE). *World Transactions on Engineering and Technology Education, UICEE*, 1:173–176, 2002.
- [17] B. Macintyre, M. Gandy, J. Bolter, S. Dow, and B. Hannigan. Dart: the designer’s augmented reality toolkit. *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, pages 329–330, 2003.
- [18] A. MacWilliams, C. Sandor, M. Wagner, M. Bauer, G. Klinker, and B. Bruegge. Herding sheep: Live system development for distributed augmented reality. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, page 123, 2003.
- [19] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D(12), December 1994.
- [20] M. Nuñez, R. Quirós, I. Nuñez, Juan B. Carda, and E. Camahort. Collaborative augmented reality for inorganic chemistry education. In *New aspects of engineering education, Proceedings of the 5th WSEAS*, pages 271–277, Jul 2008.
- [21] S. Rakov, T. Oleinik, and S. Oleinik. Computer explorations in math courses (geometry, algebra and calculus). In *ICTMT - 3: Technology in Mathematics Teaching*, 1998.
- [22] G. Roveló, F. Abad, M. Juan, and E. Camahort. A low cost augmented reality system to help students learn geometric concepts. In *Joint Virtual Reality Conference of EGVE - ICAT - EuroVR. Enviado*, 2009.
- [23] G. Roveló, F. Abad, M. Juan, and E. Camahort. Sistema de realidad aumentada para enseñanza de geometría. In *Congreso Español de Informática Gráfica (CEIG’09)*, Sep 2009.

- [24] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalávári, L. Encarnao, Gervautz M., and W. Purgathofer. The studierstube augmented reality project. *resence: Teleoper. Virtual Environ.*, pages 33–54, 2002.
- [25] D. Schmalstieg, A. Fuhrmann, Zs. Szalavári, and M. Gervautz. "studierstube an environment for collaboration in augmented reality. In *Proceedings of Collaborative Virtual Environments '96*, pages 19–20, 1996.
- [26] D. Schmalstieg, M. Wagner, H. Kaufmann, and L. Gasse. Construct3d: a virtual reality application for mathematics and geometry education. *Education and Information Technologies*, 5:263–276, 2000.
- [27] B.E. Shelton and N.R. Hedley. Using augmented reality for teaching earth-sun relationships to undergraduate geography students. In *First IEEE International Augmented Reality Toolkit Workshop*, Darmstadt, Germany, 2002.
- [28] A. Spalter and R. Simpson. Integrating interactive computer-based learning experiences into established curricula: a case study. In *ITiCSE '00: Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and technology in computer science education*, pages 116–119, 2000.
- [29] Zs. Szalavári and M. Gervautz. The personal interaction panel - a two-handed interface for augmented reality. *COMPUTER GRAPHICS FORUM*, 16(3):335–346, 1997.
- [30] G. Taxén and A. Naeve. Cybermath: Exploring open issues in vr-based learning. *SIGGRAPH 2001 Conference Abstracts and Applications*, pages 49–51, 2001.

Capítulo 7

Apéndice

El trabajo presentado está relacionado hasta hoy con las siguientes publicaciones:

[23] En este trabajo se presentan la primera versión del sistema y los primeros resultados de usabilidad obtenidos en los test aplicados a usuarios del sistema. También se describen las mejoras que se realizarán a corto plazo en el sistema. Este trabajo está aceptado y se presentará y publicará en el Congreso Español de Informática Gráfica (CEIG'09).

[22] Se presentan las mejoras realizadas al sistema y los cambios que tenemos previsto añadir al trabajo en el futuro. Se describe la Interfaz gráfica de usuario que se ha desarrollado para la descripción de las escenas y cuál es el impacto que tiene en el funcionamiento general de la herramienta. Este trabajo ha sido enviado al Joint Virtual Reality Conference, EGVE - ICAT - EuroVR 2009 y está pendiente de aceptación (fecha prevista de la comunicación: 10 de Julio).

Sistema de Realidad Aumentada para enseñanza de Geometría

Gustavo Rovelo¹, Francisco Abad², Mari Carmen Juan² and Emilio Camahort²

¹Departamento de Sistemas Informáticos y Computación

²Instituto Universitario de Automática e Informática Industrial

Universidad Politécnica de Valencia

Camino de Vera S/N 46022 Valencia, España

Abstract

Este artículo presenta el diseño e implementación de un sistema de Realidad Aumentada orientado a la enseñanza de conceptos abstractos en Geometría. Este sistema permite mostrar al alumno escenas 3D interactivas, que pueden ser exploradas desde multitud de puntos de vista y que le ayudan a mejorar su intuición espacial.

El sistema se puede utilizar en diferentes escenarios sin necesidad de hardware especializado. Se puede utilizar tanto en el aula, como herramienta de apoyo al profesor, como en casa, sirviendo de instrumento para el estudio personal.

El profesor puede crear nuevos ejercicios mediante una sencilla descripción de escenas basada en XML, y el sistema es capaz de corregirlos automáticamente, dando al alumno retroalimentación en tiempo real sobre sus respuestas.

En este artículo también presentamos la configuración idónea del entorno de trabajo derivada de nuestra experiencia. También presentamos una propuesta de utilización de distintos materiales para construir los marcadores que mejoran el proceso de detección de los mismos.

Keywords: Realidad Aumentada, Enseñanza de las Matemáticas, Intuición espacial.

Categories and Subject Descriptors (according to ACM CCS):

I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality

I.3.8 [Computer Graphics]: Applications

1. Introducción

La enseñanza de las matemáticas ha representado siempre un reto para los profesionales de la educación. Convertir un concepto abstracto en palabras y ejemplos claros que permitan al alumno asimilarlo no es una tarea fácil, y no siempre se consigue. Además, se debe tener en cuenta que no todos los alumnos tienen la misma capacidad para comprender las explicaciones de cada tema.

Además de los factores humanos en el proceso de enseñanza, la manera tradicional de impartir las clases de matemáticas se ha basado únicamente en la pizarra y modelos 2D dibujados en ella. Representar, por ejemplo, vectores en el espacio o volúmenes de sólidos es difícil de conseguir y,

muchas veces, difícil de entender para el alumno. En algunos casos se pide al alumno que construya maquetas con los modelos 3D que se ven en clase, o bien algún concepto como la suma de dos vectores. Esto facilita la comprensión del tema, pero en cambio exige una inversión de tiempo que se podría utilizar para estudiar conceptos más avanzados.

Por ello se deben proponer soluciones que permitan a los profesores contar con las herramientas necesarias para que los alumnos asimilen mejor los conceptos que por su naturaleza son difíciles de comprender. Así mismo se debe ofrecer a los alumnos la posibilidad de interactuar con los modelos presentados para conseguir a partir de la experiencia que adquieran su propia visión o intuición espacial.

Poco a poco se han ido incorporando nuevas tecnologías al aula para facilitar el proceso de la enseñanza. Los proyectores de diapositivas permiten al profesor mostrar material que refuerza su explicación. El uso de proyectores digitales conectados a computadoras permite al profesor usar software especializado en presentaciones y en ocasiones usar software interactivo, como por ejemplo *applets* Java.

Sin embargo, aún cuando este tipo de tecnología está cambiando la manera de impartir clase, en algunos casos sigue siendo insuficiente para facilitar la comprensión de los conceptos más abstractos. La Realidad Aumentada (RA) combina tecnología proveniente de los campos de visión por computador, reconocimiento de patrones y síntesis de imágenes. Su objetivo principal es superponer información útil sobre la percepción que el usuario tiene de la realidad. Dicha información viene codificada típicamente en forma de imágenes generadas por ordenador y permite enriquecer la experiencia del usuario.

De acuerdo con Azuma [Azu97] un sistema de RA debe cumplir con las siguientes características:

1. Combinar información real con información virtual
2. Debe permitir la interacción del usuario con el sistema en tiempo real
3. El proceso de registro de las imágenes sintéticas deber ser en 3D

De acuerdo con la definición anterior el dispositivo de visualización utilizado no influye en la clasificación de un sistema como RA, siempre y cuando se respeten las características antes mencionadas.

La RA en suma ha propuesto una nueva metodología de trabajo que permite gozar de algunos de los beneficios de los entornos virtuales con un significativo menor coste económico y por supuesto ganando en movilidad. Desde cuentos infantiles [BKP01] hasta complejos sistemas distribuidos pensados para la educación a distancia [LPLW02], utilizan las ventajas de la RA para proporcionar herramientas de trabajo novedosas con el objetivo de facilitar la enseñanza y captar la atención del alumno. Otra aplicación destacada en la que se ha utilizado la RA es el sistema propuesto en [SH02] para la enseñanza de la relación entre la Tierra y el Sol.

El presente trabajo propone la utilización de técnicas de RA para facilitar la visualización y manipulación de modelos geométricos en el espacio 3D. La idea principal es evitar una representación plana (el dibujo en la pizarra) de lo que realmente ocupa un volumen en el espacio. Proponemos una herramienta que permite al alumno interactuar con modelos virtuales usando elementos reales (los marcadores). Así podrá mejorar su percepción espacial viendo e interactuando con los modelos en el espacio 3D. Además, nuestro sistema se ha diseñado para usarse fundamentalmente con una webcam y un ordenador de escritorio o portátil, sin necesidad de instalaciones especiales. Esto permite su utilización tanto en el aula como en casa.

Nuestra herramienta es por tanto versátil y económica. En este artículo describimos la herramienta y resultados preliminares de su utilización por parte de un grupo de alumnos. La herramienta permite crear, animar, borrar y manipular objetos así como realizar operaciones geométricas como sumar vectores. Para el profesor nuestro sistema permite configurar modelos, ejercicios y soluciones a esos ejercicios. Nuestros tests demuestran que se trata de una aproximación eficaz a la enseñanza de la geometría del espacio 3D.

El resto del artículo se organiza como sigue. La siguiente sección presenta el estado del arte en RA aplicada a la enseñanza. En las dos secciones siguientes se enumeran las características del sistema y se muestra su estructura. En el siguiente apartado se muestra un ejemplo de uno de los ejercicios que se pueden implementar en el sistema, y a continuación se discuten los resultados de los tests preliminares en los que han participado usuarios. Finalmente se presentan las conclusiones y describen las mejoras que se han de implementar en la aplicación, como parte del trabajo futuro.

2. Antecedentes

Esta sección revisa los trabajos más relevantes en el uso de la Realidad Aumentada aplicada al campo de la educación. Estos sistemas, tienen características comunes, como el hecho de ser sistemas monousuario o multiusuario. Otro criterio de clasificación es que algunos funcionan de manera distribuida y otros funcionan solo de manera local. Por último, también se pueden clasificar según áreas de interés. Sin embargo, así como estos sistemas comparten ciertas características, existe una clara diferencia en el paradigma utilizado para enseñar. De esta manera se pueden clasificar en cuatro paradigmas diferentes. Los paradigmas son: Clase magistral, Sistemas colaborativos, Autoaprendizaje y juegos infantiles.

2.1. Clase magistral

Las Clases magistrales, son el paradigma educativo tradicional. En ellas es el profesor quien lleva el control de la clase. El alumno permanece como receptor pasivo de la información que se le proporciona.

Un buen ejemplo de los sistemas que siguen el paradigma de las clases magistrales guiadas por el profesor es el *Multimedia Augmented Reality for E-Learning* (MARIE) [LPLW02]. Diseñado como una herramienta para la educación a distancia de ingeniería. El sistema permite al usuario seguir el guión con el contenido del curso, según el diseño del profesor. También permite a los usuarios interactuar con información tanto 2D como 3D y ver al mismo tiempo las explicaciones del profesor. Se puede usar con un HMD o con un monitor de ordenador, por lo que se puede usar por más de un estudiante a la vez, o en el aula. También incorpora un módulo de sonido que puede mezclar dos fuentes: la explicación pregrabada y la voz del profesor. Uno de sus inconvenientes es que la interacción con el sistema se

basa en la combinación de la manipulación de los marcadores y el uso del teclado y el ratón, lo que puede resultar incómodo para algunos usuarios.

El trabajo presentado en [Kon06] describe tres desarrollos aplicados a la educación que hacen uso de la RA. El primer ejemplo es un libro de texto enriquecido. En este sistema se superponen gráficos 3D sobre un libro de texto tradicional para mejorar la explicación del concepto. Para ello utilizan una cámara web para identificar el patrón impreso sobre la hoja del libro. El segundo sistema se dedica a la enseñanza de Matemáticas en el entorno de un aula. El profesor da explicaciones sobre un modelo que los alumnos pueden ver en pantalla. Utilizando una cámara web y los marcadores impresos y colocados sobre su escritorio, el profesor puede interactuar con los modelos virtuales de sólidos como si se tratara de modelos reales. El último sistema permite enriquecer el esqueleto de dinosaurios con información gráfica sobre cómo podría ser su cuerpo. Además, también permite a los usuarios experimentar con algunos aspectos que todavía están en estudio, como puede ser el color de la piel de algunas especies. El artículo también presenta el desarrollo de un sistema de autor para preparar esta clase de contenido disminuyendo la necesidad de conocimientos previos sobre modelado 3D.

El sistema descrito en [LMW*04] combina Web3D y la RA para construir un sistema de ayuda a la enseñanza de conceptos de ingeniería mecánica. Web3D es un conjunto de aplicaciones que implementan el estándar X3D para transmisión de escenas y de objetos 3D a través de internet usando XML. Normalmente se usan plugins instalados en un navegador web para visualizar estos modelos. Web3D permite al usuario interactuar con modelos 2D y 3D que pueden estar animados, y puede seleccionar qué contenido le resulta más interesante para seguir estudiando. El módulo de RA está embebido en el navegador y funciona como una aplicación estándar de RA y hace uso de la biblioteca Microsoft Foundation Class (MFC) para crear su interfaz de usuario. El sistema permite usar modelos almacenados en formatos estándar. Este sistema se puede utilizar a través de Internet o de manera local durante una clase o en el laboratorio.

2.2. Sistemas colaborativos

En los sistemas colaborativos se hace énfasis en el trabajo en grupo. Se favorece la interacción entre compañeros, para conseguir que entre varios se resuelvan problemas complejos y se genere un conocimiento en base a la experiencia colectiva.

El primero de estos sistemas y quizás uno de los más importantes es Construct3D [SWK*00], un sistema de Realidad Aumentada construido sobre la plataforma colaborativa Studierstube [SFSG96]. Este sistema permite a varios usuarios provistos de cascos HMD trabajar concurrentemente sobre la misma tarea. Según los autores, Construct3D no es

una herramienta de diseño 3D profesional. El objetivo del trabajo era crear un interfaz de usuario amigable utilizando el Personal Interaction Panel (PIP) [SG97]. Este panel permite al usuario interactuar con objetos virtuales a través de una interfaz tangible y de un menú en el que puede seleccionar opciones y ejecutar acciones. Esta plataforma se diseñó originalmente para la enseñanza de geometría. Construct3D necesita un espacio dedicado para instalar el equipo, y localiza el origen del sistema de coordenadas virtual en el centro de la habitación. El sistema puede trabajar con líneas, puntos, planos y algunos volúmenes sólidos como esferas y conos. También es posible guardar las escenas en ficheros y tiene un subsistema de audio que asiste al usuario en el desarrollo del ejercicio. Por medio de encuestas, los autores encontraron que algunos usuarios preferían los sistemas CAD de escritorio tradicionales. Sin embargo, los mismos usuarios valoraban la facilidad de uso de Construct3D y lo encontraban útil como herramienta de aprendizaje.

ARiSE (Augmented Reality in School Environments) [BWG06] es una plataforma construida con el esfuerzo conjunto de varias universidades europeas, y su objetivo es facilitar la transmisión de conocimientos al estudiante. El sistema permite a varios estudiantes interactuar y trabajar en equipo. También incluye las herramientas necesarias para que un usuario no experto en RA pueda usarlo y generar contenido fácilmente. Los resultados de las pruebas de usabilidad fueron descritos en [BPI07]. El objetivo de dichas pruebas fue evaluar la usabilidad del sistema y la calidad pedagógica del material presentado por el sistema. Por usabilidad se entiende lo fácil que resulta el sistema de usar y de aprender a usarlo. Por calidad pedagógica entendemos en qué medida el sistema ayuda a mejorar el proceso de aprendizaje. De forma parecida a otros estudios de este tipo, se encontró que la novedad de esta tecnología estimula al estudiante a utilizar la herramienta, encontrándola divertida y motivante. Sin embargo, dichas pruebas revelaron que los usuarios esperaban algo más del sistema, especialmente con respecto a la retroalimentación cuando se ejecuta una acción. El usuario esperaba que de alguna manera el sistema le informara si la acción desarrollada cumplió su objetivo o no. Los tests de usabilidad mostraron también la necesidad de mejorar la interacción con el usuario, ya que hubo problemas de precisión usando la herramienta. Cabe mencionar que ninguno de los alumnos que participó en las encuestas tenía conocimientos previos de RA.

2.3. Autoaprendizaje

En el paradigma de auto aprendizaje, se le da total libertad al alumno para que genere conocimiento en base a lo que investiga. El profesor se convierte en un guía de su aprendizaje, orientándolo.

CyberMath [TN01] es un ejemplo de un sistema que soporta diferentes paradigmas educativos. Sin embargo, su diseño se inspiró en un museo, en el que cada sala presenta

un tema distinto. Así, el alumno puede repasar cada tema sin la ayuda de un profesor. Es un entorno virtual basado en la plataforma DIVE [CH93], que se puede usar en un entorno inmersivo tipo CAVE o con un HMD. Permite trabajar a varios usuarios a la vez replicando el contenido en cada máquina cliente, conectadas a través de Internet. También dispone de aulas, en las que avatares simulan la presencia de los usuarios. El uso de gráficos realistas permite que la aplicación sea más atractiva para el usuario. El sistema permite importar modelos o animaciones construidos en Mathematica, lo cual le da flexibilidad de uso. Como resultado de las pruebas de usabilidad del sistema, se observó que en algunas ocasiones el uso de los avatares obstruía la visibilidad del usuario.

Otra aplicación que hace uso de técnicas de RA en la educación con un enfoque de autoaprendizaje es la presentada por Núñez y otros [NQ*08]. En este sistema se presentan modelos tridimensionales de compuestos inorgánicos descritos en VRML para mejorar la percepción espacial del estudiante. El sistema usa un PC normal con seis cámaras USB conectadas mediante un concentrador para permitir el trabajo simultáneo de varios usuarios, de manera individual o por pequeños grupos. La secuencia de video obtenida por cada cámara se reproduce en una ventana independiente, para permitir que todos los alumnos puedan observar el trabajo de sus compañeros. Para ello se utiliza el proyector del aula. Los resultados obtenidos tras varias pruebas en la Universidad Jaume I de Castellón, España, muestran que los alumnos encuentran el sistema muy interesante y permite captar su atención e incrementar su motivación.

En [FFE*07] se presenta la evaluación de dos versiones de un sistema de "Química aumentada". Se ha diseñado para interactuar y componer modelos tridimensionales de moléculas orgánicas a partir de modelos almacenados en una base de datos. El sistema permite crear moléculas usando herramientas representadas mediante marcadores 2D, imitando el modelo de bolas y varillas usado tradicionalmente en la enseñanza de Química. El sistema se probó con estudiantes de secundaria, que en su mayoría lo encontraron atractivo y fácil de usar a pesar de que interactuar con el sistema resultaba más complejo que con el modelo de bolas y varillas tradicional.

2.4. Juegos infantiles

Finalmente, el aprendizaje lúdico, como su nombre lo dice, es aprender jugando. Hacer que los conceptos que se estudian lleguen al alumno a través de actividades divertidas.

Algunos sistemas de este tipo se han desarrollado dentro de nuestro grupo de investigación. Uno de ellos es el Sistema de Realidad Aumentada para conocer el interior del cuerpo humano [JB08]. El sistema utiliza interfaces tangibles para que el usuario pueda aprender la estructura anatómica del interior del cuerpo humano. El maniquí utilizado consiste en

un cuadro de madera sobre el que se ubican los marcadores y una tela a la que se le hicieron cortes que pueden ser abiertos y cerrados con cremalleras. De esta manera el usuario utilizando un HMD puede al abrir el cierre asomarse al interior del cuerpo, observando en la parte baja del abdomen la zona de los intestinos, o bien en la parte alta, la zona del estómago. En las pruebas realizadas durante la Escuela de Verano de la UPV, participaron 40 niños entre 8 y 10 años, que encontraron el sistema fácil de usar y estimulante para su aprendizaje, y demostraron interés por utilizarlo en otras áreas de conocimiento.

Finalmente, en [JFM*08] se presentan tres juegos de RA enfocados a la concienciación de los niños en temas de conservación del medio ambiente. En las pruebas de estos tres juegos participaron 160 niños de la Escuela de verano de la UPV. En el primer juego *Magnetic Cubes* el usuario debe encontrar alguno de los animales en peligro de extinción que contempla el sistema, y si acierta, puede elegir si desea ver un video con información sobre dicho animal. Usa como interfaz tres cubos que se mantienen unidos uno a otro magnéticamente. A través de la combinación de los marcadores colocados en las caras visibles de los cubos el usuario realiza las acciones necesarias para interactuar con el sistema. El siguiente juego denominado *Finding pairs*, es el clásico juego de memoria, en donde el usuario debe encontrar la tarjeta que completa el par con el animal en peligro de extinción mostrado. Si el usuario acierta, tiene la posibilidad de ver un video con información relevante sobre la especie que ha encontrado. El tercer juego, fue desarrollado para funcionar en un teléfono móvil, y en él el usuario debe identificar la imagen que aparece sobre el marcador, para de esa manera colocarlo en el cesto de basura que corresponda a la categoría de reciclado adecuada. En total se usan 5 marcadores diferentes, uno para cada contenedor de basura, y uno sobre el que se muestra el objeto a clasificar.

Todos los sistemas presentados son muy completos. Sin embargo, en algunos casos, el coste de la infraestructura del sistema los hace inalcanzables para un presupuesto reducido. Otro problema importante es que en algunos de estos sistemas el alumno permanece como espectador en el proceso de aprendizaje. Con el diseño de la aplicación que presentamos queremos afrontar estos problemas. Damos al alumno buena parte del protagonismo en el proceso del aprendizaje. Por otra parte, el profesor dispone de una herramienta versátil para mejorar la calidad de sus explicaciones y evaluar al alumno. Y sobre todo, pensamos en un diseño austero que permita implantar el sistema sin invertir una fuerte cantidad de dinero.

3. Descripción del sistema

Presentamos aquí un sistema de RA de bajo coste para facilitar el entrenamiento de la percepción espacial, en el que el alumno se convierte en sujeto activo durante su aprendizaje.

En las siguientes secciones se muestran los detalles de la implementación del sistema. De acuerdo con nuestra experiencia, la forma más cómoda para trabajar con el sistema es la que se muestra en la figura 1.

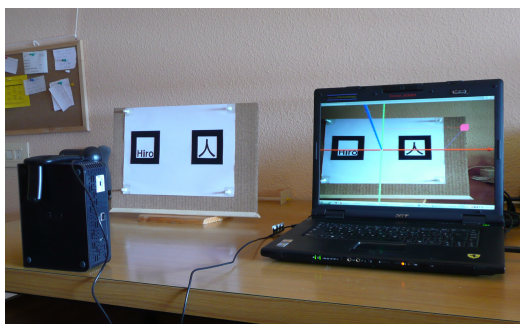


Figura 1: Configuración aconsejada del espacio de trabajo.

Nuestro sistema está diseñado de manera que se pueda utilizar por todos los alumnos en casa. Por ello, los requisitos de hardware se limitan a un PC actual y una cámara web. El resto de los componentes del sistema se pueden construir a partir de materiales comunes disponibles en el mercado.

El sistema permite construir escenas con las siguientes primitivas: sistemas de coordenadas 3D, puntos, vectores, modelos 3D (en VRML) y volúmenes sólidos como conos, esferas, cajas, cilindros y planos. El sistema también permite definir grupos de primitivas, sobre los que aplicar acciones.

Dichas acciones pueden ser: situar un objeto en la escena, dibujar primitivas usando la herramienta de dibujo del sistema, aplicar las transformaciones básicas definidas en la descripción de la escena (traslaciones, rotaciones y escalados), seleccionar entre varias opciones para responder las preguntas planteadas, reiniciar el trabajo con la escena eliminando las primitivas que ha dibujado y guardar el trabajo que ha realizado para utilizarlo en una sesión posterior. Cada una de las acciones se puede asociar a un marcador distinto, que el usuario deberá seleccionar y mover dentro de la escena. Por supuesto, el usuario podrá siempre cambiar el punto de vista de la escena moviendo la cámara.

3.1. Tipos de ejercicios

Otro aspecto importante en cualquier aplicación de enseñanza asistida por ordenador es la posibilidad de realizar evaluaciones en tiempo real de los ejercicios. El sistema permite implementar tres tipos de ejercicios, donde el alumno debe:

- elegir una entre varias respuestas,
- dibujar la respuesta, y
- dar la secuencia de transformaciones necesarias para transformar un estado inicial de la escena en un estado final dado.

El primer tipo de ejercicio es el de respuesta múltiple, donde el profesor plantea una pregunta sobre la escena presentada, y el alumno debe elegir entre una de cuatro posibles respuestas. El sistema entonces puede comprobar si la respuesta es correcta y, en caso contrario, ofrecer una explicación adicional sobre la respuesta esperada (véase la figura 2).

Para elegir una de las opciones, el usuario deberá cubrir momentáneamente el marcador, ya sea parcial o totalmente, mientras que se encuentra sobre la respuesta elegida. Para cubrir el marcador, el usuario puede usar simplemente un dedo, tal y como muestra la figura 2. De esta forma se simula el clic de ratón en una interfaz tradicional.

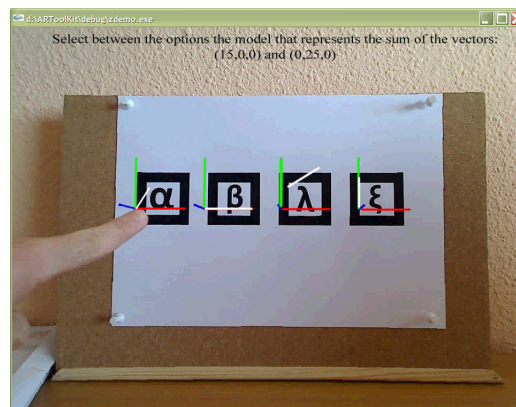


Figura 2: Ejercicio de respuesta múltiple; los ejes están dibujados en rojo x, en verde y y en azul z en un sistema de coordenadas dextrógiro.

En el segundo tipo de ejercicios, el estudiante debe dibujar la primitiva en la posición y orientación que responda al problema planteado. Por ejemplo, ante un ejercicio de suma de vectores, el alumno deberá seleccionar el marcador para dibujar un vector y calcular el vector resultado.

Un problema que plantea este tipo de ejercicios es la precisión del dibujado a mano alzada en un espacio 3D. El sistema ofrece una serie de ayudas para situar un objeto correctamente en 3D. En particular, el sistema permite mostrar la proyección sobre los ejes principales de la posición del marcador (véase la figura 3). Estas ayudas permiten al usuario posicionar adecuadamente el puntero en el espacio. Por otro lado, el sistema también tiene un rango de tolerancias ajustable, que permite determinar si la respuesta del alumno es correcta o no.

Por último, el sistema permite implementar ejercicios que presentan una escena compuesta por una serie de primitivas y el estado al que se desea llegar. En este caso, el alumno debe desarrollar las operaciones necesarias para transformar la escena inicial al estado final. Un ejemplo de este tipo de ejercicios es cuando el alumno debe aplicar dos transformaciones, una traslación y una rotación, para llegar a la posición deseada. Este tipo de ejercicios permiten darse cuenta

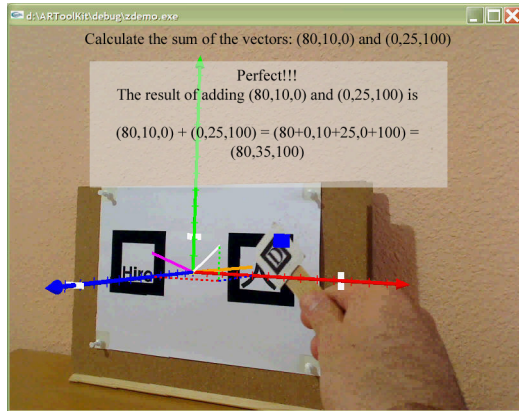


Figura 3: Ejercicio donde el alumno debe dibujar el resultado.

de la importancia del orden en el que se aplican las transformaciones (véase la figura 4).

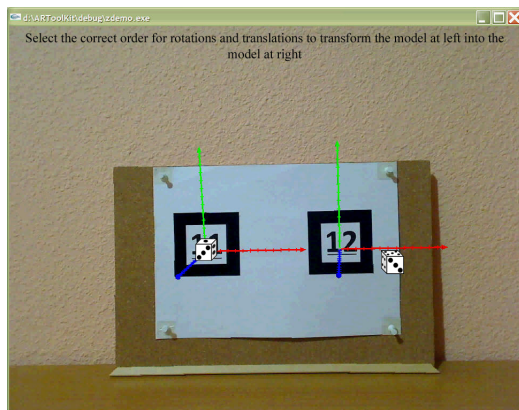


Figura 4: El alumno debe aplicar las transformaciones en el orden oportuno.

Cada transformación va asociada a un marcador. El orden de aplicación de las transformaciones al objeto está definido por el orden en el que se introducen los marcadores en la escena. En el ejemplo de la figura 4, el usuario debería mostrar primero el marcador de rotación y posteriormente el de traslación para conseguir el resultado correcto. Las transformaciones que se aplicarán a las primitivas dentro de la escena, deben estar definidas en el archivo de la escena correspondiente, en la sección de acciones permitidas.

Nuestro sistema permite así plantear tres tipos de ejercicios geométricos que manejan tanto primitivas como distintos tipos de transformaciones. Estos ejercicios sirven para que los alumnos adquieran intuición espacial en el manejo de dichas primitivas y transformaciones 3D.

3.2. Otras características del sistema

El profesor podrá elegir en el momento de crear la descripción de la escena si el resultado de aplicar una transformación sobre las primitivas se muestra con una animación. La animación mostrará cuadro a cuadro el cambio de posición de la primitiva. En caso de que el usuario cometa algún error, existe un marcador que simula la acción de un borrador, y que reinicia la escena a su estado inicial (como fue cargada desde el archivo). También se pueden eliminar primitivas de la escena. Para ello se puede usar la opción del menú correspondiente y seleccionar la primitiva a borrar en la lista desplegable (ver figura 5).

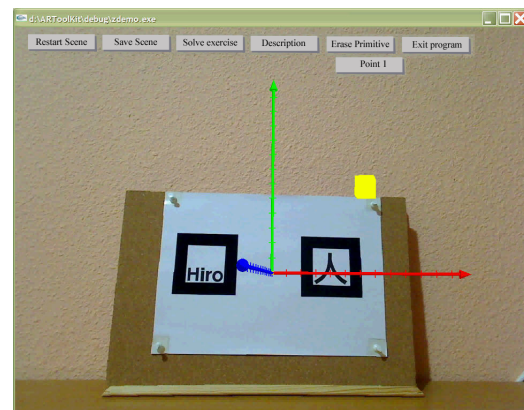


Figura 5: Para eliminar una primitiva el usuario la selecciona mediante el cursor desde la lista que aparece en el menú.

Cada ejercicio puede usar un conjunto de marcadores distinto. De esa forma, cambiar de un ejercicio a otro es tan sencillo como cambiar los marcadores de fondo. Se podría disponer de un conjunto de fondos a modo de libro que permitiera cambiar fácilmente de un ejercicio a otro.

Aparte de la interacción mediante los marcadores existe otra forma de interactuar con el sistema: el menú en pantalla. Para minimizar el uso del teclado, la aplicación permite introducir un marcador para mostrar el menú principal. En ese momento se interrumpe la interacción con la escena, y se muestra la lista de opciones principales: reiniciar la escena, guardar la escena, resolver el ejercicio, mostrar/ocultar la descripción de la escena, borrar primitivas y salir (véase la figura 5). Para los usuarios avanzados se han definido una serie de atajos de teclado para lanzar acciones sin necesidad de interactuar con el menú.

3.3. Escenarios de la aplicación

Considerando las características descritas, planteamos el uso de la aplicación en dos escenarios distintos: durante una clase, como material de apoyo a la explicación del profesor,

y en el laboratorio o en casa, donde el alumno resuelve ejercicios planteados por el profesor.

En el primer escenario, el profesor es quien dispone de los marcadores que se van a utilizar en la sesión. Cuando el profesor desee reforzar algún concepto, tan sólo deberá seleccionar los marcadores correspondientes en el fondo, y apuntar la cámara. Una vez que el sistema detecta los marcadores, el profesor puede manipular la escena según las acciones previstas en su definición. En cualquier momento es posible mover la cámara para cambiar el punto de vista y observar el modelo desde otra posición.

En el segundo escenario la principal diferencia es que las instrucciones para la realización del ejercicio están codificadas en el fichero de escena. En ambos casos, el usuario puede guardar el estado actual de la escena y continuar en otra sesión. La ventaja de usar ficheros para almacenar la descripción de las escenas es que el profesor puede definir dichos ficheros y puede distribuirlos a los alumnos mediante correo electrónico o mediante una página web. También puede crear distintos ficheros y enviarlos por separado a distintos alumnos o grupos con el propósito de evaluar sus conocimientos.

4. Detalles de implementación

Nuestro sistema únicamente necesita un ordenador de tipo PC (no muy antiguo) y una webcam. La webcam permite capturar el vídeo sobre el que se mostrará la escena, que se dibujará alineada con los marcadores utilizados como fondo (véase por ejemplo, la figura 5). Para resolver el *tracking* de la cámara se ha usado la librería ARToolKit [KB99].

Cada ejercicio se representa en el sistema como un grafo de escena que contiene las primitivas gráficas y las acciones que se pueden ejecutar sobre el grafo. Cada escena viene definida por dos marcadores que cumplen dos objetivos: identificar la escena y definir la posición y orientación del sistema de coordenadas donde está definida. En nuestras pruebas, dichos marcadores están contruidos sobre una hoja de papel pegada a un tablero de corcho. El usuario puede cambiar en cualquier momento el punto de vista desde donde se visualiza la escena moviendo la cámara con respecto a dichos marcadores.

4.1. Conjunto de marcadores

Cada acción permitida dentro de la escena está asociada a un marcador. Dichos marcadores, que son más pequeños que los usados como fondo, vienen fijados a un trozo de madera a modo de varita, para facilitar su manejo y orientación dentro de la escena (véase la figura 6).

La figura 1 muestra la configuración típica del entorno de trabajo para la utilización del sistema. En esta configuración el usuario puede colocarse cómodamente del lado de la mano con la que escriba.

Uno de los problemas principales de las aplicaciones que

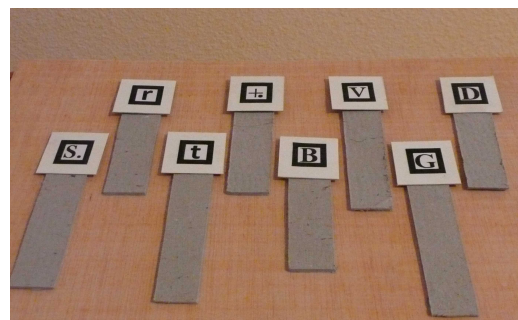


Figura 6: Distintos marcadores utilizados para ejecutar acciones.

usan ARToolKit es que el algoritmo implementado para la detección de los marcadores es muy sensible a las condiciones de iluminación. Los reflejos producidos por las luces de la habitación sobre la tinta negra de los marcadores reducen la fiabilidad del tracking. Esto provoca frustración en los usuarios, puesto que, aparte del propio ejercicio, deben preocuparse de la posición relativa entre la cámara, el marcador y las luces. Para buscar una solución a este problema, se probaron cuatro materiales distintos para construir los marcadores: tinta de impresora láser, tinta de impresora de inyección, cartulina negra y papel de terciopelo. Los mejores resultados se obtuvieron con el papel de terciopelo. Al ser un material mate, se reducen los brillos y se facilita el proceso de segmentación de ARToolKit (véase la figura 7). Para facilitar la construcción de los marcadores, la parte interior de los mismos, que sirve para identificarlos, se sigue imprimiendo con tinta en lugar de usar papel de terciopelo.



Figura 7: Resultados de la segmentación de un marcador hecho con papel terciopelo (izquierda) y tinta de impresora de inyección (derecha).

Como se ha comentado, cada escena viene identificada por dos marcadores, lo que permite reducir los problemas que se producen cuando el usuario tapa parcial o totalmente uno de los marcadores. La figura 8 muestra un ejemplo de este problema. Cuando el sistema es capaz de reconocer ambos marcadores, combina la información extraída de cada uno de ellos para obtener un resultado más estable.

4.2. Ficheros de configuración

El sistema trabaja con varios tipos de ficheros:

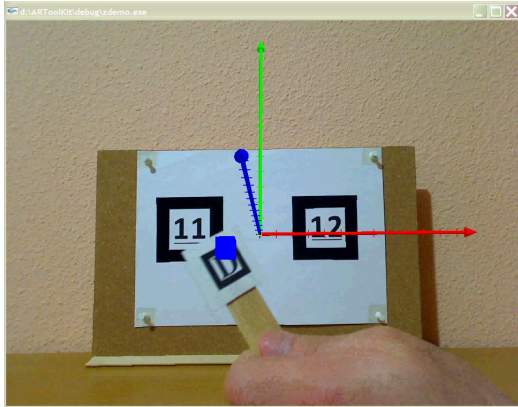


Figura 8: Usar dos marcadores permite al sistema funcionar incluso cuando uno de ellos no es visible.

- los propios marcadores,
- una lista con los marcadores disponibles en el sistema,
- la lista con las herramientas que se pueden utilizar,
- los marcadores que definen el fondo de cada ejercicio, y
- el fichero de sesión.

Para facilitar la creación, edición y lectura de estos ficheros se utiliza un formato basado en XML. Los marcadores se describen mediante el formato de ARToolKit.

El fichero que contiene la lista de marcadores es único en el sistema y contiene la ruta del fichero donde se encuentra cada marcador. De esta forma, cada marcador está identificado en el sistema por su posición dentro de este fichero, desacoplando cada ejercicio de la ruta donde se encuentran los ficheros de marcadores. Esto permite reutilizar marcadores en distintos ejercicios y sustituir, añadir o eliminar marcadores con facilidad.

La lista de herramientas disponibles relaciona acciones con marcadores. Cada ejercicio podrá especificar las acciones permitidas.

Para establecer el sistema de coordenadas de cada ejercicio, en el fichero de definición de fondos, se define la posición del origen con respecto a cada marcador. Por defecto, dicho sistema de coordenadas está situado en el centro de los dos marcadores de fondo de cada ejercicio.

Cada fichero de sesión puede contener varios ejercicios. Cada ejercicio tiene la siguiente información:

- una descripción donde el profesor explica lo que se espera que aprenda el alumno con el ejercicio,
- los identificadores de los dos marcadores usados como fondo,
- la lista de primitivas que se podrán dibujar en la escena, aunque éstas no sean visibles inicialmente,
- la lista de acciones que se pueden realizar en la escena,

- el máximo número de primitivas que podrá insertar el usuario en la escena, y
- uno o más grupos de primitivas.

Los grupos de primitivas especifican qué primitivas del ejercicio se pueden manipular. A cada grupo de primitivas se le pueden asignar una o más acciones que afectarán a las primitivas que estén dentro del grupo. Las primitivas que no están incluidas en ningún grupo no se verán afectadas por ninguna acción que modifique la escena.

5. Ejemplo

Esta sección describe un ejemplo de uso del sistema. El objetivo del ejemplo mostrado en la figura 9 es estudiar la suma vectorial. Para ello, el profesor debe preparar una escena con los elementos necesarios. A continuación se muestra un extracto de una posible implementación del ejercicio:

```
<SCENE>
...
<PRIMITIVESLIST>
...
  <VECTOR>
    <PROPERTIES>
      IDENT VECTOR1
      COLOR 1.0 0.647 0.0 1.0
      WIDTH 5
      COMPONENTS 1
      VISIBLE 0
    </PROPERTIES>
  </VECTOR>
  <VECTOR>
    <PROPERTIES>
      IDENT VECTOR2
      COLOR 1.0 0.0 1.0 1.0
      WIDTH 5
      COMPONENTS 1
      VISIBLE 0
    </PROPERTIES>
  </VECTOR>
  <VECTOR>
    <PROPERTIES>
      IDENT VECTOR3
      COLOR 1.0 1.0 1.0 1.0
      WIDTH 5
      COMPONENTS 1
      VISIBLE 0
    </PROPERTIES>
  </VECTOR>
</PRIMITIVESLIST>
<ACTIONLIST>
  DRAWVECTOR 1
  SUMVECTOR 1
</ACTIONLIST>
<GROUPS>
  <GROUP>
    <ACTIONS>
      SUMVECTOR
    </ACTIONS>
  <PRIMITIVES>
```

```

    VECTOR1
    VECTOR2
  </PRIMITIVES>
  <RESULT>
    VECTOR3
  </RESULT>
</GROUP>
</GROUPS>
</SCENE>

```

Al inicio se ve la descripción de las propiedades de cada primitiva en la escena. Se observa también la enumeración de las acciones permitidas en la escena: dibujar vectores y calcular su suma. También se puede ver cómo se agrupan las primitivas. En este caso, se especifica cuáles serán los operandos, y dónde se deberá poner el resultado.

Inicialmente, al apuntar la cámara a los marcadores de fondo, se muestra únicamente el sistema de coordenadas. Para dibujar los vectores a sumar, el profesor debe seleccionar la herramienta de dibujo adecuada e introducirla en el campo de visión de la cámara. Una vez que el sistema detecta el nuevo marcador, inmediatamente entra en modo de dibujo, añadiendo ayudas visuales sobre los ejes para posicionar más fácilmente el vector en el espacio.

Una vez que se han dibujado los operandos, el profesor deberá introducir en la escena el marcador asignado a la acción de suma vectorial. El sistema mostrará el resultado de sumar los vectores automáticamente. Una vez que se ha completado el ejemplo, el profesor podrá mover la cámara para observar la escena desde otro punto de vista.

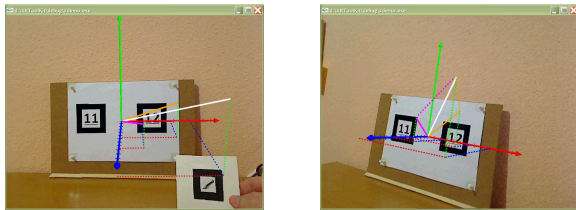


Figura 9: Después de dibujar dos vectores, uno en amarillo y otro en rosa, se aplica la acción de sumar vectores para mostrar el vector resultado en blanco; izquierda, vista frontal y derecha, vista lateral.

Una vez que se ha completado el ejemplo, el usuario puede guardarlo en un fichero para su uso en otra sesión. Para completar el proceso de aprendizaje, el profesor puede dar a los alumnos un ejercicio para resolver, evaluarlo y asignar una nota.

6. Resultados

En esta sección se presentan los resultados de los primeros tests de usabilidad del sistema. En estas pruebas participaron 12 personas con un promedio de edad de 26 años, que estudian en diferentes programas de posgrado de la Universidad

Politécnica de Valencia. De los 12 usuarios encuestados, 7 son hombres y 5 son mujeres.

El sistema se ha probado con usuarios no acostumbrados a trabajar con aplicaciones de RA. De los 12 usuarios 5 habían oído hablar de esta tecnología y 7 de ellos desconocían totalmente su existencia. Previo a la experimentación, los usuarios recibieron una plática de introducción de aproximadamente 10 minutos. En esta charla se les explicó el objetivo de las pruebas y el funcionamiento básico del sistema. Se les hizo una demostración de las funcionalidades de la herramienta y se les explicó las tareas a realizar. El experimento consistió en una serie de 5 ejercicios en los que los usuarios debían alcanzar ciertas metas. Los ejercicios incluyeron las acciones básicas dentro del sistema: cambiar entre una escena y otra, dibujar primitivas, borrar primitivas, aplicar transformaciones y seleccionar elementos simulando el clic sobre los marcadores. Después de 25 minutos en promedio trabajando con el sistema, se les pasó un cuestionario sobre su experiencia.

De acuerdo con las encuestas realizadas, a los usuarios les resultó más fácil manipular las herramientas del sistema cuando se encontraban detrás de la cámara, preferentemente del lado de la mano con la que el usuario escribe.

Respecto a las cuestiones de usabilidad del sistema, 8 de los 12 usuarios encuestados, encontraron fácil o muy fácil utilizar las herramientas del sistema para interactuar con las primitivas dibujadas en la escena. También la misma cantidad de usuarios encontró igualmente fácil la simulación del clic sobre los marcadores para confirmar la realización de una escena. Solo una de las 12 personas encuestadas demostró cierto disgusto por la manera en como el sistema relaciona cada escena con un par de marcadores. De acuerdo a esta persona, resultaría más fácil si no hubiera que cambiar de marcadores para trabajar con una escena diferente.

Otro aspecto del sistema que recibió elogios fue el módulo de autoevaluación, ya que a todos los usuarios les pareció fácil o muy fácil la manera de resolver los ejercicios presentados.

Las conclusiones preliminares que se pueden extraer de dichos cuestionarios son que a la mayoría de los usuarios que probaron el sistema les pareció una idea novedosa. Así, dichos usuarios creen que el uso del sistema facilita el aprendizaje y la mayoría usaría el sistema si tuvieran que estudiar los conceptos tratados en él. También se ha llegado a la conclusión de que hay que mejorar la robustez del sistema frente a los cambios de iluminación.

7. Conclusiones y trabajo futuro

Se ha presentado un sistema de RA para la enseñanza de geometría como un ejemplo de la aplicación de esta tecnología en el campo educativo. Hemos presentado una versión funcional del sistema, pero se han detectado áreas suscepti-

bles de mejora. Se ha probado el sistema con usuarios y se han presentado los resultados de dichos tests de usabilidad.

De los resultados se puede deducir el impacto que tiene la RA sobre los alumnos, como elemento motivador y como herramienta para mejorar la comprensión de ciertos conceptos, que es uno de los objetivos principales del desarrollo de la herramienta. En general, se obtuvo una buena aceptación del sistema, incluso en aquellos usuarios que nunca habían oído hablar de Realidad Aumentada.

Con respecto a las mejoras que se pretenden introducir a corto plazo, cabe mencionar la introducción del módulo de reconocimiento de comandos por voz para la simplificación de la interacción del usuario con el sistema. Este nuevo módulo contemplaría comandos simples para algunas tareas como: guardar una escena, borrar primitivas o resolver un ejercicio por citar algunos ejemplos.

Para dotar al sistema de mayor flexibilidad y facilitar al usuario la generación de escenas se piensa desarrollar una herramienta de autor que facilite al profesor la creación de las escenas. Por otro lado, también se hace necesaria la evaluación más profunda del sistema involucrando más alumnos y profesores.

Se ha planeado usar la aplicación como herramienta didáctica en las asignaturas básicas de introducción a los gráficos por computador de la Escuela Técnica Superior de Ingeniería Informática de la Universidad Politécnica de Valencia. Se evaluarán aspectos de usabilidad y diseño de la aplicación para validar si el sistema logra su propósito pedagógico. Se evaluará si la herramienta ayuda a los estudiantes a comprender mejor conceptos como las transformaciones, al mismo tiempo que se convierte en una herramienta accesible y fácil de usar para el profesor.

Agradecimientos

Esta investigación ha sido parcialmente patrocinada por el Consejo Nacional de Ciencia y Tecnología de México como parte del Programa Especial de Ciencia y Tecnología.

Quisiéramos agradecer también a las personas que han participado desinteresadamente en las pruebas al sistema.

Bibliografía

- [Azu97] AZUMA R. T.: A survey of augmented reality. *Presence: Teleoperators and Virtual Environments* 6, 4 (1997), 355–385. 2
- [BKP01] BILLINGHURST M., KATO H., POUPYREV I.: The magicbook - moving seamlessly between reality and virtuality. *Computer Graphics and Applications, IEEE* 21, 3 (May/June 2001), 6–8. 2
- [BPI07] BALOG A., PRIBEANU C., IORDACHE D.: Augmented reality in schools: Preliminary evaluation results from a summer school. In *Proceedings of WASET International Conference on Technology and Education - ICTE 2007* (Oct 2007), pp. 114–117. 3

- [BWG06] BOGEN M., WIND J., GIULIANO A.: ARiSE - augmented reality in school environments. In *First European Conference on Technology Enhanced Learning, EC-TEL 2006* (Crete, Greece, 2006), pp. 709–714. 3
- [CH93] CARLSSON C., HAGSAND O.: DIVE a multi-user virtual reality system. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE* (Sep 1993), pp. 394–400. 4
- [FFE*07] FIELD M., FREDRIKSSON J., EJDESTIG M., DUCA F., BÝTSCHI K., VOEGTLI B. M., JUCHLI P.: Tangible user interface for chemistry education: comparative evaluation and redesign. In *Special Interest Group on Computer-Human Interaction (SIGCHI)* (2007), pp. 805–808. 4
- [JB08] JUAN M., BEATRICE F. C. J.: An augmented reality system for learning the interior of the human body. In *The 8th IEEE International Conference on Advanced Learning Technologies (ICALTŠ08)* (2008), pp. 186–188. 4
- [JFM*08] JUAN M., FURIÓ D., MOLLÁ R., VICENT M. J., VÍVÓ R., GIMÉNEZ M.: Edutainment games included as activities in the summer school of the technical university of valencia. In *Game-On* (Valencia, España, 2008). 4
- [KB99] KATO H., BILLINGHURST M.: Marker tracking and hmd calibration for a video-based augmented reality, conferencing system. In *2nd IEEE and ACM International Workshop on Augmented Reality (IWAR 99)* (1999). <http://www.hitl.washington.edu/artoolkit>. 7
- [Kon06] KONDO T.: Augmented learning environment using mixed reality technology. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006* (2006), pp. 83–87. 3
- [LMW*04] LIAROKAPIS F., MOURKOSSIS N., WHITE M., DARCY J., SIFNIOTIS M., PETRIDIS P., BASU A., LISTER P. F.: Web3d and augmented reality to support engineering education. *World Transactions on Engineering and Technology Education* 3, 1 (2004), 11–14. 3
- [LPLW02] LIAROKAPIS F., PETRIDIS P., LISTER P. F., WHITE M.: Multimedia augmented reality interface for e-learning (MARIE). *World Transactions on Engineering and Technology Education, UICEE 1* (2002), 173–176. 2
- [NQN*08] NUÑEZ M., QUIRÓS R., NUÑEZ I., CARDA J. B., EMILIO C.: Collaborative augmented reality for inorganic chemistry education. In *New aspects of engineering education, Proceedings of the 5th WSEAS* (Jul 2008), pp. 271–277. 4
- [SFSG96] SCHMALSTIEG D., FUHRMANN A., SZALAVÁRI Z., GERVAUTZ M.: Studierstube - an environment for collaboration in augmented reality. In *Proceedings of Collaborative Virtual Environments '96* (1996), pp. 19–20. 3
- [SG97] SZALAVÁRI Z., GERVAUTZ M.: The personal interaction panel - a two-handed interface for augmented reality. In *Computer Graphics Forum* (1997), vol. 16, pp. 335–346. 3
- [SH02] SHELTON B., HEDLEY N.: Using augmented reality for teaching earth-sun relationships to undergraduate geography students. In *First IEEE International Augmented Reality Toolkit Workshop* (Darmstadt, Germany, 2002). 2
- [SWK*00] SCHMALSTIEG D., WAGNER M., KAUFMANN H., KAUFMANN H., GASSE L.: Construct3d: a virtual reality application for mathematics and geometry education. *Education and Information Technologies* 5 (2000), 263–276. 3
- [TN01] TAXÉN G., NAEVE A.: Cybermath: Exploring open issues in vr-based learning. *SIGGRAPH 2001 Conference Abstracts and Applications* (2001), 49–51. 3

A Low Cost Augmented Reality System to Help Students Learn Geometric Concepts

Abstract

This paper presents a new Augmented Reality system aimed at teaching abstract concepts in Geometry education. The system shows interactive 3D scenes that can be explored and manipulated by the students using physical markers. It also allows them to solve exercises prepared by the professor.

The system can be used in different scenarios without special hardware: in the classroom as an aid to the professor, or at home as a self-study tool. New exercises can be created using an interactive tool that generates scene descriptions using an XML-based language. Later, the system provides real-time feedback to the student, automatically correcting her answers.

Our paper shows how to appropriately configure the students' working environment. It also presents different exercises and case studies. Finally, we present the results obtained from a survey completed by a group of students that tested the system.

Keywords: Spatial Intuition, Augmented Reality, Mathematics Education, Geometry

Categories and Subject Descriptors (according to ACM CCS):

I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality

I.3.8 [Computer Graphics]: Applications

1. Introduction

Mathematics education has always been a challenge for teachers and professors. Explaining abstract concepts to the students using words and diagrams is not an easy task. Besides the difficulty of the concepts, the teacher has to take into account each student's capabilities. Not all the students have the same learning ability to comprehend the explanations of every subject.

Traditionally Mathematics teaching only uses 2D models drawn on the chalkboard. Vectors defined in 3D space, or solid volumes are not clearly drawn on 2D surfaces. For this reason, the material is hard to understand by a student. She may also be asked to build models of the 3D concepts studied. This simplifies understanding the concepts, but requires investing time that could be used for other concepts. For these reasons we need solutions to help instructors explain difficult concepts in a better way. We also want a tool for the students to interact with 3D models so they can acquire the necessary spatial intuition.

New technologies are being introduced into the classroom

to improve learning. Slide projectors let the professor show material that reinforces her explanations. Digital projectors connected to computers allow specialized software to show animations and interactive applications. This kind of technology is changing the way classes are taught. However they are not enough to comprehend difficult concepts. Augmented Reality (AR) combines computer vision, pattern recognition and image synthesis technology. Its main objective is to enhance the user's perception of reality by superimposing information. This information is typically coded as computer generated images that enrich user's experience.

According to Azuma an AR system must provide the following features [Azu97]: (i) mix reality with virtual stimuli, (ii) real-time interaction with the system, and (iii) 3D registration of synthetic images. AR provides some of the benefits of virtual environments, at a significantly lower cost than a VR environment. AR also allows better mobility. We can find a variety of AR systems, going from storytelling systems [BKP01] to complex distributed systems for e-learning [LPLW02]. This kind of systems represents a novel and attractive tool. They try to make the teaching

process easier, and the learning process more appealing. A good example is Shelton and Hedley's system to teach the relationship between the Earth and the Sun [SH02].

This paper proposes the use of AR to simplify the visualization and manipulation of geometric models in 3D space. We propose a tool that lets the student interact with virtual models using real objects (markers). We think that our system improves the students' spatial intuition by allowing them to watch and interact with geometric 3D models. Moreover, our system runs using a webcam and a PC. It does not require specialized or expensive hardware. This enables using it both in the classroom and at home.

Our system supports creating multiple types of exercises like selecting and placing an object in a given 3D position, or adding two vectors. The user can also create, animate, erase and manipulate primitives, and perform operations like affine transformations. The system allows the instructor to create models, exercises and solutions to them. It provides a visual tool that does not require the instructor to learn a 3D description language. In this paper we describe our tool and present preliminary usability results obtained with a group of university students. The survey shows that this is an effective approach for teaching 3D Geometry.

The rest of the paper is organized as follows. The next section presents the state of the art of AR systems designed for educational purposes. The following two sections list the features of our system and show its structure. An example of one of the exercises that can be implemented in the system is presented in the following section. Then we discuss the results of preliminary tests with users. Finally we present conclusions and future improvements on our system.

2. Previous Work

In this Section we review the most relevant educational applications of AR. Applications can be classified according to area of interest. They can also be classified into single or multiuser systems. The users may work alone locally or they may collaborate remotely. When evaluating a system we look for usability and pedagogical quality. By usability we mean ease of use and ease of learning the system. By pedagogical quality we understand how much the system improves the learning process. Finally, we distinguish the educational paradigms followed by each system. They are traditional lectures, collaborative systems, self-study systems and learning by playing.

2.1. Traditional Lectures

We describe two systems based on the traditional lecture paradigm. The MARIE (*Multimedia Augmented Reality for E-Learning*) system was designed to be an engineering e-learning tool [LPLW02]. The system lets the user follow a script given by the professor. It allows interaction with 2D

and 3D models during the professor's explanations. It can be used with a Head Mounted Display (HMD) or with a standard PC monitor supporting one or more students at the same time. The system has a sound module that can mix different sources, like the professor's voice or a prerecorded explanation. The problem is that it requires the student to use the keyboard, the mouse and 2D markers to interact with it. This can be a problem for inexperienced users.

The work presented in [Kon06] describes three AR systems designed for education. The first one is an enhanced textbook, where 3D images are overlaid on a traditional textbook to improve conceptual explanations. It uses a webcam to locate a 2D pattern printed on the textbook pages. The second system is for Math education inside a classroom. With it, the professor explains 3D models to the students using a projection screen. With a webcam and markers the professor can interact from her desk with the virtual models of solid volumes. The last system augments dinosaurs' skeletons with graphics information about the shapes of their bodies. It allows the user to experiment with dinosaur aspect that are still under study like, for example, their skin color. A design tool is included to help professors prepare 3D content for the three systems.

2.2. Collaborative Systems

Construct3D [SWK*00] is an AR system based on the Studierstube [SFSG96] collaborative platform. Using HMDs, multiple users can work with it on the same task. The system is not a professional design tool, but a friendly GUI that uses the Personal Interaction Panel (PIP) [SG97]. This panel enables users to interact with virtual objects using a tangible interface and a menu of actions and options. The primitives supported by the system are points, lines, planes and 3D objects, like spheres and cones. The user can save scenes into a file. It has an audio subsystem for improved user interaction. According to the survey done to evaluate the system, some users prefer regular CAD systems instead of Construct3D. However the same users found Construct3D easy to use and useful as a learning tool.

ARiSE (Augmented Reality in School Environments) is a platform built by several European Universities [BWG06]. It can be simultaneously used by different users. It has an authoring tool to easily create 3D content. The authors did usability tests to evaluate the system's ease of use and pedagogical quality [BPI07]. They found that the novelty of the technology stimulated the students to use it. The usability tests, however, showed that users expected more feedback. They wanted the system to better report whether the user actions met their goals. Some precision problems while using the interaction tool were also detected. Note that none of the users had previous experience using an AR system.

2.3. Self-Study Systems

CyberMath [TN01] is a system inspired by a museum that supports different educational paradigms. It is made of rooms where the student can study different subjects. The system is a virtual environment based on DIVE [CH93] that can be used in a CAVE or with an HMD. It is distributed allowing multiple users to work simultaneously on the Internet. The classrooms of the system have avatars of the user as well as realistic graphics that make the application very appealing. The system allows importing Mathematica models. Some usability tests were done. They showed that in some cases avatars blocked the users' visibility.

Another AR application for self-study is the one presented by Nuñez et al. [NQN*08]. In their system 3D VRML models of inorganic chemical compounds are shown to the students. The system has six USB cameras connected to a standard PC via a hub. This architecture enables a small group of users to work with the system. The video sequence obtained by each camera is shown on an independent window. Therefore every user can see everybody else's work on a projection screen. Preliminary tests done by the authors showed that the students enjoyed the system and their motivation was improved.

2.4. Learning by Playing

Juan et al. present three AR games related to environmental issues [JFM*08]. 160 children participated in the game tests. In the first game, *Magnetic Cubes*, the user selects endangered species from a set of animals. If the user chooses correctly, a video about the species is displayed. The interface is made of three cubes magnetically attached to each other. Combining the markers of the visible faces, the user performs the tasks needed to interact with the system. *Finding pairs* is the second game, a classic memory game where the user finds the card that completes a pair with an endangered animal. If the user succeeds, she can decide to watch a video about that species. The third game was developed for mobile phones and deals with recycling. The user has to identify an image shown over a 2D marker and put it in the correct recycling bin. There are five different markers, one for each recycling bin and one for the object to recycle.

All of the above systems are very complete. However, many are expensive to implement at a school level. Another problem is that the student may just be as a spectator in the learning process. Our application aims at solving these problems. We give the student a leading role in the learning process. We also give the professor a versatile tool to create more attractive, more effective lessons. Finally, our system has an interactive way of evaluating the students. Our design is simple and inexpensive, and can thus be implemented with a reduced budget.

3. System Description

Figure 1 shows our system's configuration: a PC, a webcam and several markers that can be built by the students. The system supports primitives, like 3D coordinate axes, points, vectors, spheres, cones, cubes, cylinders and 3D VRML models stored in VRML files. Primitives can be grouped and actions can be applied to both single primitives and groups. Examples of actions supported by our system are: placing an object in the scene, applying transformations to primitives (translations, rotations and scales), choosing an option in a multiple choice question, resetting the scene by removing all of its primitives, and saving the current state of the scene, among others. Most of these actions are associated to markers. They are applied by showing the right marker to the camera.

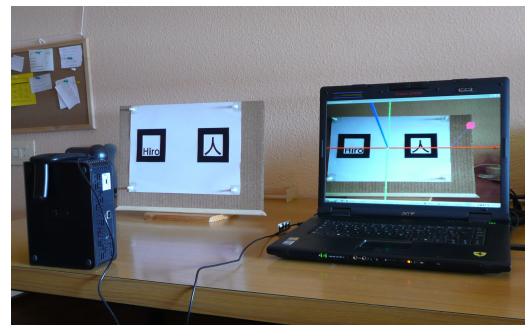


Figure 1: Recommended configuration for our system's workspace.

3.1. Types of Exercises

Given the above sets of primitives and actions, our system allows the instructor to prepare exercises for the students. Exercises may be evaluated in real time. We distinguish three types of supported exercises: multiple-choice exercises, drawing exercises and transformation exercises.

In a multiple choice exercise, the instructor asks a question about the scene and the student chooses one of several answers. The system checks if the answer is correct and, if not, describes the correct answer (see Figure 2). To make a choice, the student must cover the desired marker, partially or completely, during a given period of time. She can use her fingers to do so, as shown in Figure 2. When a marker is covered, the system places an arrow above it to provide feedback about the option selected.

In drawing exercises the students draw primitives with their positions and orientations. For example, to add two vectors, the student uses a drawing tool to draw the resulting vector in 3D space. A problem with this exercise is the limited drawing accuracy of the hand. To help with it we display the projections of the 3D cursor onto the main axes, and we

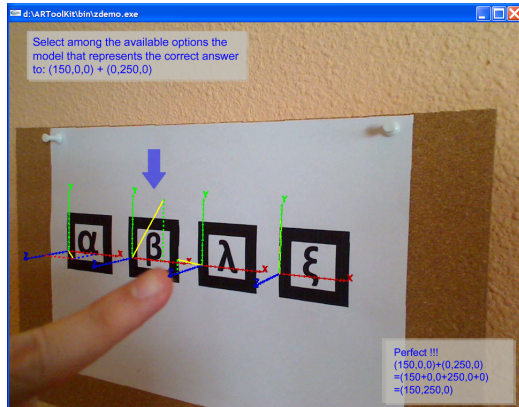


Figure 2: Multiple choice exercise; the axes have been drawn in red (x-axis), green (y-axis), and blue (z-axis) using a right-handed coordinate system. The possible answers are drawn in yellow. A light-blue arrow points to the current choice.

display the numeric 3D coordinates of the cursor in the lower right corner of the screen (see Figure 3). This helps the student to precisely locate the cursor in 3D space. Additionally, the system has a tolerance to determine if the student has given the correct answer.

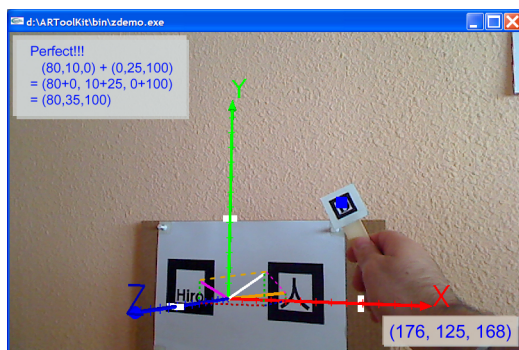


Figure 3: Drawing exercise: the student has to draw the resulting vector.

Transformation exercises teach affine transformations. For example, an object may be shown in two possible configurations. The student has to give a sequence of transformations to take the object from the first to the second configuration. A simple version is when only a rotation and a translation are needed. Each transformation is associated to a different marker and the student has to select them in the proper order. The order is given by the order in which the markers appear in the scene (see Figure 4).

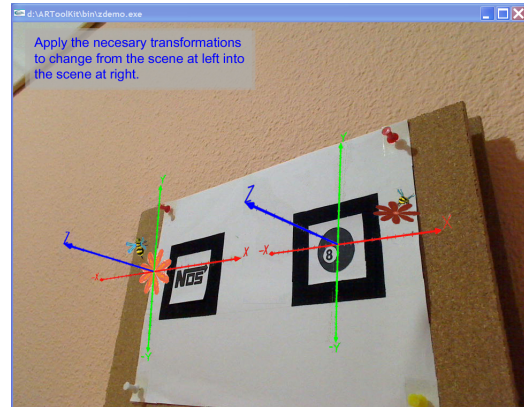


Figure 4: The student has to apply the transformations to take the flower to the desire position and orientation; they have to be applied in the right order.

3.2. Other System Features

We have implemented animated transformations. The system can display an object being transformed from its initial to its final state. This playback feature helps the student understand how the transformations work.

Primitives may also be removed from the scene. When a student makes a mistake she can either delete a primitive or reset the scene to its original state, when it was loaded from a scene description file. To delete a primitive, she can choose from a list shown after selecting a menu option (see Figure 5). To reset the scene, the student only needs to show the appropriate marker.

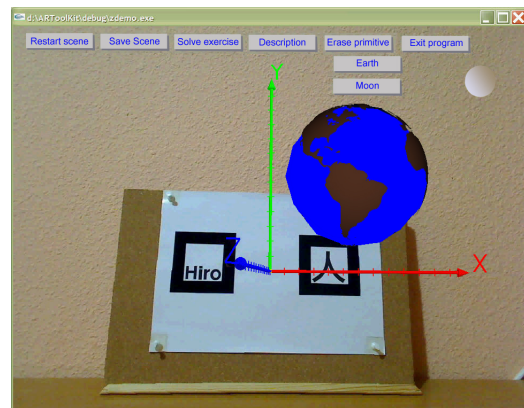


Figure 5: The user can select the primitive to be removed from the list shown in the menu.

Exercises are defined by background markers. To change exercises it suffices to show a different set of markers to the camera. This allows each student to have her own book of markers. Interaction with our system is done using markers

and a menu. The menu is enabled by a specific marker. No keyboard use is needed. When the menu is on screen, the user can not interact with the scene. The options of the system menu are: reset the scene, save the work, solve an exercise (showing the answer), show/hide a description of the exercise, delete primitives, and exit (see Figure 5). For advanced users there are keyboard shortcuts that allow doing these actions.

Our system also includes a light source that improves the perception of the 3D objects in the scene. Its properties can be defined for each scene. Currently, we only allow adjusting its position and color. The position can be fixed in the scene or can follow the position of camera, pointing at the markers.

3.3. Scene Description Tool

Our system uses scene description files to store the scenes used by students and professors. Scenes are written in a language similar to XML. To easily create them a GUI-based scene construction tool is provided (see Figure 6). The tool allows describing the scene and the markers, and adding groups of primitives and lights. Properties of this elements may also be included, like the color of a light. For transformation exercises the professor must include the initial and the final scene configurations. Valid actions are also included in this file and associated to the appropriate scene elements. Finally, a file may contain multiple scenes.

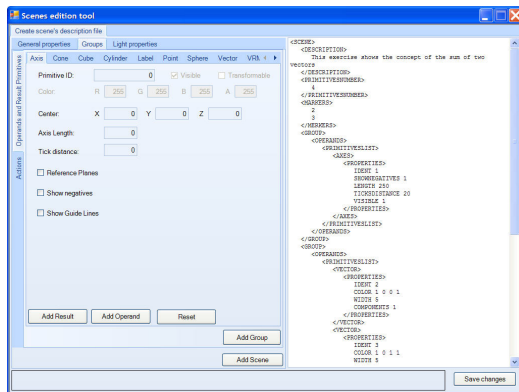


Figure 6: The GUI of our scene description tool.

3.4. Application Scenarios

Given the previous features, we suggest using the application in two different scenarios: (i) during a class, to support the professors' explanations, and (ii) in the lab or at home, where the student solves the exercises prepared by the professor.

In the first scenario, the professor has all the markers for the exercises. When the professor needs to stress a concept, she just sets the markers in the background in front of the

camera. When the system detects the markers the professor manipulates the scene according to the actions defined for those markers. The viewpoint may be changed at any time to see the 3D models from different viewpoints.

In the second scenario the instructions for the exercises are coded in the scene description file. From them the student can reproduce the professor's explanations either at home or in the lab. In both scenarios the student can save the current state of the scene and continue during a later session. The advantage of using description files is that the professor can distribute them via e-mail or a web page. Furthermore, the professor can create different exercises targeted at different groups of students. Therefore students can work individually or in groups to test their understanding of primitives, transformations and spatial relationships.

4. Implementation Details

Our system uses a PC and a webcam. The webcam captures video and the PC overlays a virtual scene on the video. The scene is drawn aligned to the background markers (see, for example, Figure 5) using ARToolKit [KB99]. The exercises are represented by a scene graph composed of primitives and actions that can be applied to primitives. Each scene is identified by two background markers that also define the coordinate axes. The user may change the viewpoint at any time moving the camera around the markers.

4.1. Marker Set

Each action is associated to a marker. Action markers are smaller than background markers, and are glued to a wooden stick so they can be easily handled (see Figure 7).

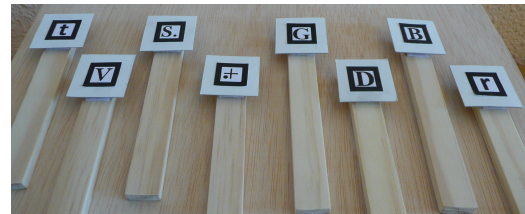


Figure 7: Different markers used to perform actions.

Figure 1 shows the recommended working environment of the system. The system can be configured for a left-handed or a right-handed user, depending on the relative position of the background markers and the screen.

A problem in ARToolKit-based systems is their sensitivity to light conditions. Highlights on the markers affect the reliability of the tracking algorithm. To improve reliability we tried four different materials to build the markers: laser printer ink, inkjet printer ink, black carton and velvet paper. The best results were obtained using velvet paper since it reduces reflections because of its matte nature (see Figure 8).

Still, to simplify construction we printed the interior of all the markers with printer ink.

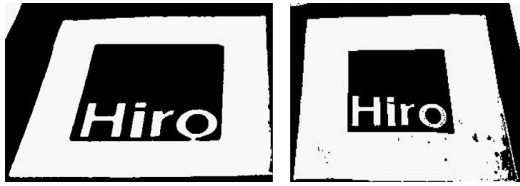


Figure 8: Segmentation results using a velvet paper marker (left) and an inkjet printed paper marker (right).

Using two markers to identify each scene avoids the problem of the user covering one marker. Figure 3 shows an example of this problem. Using both markers, the system can obtain a more accurate result. If both markers are covered, the system keeps the 3D models still on screen for a period of time. Here we assume that the user does not change the viewpoint while changing the elements in the scene. The system also implements a *hold* state that will keep the 3D models on screen no matter what happens. This is useful when the instructor wants to highlight some concept while keeping the scene still.

4.2. Configuration Files

The system maintains the following configuration information:

- a list of known marker files,
- a list with the system markers,
- a list with the tools used in the system,
- the background markers for each scene, and
- a session file.

Configuration files use a format similar to XML. This simplifies creation, editing and reading. The marker files are written using ARToolKit's format. Specifically, we store each marker as its file path, identified by its position in the marker file. This allows reusing markers in different exercises, and adding, replacing, and deleting markers in an easy way.

A list of tools connects actions to markers. Then, each exercise specifies the allowed actions. To set the coordinate system of each exercise we use the background markers. By default, the coordinate system is located at the center of the two markers.

Session files may contain several exercises. Each exercise has the following information:

- a description given by the professor and explaining what the student is expected to learn,
- the identifiers of the background markers,
- the primitives that may be drawn in the scene, even if they are not visible initially,

- the list of allowed actions,
- the maximum number of primitives allowed in the scene, and
- one or more groups of primitives.

The groups of primitives indicate which primitives can be manipulated. One or more actions can be assigned to a group. They will only affect the primitives in that group.

5. Example

In this section we describe a simple exercise implemented in our system. Figure 9 shows an image of an exercise where the student has to add two vectors. The instructor has to prepare the scene with the primitives. The exercise is implemented as follows:

```
<SCENE>
...
<GROUP>
  <OPERANDS>
    <PRIMITIVESLIST>
      <AXES>
        ...
      </AXES>
      <VECTOR>
        <PROPERTIES>
          IDENT 2
          POINT1 0 0 0 1
          POINT2 0 0 0 1
          COLOR 1.0 0.647 0.0 1.0
          WIDTH 5
          COMPONENTS 1
          VISIBLE 0
          TRANSFORMABLE 0
        </PROPERTIES>
      </VECTOR>
      <VECTOR>
        <PROPERTIES>
          IDENT 3
          POINT1 0 0 0 1
          POINT2 0 0 0 1
          COLOR 1.0 0.0 1.0 1.0
          WIDTH 5
          COMPONENTS 1
          VISIBLE 0
          TRANSFORMABLE 0
        </PROPERTIES>
      </VECTOR>
    </PRIMITIVESLIST>
  </OPERANDS>
  <RESULT>
    <PRIMITIVESLIST>
      <VECTOR>
        <PROPERTIES>
          IDENT 4
          POINT1 0 0 0 1
          POINT2 0 0 0 1
          COLOR 1.0 1.0 1.0 1.0
          WIDTH 5
          COMPONENTS 1
```

```

        VISIBLE 0
        TRANSFORMABLE 0
    </PROPERTIES>
</VECTOR>
</PRIMITIVESLIST>
</RESULT>
<ACTIONLIST>
    DRAWVECTOR 1
    SUMVECTOR 1
</ACTIONLIST>
</GROUP>
<LIGHT>
    FOLLOWCAM 1
    COLOR 1.0 1.0 1.0
</LIGHT>
</SCENE>

```

Here there is a group of primitives, each with its own properties. The actions allowed for the group are draw vectors and calculate their sum. Two vectors are the operands of the exercise. Another vector is the container of the solution. The light source's features appear at the end of the scene: a white light that follows the camera pointing at the coordinate center.

Initially the user aims the camera at the background markers, and only the coordinate axes appear on the screen. To draw the vectors to be added, the professor selects the drawing tool and puts it front of the camera. When the system detects it, the drawing mode is activated and visual aids are added to the axes to simplify drawing the vectors to be added. Once the vectors have been drawn, the instructor puts the vector sum marker in the scene. The system then shows the result automatically. When the instructor completes the exercise, the scene can be explored from different viewpoints by moving the camera.

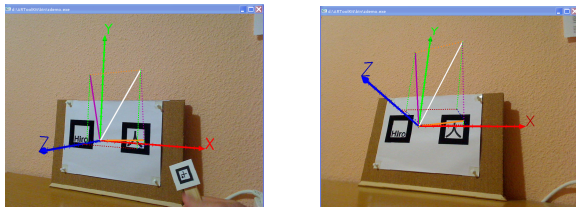


Figure 9: After drawing the yellow and pink vectors, the action “sum vectors” is performed to show the result, in white; left, front view and right, side view.

The work can be saved in a file for future use. To complete the learning process the instructor can give his students an exercise to solve, then he can grade the results.

6. Results

In this Section we present the preliminary results of the usability tests of the system. Twelve graduate students participated in the tests with an average age of 26 years old.

Five of them are women and seven are men. None of the students ever worked with AR before, but five of them had heard about it. The usability tests were made using five exercises where the users had to reach some goals. The basic actions they had to perform were: switch between scenes, draw primitives, erase primitives, apply transformations, and select primitives simulating a click on the markers. After 25 minutes of average work with the system, the users were asked to fill out a survey about their experience.

	Usability	Educational goal
Draw Primitives	3.76	4.23
Change between scenes	4.54	
Change view point	3.91	
Solve exercises	4	

Figure 10: Results of the survey: average marks given by the testers (1: lowest, 5: highest)

The users found it easy to manipulate the tools of the system when they were behind the camera. They were asked to rate some usability and pedagogical questions on a scale of 1 to 5, being 1 the lowest score and 5 the highest. Figure 10 shows the results. Most of the scores were in the 4-to-5 range. However, this can be improved. The students complained that they had to change background markers in order to change exercises. We are working on a solution to this problem. Otherwise, the self-test module received good ratings, because the users found it very useful and easy to use.

Regarding the educational aspect of the system we also got a good grade, 4,23 on average. The users said that the system helped them learn geometric concepts. They also expressed interest in continuing using the system.

Our conclusions from the survey are that most of the users agree on the usefulness of the system and that it simplifies the learning process. They say they would use it if they had to study the concepts taught by the system. They also encouraged us to apply the technology to other areas.

7. Conclusions and Future Work

We have presented an educational Augmented Reality system applied to teach geometric concepts. Our system allows a professor to set up exercises that she can solve in front of the class or students can solve in the lab or at home. We have tested the system with a group of students, and we have obtained good results in our usability and pedagogical tests. In fact the system made a good impression on the students that found it both motivating and helpful to learn Geometry and other possible disciplines.

Our system can benefit from several improvements. For example, we need to improve its robustness in the presence of changing lighting conditions. We also want to add a voice

recognition module to simplify interaction. To add flexibility to the scene descriptions we want to port the description tool to Linux. We are also aware of the need of further evaluation involving more students and professors. To do so, we want to use our system as a teaching tool in an introductory Computer Graphics class at our University. We plan on testing both usability and the design of our application. Our goals are to simplify the task of the professor and to confirm that the system is effective at teaching geometric concepts to our students.

Acknowledgments

We would like to thank those persons that participated in the survey of our system.

References

- [Azu97] AZUMA R. T.: A survey of augmented reality. *Presence: Teleoperators and Virtual Environments* 6, 4 (1997), 355–385. [1](#)
- [BKP01] BILLINGHURST M., KATO H., POUPYREV I.: The magicbook - moving seamlessly between reality and virtuality. *Computer Graphics and Applications, IEEE* 21, 3 (May/June 2001), 6–8. [1](#)
- [BPI07] BALOG A., PRIBEANU C., IORDACHE D.: Augmented reality in schools: Preliminary evaluation results from a summer school. In *Proceedings of WASET International Conference on Technology and Education - ICTE 2007* (Oct 2007), pp. 114–117. [2](#)
- [BWG06] BOGEN M., WIND J., GIULIANO A.: ARiSE - augmented reality in school environments. In *First European Conference on Technology Enhanced Learning, EC-TEL 2006* (Crete, Greece, 2006), pp. 709–714. [2](#)
- [CH93] CARLSSON C., HAGSAND O.: DIVE a multi-user virtual reality system. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE* (Sep 1993), pp. 394–400. [3](#)
- [JFM*08] JUAN M., FURIÓ D., MOLLÁ R., VICENT M. J., VIVÓ R., GIMÉNEZ M.: Edutainment games included as activities in the summer school of the technical university of valencia. In *Game-On* (Valencia, España, 2008). [3](#)
- [KB99] KATO H., BILLINGHURST M.: Marker tracking and hmd calibration for a video-based augmented reality, conferencing system. In *2nd IEEE and ACM International Workshop on Augmented Reality (IWAR 99)* (1999). <http://www.hitl.washington.edu/artoolkit>. [5](#)
- [Kon06] KONDO T.: Augmented learning environment using mixed reality technology. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006* (2006), pp. 83–87. [2](#)
- [LPLW02] LIAROKAPIS F., PETRIDIS P., LISTER P. F., WHITE M.: Multimedia augmented reality interface for e-learning (MARIE). *World Transactions on Engineering and Technology Education, UICEE* 1 (2002), 173–176. [1](#), [2](#)
- [NQN*08] NUÑEZ M., QUIRÓS R., NUÑEZ I., CARDA J. B., EMILIO C.: Collaborative augmented reality for inorganic chemistry education. In *New aspects of engineering education, Proceedings of the 5th WSEAS* (Jul 2008), pp. 271–277. [3](#)
- [SFG96] SCHMALSTIEG D., FUHRMANN A., SZALAVÁRI Z., GERVAUTZ M.: Studierstube - an environment for collaboration in augmented reality. In *Proceedings of Collaborative Virtual Environments '96* (1996), pp. 19–20. [2](#)
- [SG97] SZALAVÁRI Z., GERVAUTZ M.: The personal interaction panel - a two-handed interface for augmented reality. In *Computer Graphics Forum* (1997), vol. 16, pp. 335–346. [2](#)
- [SH02] SHELTON B., HEDLEY N.: Using augmented reality for teaching earth-sun relationships to undergraduate geography students. In *First IEEE International Augmented Reality Toolkit Workshop* (Darmstadt, Germany, 2002). [2](#)
- [SWK*00] SCHMALSTIEG D., WAGNER M., KAUFMANN H., KAUFMANN H., GASSE L.: Construct3d: a virtual reality application for mathematics and geometry education. *Education and Information Technologies* 5 (2000), 263–276. [2](#)
- [TN01] TAXÉN G., NAEVE A.: Cybermath: Exploring open issues in vr-based learning. *SIGGRAPH 2001 Conference Abstracts and Applications* (2001), 49–51. [3](#)