

Titulo del proyecto



Autor del proyecto

Índice general

1. Introducción	9
2. Estado actual del tema	11
3. Objetivos	13
3.1. Objetivos académicos	13
3.2. Objetivos generales del proyecto	13
4. Competencias Específicas Cubiertas	15
5. Aportaciones del TFG	17
6. Recursos utilizados	19
6.1. Hardware	19
6.2. Software	19
6.3. Librerías	20
6.4. Otras herramientas utilizadas	20
6.5. Requerimientos del software	22
7. Plan de trabajo y temporización	23
8. Etapa1. Algoritmo de Harris	25
8.1. Detectores de esquinas	25
9. Seguimiento de puntos	27
9.1. Proceso del seguimiento	27
9.1.1. Conceptos	27
10.Resultados	29
11.Conclusiones	31
12.Trabajo Futuro	33

A. Detalles de implementación	35
A.1. Uso de las STL (Standard Template Library)	35

Índice de figuras

Índice de cuadros

Agradecimientos

Deseo dedicar este proyecto ...

Y, en general, agradecer a todas las personas que han intervenido directa o indirectamente en el proyecto.

Capítulo 1

Introducción

...

El gran avance en los últimos años de la tecnología digital de imágenes, la aparición del vídeo digital, el enorme aumento en las prestaciones de los ordenadores y el desarrollo de la informática gráfica que permite la generación de objetos y escenarios sintéticos, ha permitido el desarrollo de una nueva tecnología que permite la generación de vídeos digitales combinando imágenes reales y sintéticas. Las aplicaciones de este tipo de tecnología son muy variadas, podemos destacar los efectos especiales en cine, publicidad, la generación de estudios sintéticos para televisión, simulación del impacto visual de un edificio en un entorno real, etc.. Es en este contexto donde se enmarca el proyecto fin de carrera presentado en esta memoria.

...

Siguiendo esta línea general de trabajo que pretendemos abordar, hemos estructurado el contenido de este documento de la siguiente forma:

Comenzaremos analizando el entorno del proyecto. En los primeros cuatro capítulos hablaremos, en primer lugar del *estado actual del tema* haciendo referencia a trabajos anteriores en los que nos hemos basados en nuestro proyecto y herramientas relacionadas con algunas etapas del proyecto. En el capítulo de *objetivos* se hace una descripción general de los retos que se pretender alcanzar en cada una de las etapas en las que está dividido el proyecto. En el capítulo de *recursos* se comenta los principales recursos *hardware* y *software* empleados. En el software se hará una distinción entre las aplicaciones finales que hemos utilizado, y las librerías que nos han ayudado a desarrollar los componentes software de nuestro proyecto. En la *planificación* se hablará sobre la estimación inicial prevista y el tiempo real invertido. Se comentarán algunos aspectos que han influido notablemente en esta diferencia.

A continuación, se abordará en detalle la descripción de cada una de las etapas en las que se ha dividido el proyecto. Esta descripción se ha dividido en cinco capítulos, siguiendo un desglose por etapas, y supone el eje central de la memoria.

En la *primera etapa* se hará una breve introducción a los distintos tipos de detectores de puntos singulares que existen, destacando sus principales características. Describiremos la teoría matemática que subyace al detector de Harris. Y por último, se mostrarán y analizarán los resultados experimentales que se han obtenido con este detector. En la *segunda etapa* se estudia el funcionamiento del proceso de seguimiento de puntos singulares a través de una secuencia de vídeo; estructurando el capítulo en tres partes: definición de conceptos, criterios de sucesión y la descripción del proceso de seguimiento. Al final del capítulo se comentarán los resultados experimentales y diversas cuestiones a tener en cuenta. En la *tercera etapa* se hace una descripción del problema de calibración, tomando como base el modelo *pin-hole* y analizando los algoritmos de calibración que se utilizarán en el proyecto. En la *cuarta etapa* se analiza el proceso de inserción de un objeto 3D en el mundo. Este proceso se divide en tres partes: insertar el objeto, renderizar la escena y la generación de la secuencia de vídeo. En este capítulo se hará una exhaustiva descripción de las alternativas planteadas en esta etapa y los diversos problemas que finalmente se solventaron. En la *quinta etapa* se describe el proceso de creación de la interfaz de usuario de la aplicación. Este capítulo incluye una fase de análisis donde se plantean las características del software; otra de diseño en la que se define la estructura de la aplicación (datos e interfaz) y por último, se comenta detalles de implementación de relevancia.

Finalmente se exponen los resultados y conclusiones obtenidas del proyecto. Se planteará los posibles trabajos futuros que se dejaron pendientes por cuestiones de tiempo y esfuerzo.

Hemos añadido un apéndice a la memoria que se compone de un único capítulo en el que se explican con todo lujo de detalles la implementación de cada una de las etapas del proyecto, incluyendo las alternativas y problemas que se nos plantearon, así como las soluciones adoptadas.

Capítulo 2

Estado actual del tema

En este capítulo se describe el estado actual del tema con referencia a trabajos anteriores o relacionados con nuestro proyecto.

Capítulo 3

Objetivos

3.1. Objetivos académicos

Una vez finalizados los estudios del Grado en Ingeniería en Informática en la Escuela de Ingeniería Informática perteneciente a la Universidad de Las Palmas de Gran Canaria, y con objeto de obtener la titulación que acredite los mencionados estudios se ha desarrollado, con el apoyo del tutor, el presente proyecto cuyo título es:

...

3.2. Objetivos generales del proyecto

Los objetivos que se persiguen en este proyecto se resumen en dos. En primer lugar, abordar el desarrollo e implementación de los algoritmos necesarios para llevar a cabo la inserción de objetos geométricos en una secuencia vídeo. En segundo lugar, desarrollar e implementar un interfaz de usuario que permita su utilización por un potencial usuario final.

La implementación del proyecto tiene dos factores críticos: el tiempo de cómputo y el uso de recursos.

Los objetivos generales se deben descomponer y asignar unos objetivos parciales a cada etapa para que el cumplimiento de éstos, repercuta en la consecución del objetivo del nivel superior.

Los principales objetivos que se han perseguido en cada etapa del proyecto son la precisión y fiabilidad. Los resultados de cada etapa serán los datos de entrada en la siguiente. Los errores, en el caso de producirse, se propagan y repercuten en los resultados finales.

...

Capítulo 4

Competencias Específicas Cubiertas

*) JUSTIFICACIÓN DE LAS COMPETENCIAS ESPECÍFICAS CUBIERTAS: indicar, para las especialmente relacionadas con el trabajo desarrollado, cómo se han cubierto con este TFT.

Capítulo 5

Aportaciones del TFG

*) APORTACIONES: justificar qué es lo que este TFG aporta a nuestro entorno socio-económico, técnico o científico.

Capítulo 6

Recursos utilizados

Durante el desarrollo del proyecto se ha utilizado una serie de recursos para llevar a cabo los objetivos. Los recursos se pueden distinguir en cuatro grupos: *hardware*, *software*, *librerías* y *lenguajes*.

6.1. Hardware

Los recursos hardware de este proyecto, son muy básicos. Bastaba simplemente con un ordenador personal. Se ha utilizado dos ordenadores, una estación de trabajo con dos pentium III Xeon a 1,7GHz y un ordenador personal pentium III a 1,1GHz. La estación se utilizaba, principalmente, para realizar los cálculos de la calibración ya que eran los que mayor tiempo de cómputo requerían. En el ordenador personal se hizo el diseño e implementación de todos los apartados del proyecto, ya que no se requería un gran potencia de cálculo.

Por otro lado, también se contó con una cámara digital con la que tomar las secuencias de imágenes.

6.2. Software

En la descripción de los recursos se ha hecho una distinción entre el *software* y *librerías*. A nuestro parecer, el software es un conjunto de herramientas que nos permiten realizar el proyecto, y nunca van a formar parte de él. A diferencia del software, las librerías van a quedar ligadas al proyecto de manera indefinida.

Sistema Operativo

El sistema operativo sobre el que se apoya la aplicación y las librerías del proyecto es Linux. *Linux* es un poderoso y sumamente versátil Sistema Operativo de 32 bits, multi-usuario y multi-área. Fue creado en 1991 por Linus Torvalds, siendo entonces un estudiante de la Universidad de Helsinki. Linux se basó sobre *Unix*.

Linux y toda la comunidad que gira alrededor de él hacen que sea una buena plataforma de desarrollo. Por un lado, se dispone un sistema operativo estable, seguro y en constante evolución. Por otro lado, existe las herramientas gratuitas de todo tipo y que además son de alta calidad.

...

6.3. Librerías

Las librerías son un software que resuelven un problema al programador, ocultándole la complejidad subyacente. De esta manera, el programador se apoya en las librerías para resolver su problema (de más alto nivel). Veamos las principales librerías que se han utilizado a lo largo del proyecto. La mayoría de ellas, en las dos últimas etapas.

STL

STL es la librería de plantillas estándar (*Standard Template Library*). Sus principales componentes son *contenedores*, *algoritmos*, *iteradores* y *objetos función*. Esta librería es parte del estándar ISO C++. STL es más que una librería, es un marco de trabajo (*framework*) y un paradigma en la programación.

Todas las estructuras contenedores del proyecto se han implementado usando *STL*. Como ya se describirá en el apéndice de la memoria las ventajas que aporta son múltiples.

...

6.4. Otras herramientas utilizadas

Lenguaje de programación C/C++

El lenguaje de programación por el que se ha optado para la implementación de este proyecto ha sido C++, ya que se pensó que un lenguaje orientado a objetos era la forma más adecuada de abordar las metas establecidas.

Este lenguaje es un superconjunto del lenguaje conocido como C, uno de los más extendidos en el mundo informático, utilizado por ejemplo para la implementación de sistemas operativos como Linux o la librería Qt. Fue diseñado por Bjarne Stroustrup a principios de los ochenta en AT&T Bell Laboratories, incrementando las características de C y añadiendo recursos para la programación orientada a objetos.

El paradigma de la programación orientada a objetos es ofrecer una serie de facilidades caracterizadas por:

- La abstracción de datos, habilidad de construir estructuras de datos para definir un objeto y usarlos dentro de un programa sin tener que atender a sus detalles internos.
- El encapsulamiento, que suministra el reforzamiento necesario para que sólo las funciones asociadas a las estructuras tengan acceso a los detalles internos.
- La herencia, que permite al programador definir nuevos objetos en términos de objetos previamente definidos. La habilidad de ser también objeto de la clase base se denomina *polimorfismo*.

C++ es cien por cien compatible con C. Es decir, se puede usar cualquier código escrito en C en programas desarrollados en C++ sin ningún problema. Por lo tanto, ofrece todas las posibilidades de C y algunos añadidos como la abstracción de datos, verificación estricta de tipos, paso de argumentos por referencia, sobrecarga de operadores y funciones, declaración de las variables en cualquier lugar, tipos genéricos o parametrizados (templates), tratamiento de excepciones y conversión de tipos.

Una vez elegido el lenguaje de programación debía elegir el compilador a usar. El GNU C++ es el compilador de C++ que viene por defecto en todas las distribuciones de Linux.

UML

El Lenguaje Unificado de Modelado (UML, Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona

una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.

UML se ha utilizado en el proyecto para modelar la aplicación principal del proyecto. Apoyándonos en el UML se extrajo una primera aproximación de los requerimientos y de cómo debía ser la estructura de la aplicación.

L^AT_EX

L^AT_EX es un paquete de macros que le permite al autor de un texto componer e imprimir sus documentos de un modo sencillo, con la mayor calidad tipográfica, utilizando para ello patrones previamente definidos. L^AT_EX desempeña el papel del diseñador tomando parte en el formato del documento (longitud del renglón, tipo de letra, espacios, etc) para darle luego instrucciones a T_EX.

El tratamiento del texto es totalmente diferente a procesadores tales como *Microsoft Word*, *Word Perfect* o *Framemaker* en los cuales el autor ve en la pantalla exactamente lo que luego aparecerá por la impresora. Esto tiene sus ventajas e inconvenientes. Aún así L^AT_EX tiene la posibilidad, una vez procesado el fichero, de ver el resultado final por la pantalla.

L^AT_EX se ha utilizado para crear toda la documentación del proyecto.

6.5. Requerimientos del software

Una vez leído, cuáles han sido los recursos más significativos que se han utilizado para llevar a cabo todas las etapas del proyecto, veamos los requerimientos del producto final, AMICam.

- Sistema operativo Linux.
- Librería Qt, versión 3.0 ó superior.
- Open Inventor, versión 2.1.5 en la implementación de SGI.
- SoQt, versión 1.0 ó superior.

Capítulo 7

Plan de trabajo y temporización

En este capítulo se presenta el plan de trabajo desglosado en etapas, con una estimación en cada etapa del tiempo de ejecución.

- Etapa 1. ...

El tiempo invertido en esta etapa fue de 40 horas.

- Etapa 2.

El tiempo invertido en esta etapa fue de 380 horas.

- Etapa ...

El tiempo empleado en esta etapa fue de 100 horas.

El tiempo invertido en la escritura de esta documentación fue de 200 horas.

Por tanto, el tiempo total de dedicación al proyecto ha sido de unas xxxx horas.

Capítulo 8

Etapal. Algoritmo de Harris

.....

La primera etapa del proyecto consiste en la implementación de un algoritmo para la extracción de puntos singulares en una imagen. En especial nos centraremos en el algoritmo de Harris. Este detector de puntos destaca por su rapidez de cómputo y la precisión de los datos obtenidos.

En este capítulo haremos una breve introducción a los distintos tipos de detectores de puntos singulares que existen, destacando sus principales características. Describiremos la teoría matemática que subyace al detector de Harris. Por último, se mostrarán y analizarán los resultados experimentales que se han obtenido.

8.1. Detectores de esquinas

Una imagen contiene una gran cantidad de datos, la mayoría de los cuales proporciona muy poca información para interpretar la escena. Nuestro proceso debe extraer de la forma más eficaz y robusta posible determinadas características que nos proporcionen la máxima información. Estas características deben cumplir, entre otras, las siguientes condiciones:

- Su extracción a partir de la imagen no debe suponer un coste excesivo. El tiempo total de extracción debe ser lo más pequeño posible.
- Su localización debe ser muy precisa. El error cometido en la estimación de las características también debe ser lo más pequeño posible.
- Deben ser robustas y estables. Deberían permanecer a lo largo de una secuencia.

Las esquinas de los objetos presentes en la imagen satisfacen estas condiciones. Aparecen de forma natural en la mayoría de escenarios tanto naturales como artificiales. Un punto esquina es un punto del contorno de un objeto donde la curvatura es alta.

Desde el punto de vista computacional, se han propuesto una serie de enfoques para la detección de este tipo de características bidimensionales. Podemos clasificar estos enfoques en dos grupos principales:

1. Métodos que obtienen las aristas de la imagen mediante algún método de detección de aristas, para, a continuación, detectar puntos de cruce entre aristas o con un cambio sustancial en la dirección de la arista. Estos puntos se clasifican como puntos esquina.
2. El resto de métodos trabajan directamente sobre las imágenes, es decir, las esquinas no se infieren a partir de la extracción de aristas.

Puntos de interés extraídos a partir de aristas

Todos los métodos de obtención de esquinas y/o uniones dentro de este grupo tienen en común una primera fase de extracción de aristas. Algunos métodos de extracción de aristas pueden ser *Torre-Poggio*, *Canny* o *Smith-Brady*. En general, la eficiencia de estos métodos dependen directamente de la calidad del método empleado para la obtención de aristas: si este último no localiza correctamente los puntos de arista, difícilmente podremos detectar puntos de esquina de forma exacta. Además se añade un tiempo extra de procesamiento que en ciertos sistemas puede ser prohibitivo.

Capítulo 9

Seguimiento de puntos singulares a través de una secuencia de vídeo

En este capítulo vamos a estudiar el funcionamiento del proceso de seguimiento de puntos singulares a través de una secuencia de vídeo. El seguimiento consiste en hallar la trayectoria de los puntos singulares a través de los distintos frames. La trayectoria de un punto 3d consiste en una lista con las proyecciones de dicho punto en los distintos frames donde es visible.

Este capítulo se divide en tres partes: definición de conceptos, criterios de sucesión y la descripción del proceso de seguimiento. El apartado de definición de conceptos es imprescindible para introducir al lector en la terminología que se maneja. Los criterios de sucesión definirán el comportamiento que tendrá el proceso de seguimiento.

9.1. Proceso del seguimiento

9.1.1. Conceptos

A lo largo de los siguientes apartados manejaremos una serie de conceptos por lo que necesitaremos entenderlos perfectamente. A continuación se definen dichos conceptos:

- **Detector de esquinas:** Herramienta software que nos identifica un conjunto de esquinas (extremos de curvatura) en una imagen.

- **Frame:** Instantánea que recoge información captada por una cámara (ya sea de vídeo o de fotografía).
- **Parámetros de Harris:** El algoritmo de Harris es el detector de esquinas que utilizaremos el seguimiento de los puntos. Sus parámetros han sido descritos en el capítulo anterior.
- **Punto sucesor:** Punto continuación de otro situado en el frame anterior que corresponde al mismo punto 3d.
- **Secuencia de puntos:** Valor devuelto en el seguimiento, en que se identifica como una lista de puntos aquellas proyecciones de frames consecutivos que correspondan al mismo punto 3d.
- **Secuencia de vídeo:** Conjunto de frames desglosados uno independiente del otro.

Capítulo 10

Resultados

En cada uno de los capítulos anteriores se han comentado de manera exhaustiva los resultados empíricos obtenidos en cada etapa. En este último apartado vamos a globalizar el resultado final aplicado a una secuencia de vídeo.

Partimos de una secuencia de 120 imágenes reales de un despacho. Cada una de estas imágenes fueron tomadas por una cámara digital con un zoom constante. La cámara está situada en el fondo del despacho y se va a ir desplazando paralelo a la pared. El movimiento seguido por la cámara se asemeja a una recta. La diferencia entre dos imágenes consecutivas es pequeña. Al no tomarse esta secuencia con una cámara de vídeo digital, se observarán pequeño saltos entre imágenes por lo que el movimiento no será fluido.

La cámara se ha situado encima de un soporte móvil y es éste quien se mueve. De esta manera se evita el cabeceo de la cámara. La fuente de luz en la escena está situada detrás de la cámara y la imagen tomadas son de buena calidad y de gran claridad.

En las figuras ??, ?? y ?? podemos observar una mini secuencia de seis imágenes. Estas imágenes son un subconjunto de secuencia de vídeo inicial (120). Cada una de estas imágenes pertenece a distintas partes de esta secuencia. En concreto se han tomado las imágenes de las posiciones 0, 30 50, 90, 100 y 119.

Capítulo 11

Conclusiones

El proyecto fin de carrera tiene por objetivo la elaboración de un proyecto informático aplicando los conocimientos adquiridos a lo largo de las asignaturas de la Ingeniería en Informática.

La inserción de objetos geométricos en una secuencia de vídeo, perseguía dos objetivos muy claros: (1) desarrollo e implementación de los algoritmos necesarios para llevar a cabo esta tarea y (2) desarrollar e implementar un interfaz de usuario que permitiera su utilización por un potencial usuario final.

Como se ha visto a lo largo de esta memoria, los objetivos iniciales han sido superados, ya que, no sólo se han implementado los requisitos de cada etapa del proyecto, sino que:

- Cada etapa del proyecto se ha diseñado utilizando una estructura modular, facilitándose su mantenimiento y reutilización.
- La funcionalidad prevista se ha aumentado con el fin de que el usuario disponga del mayor número de herramientas posible.
- Se han optimizado ciertos procesos del proyecto con el objetivo de reducir el tiempo de cómputo y el uso de recursos.

El tiempo transcurrido desde la elección del proyecto hasta la finalización del mismo ha sido de 18 meses. De ese tiempo, alrededor de 11 meses es el tiempo total invertido en el proyecto. El resto del tiempo se ha empleado en el aprendizaje de numerosas herramientas como son las librerías Qt y Open Inventor.

Con respecto al diseño, hemos tenido que replantear en varias ocasiones las estrategias utilizadas lo que ha supuesto un aumento considerable de horas de trabajo respecto al plan inicial previsto. También nos hemos

tenido que enfrentar a numerosos problemas que a priori no sabíamos si tenían solución. Las etapas en las que más tiempo se invirtió fueron la de la creación del interfaz y la de inserción de un objeto en la escena. La creación del interfaz es un proceso largo y laborioso. El desarrollo del software se planteó siguiendo cada una de las fases que dicta la ingeniería del software.

Un punto a destacar es que la creación de una interfaz amigable y de fácil manejo como es AMICam, implica la gestión de numerosos eventos y detalles. Esa funcionalidad se traduce en tiempo de desarrollo y depuración. Finalmente, AMICam se convirtió en una interfaz donde el peso de la gestión recae en el sistema y no en el usuario. En el proyecto nos hemos centrado en el sistema operativo Linux. La posibilidad de ser ejecutada sobre el sistema operativo Windows no se tuvo en cuenta ya que no formaba parte de los objetivos del proyecto. A priori el proyecto es totalmente transportable a otras plataformas. Sin embargo no estoy seguro si la librería de calibración utiliza otras librerías que sólo están implementadas para Linux.

Sin embargo, la etapa más compleja debido al desconocimiento de la herramienta y cómo resolver el problema que se planteaba, fue sin dudas la inserción de un objeto geométrico en la escena. El hecho de encajar la proyección de una cámara real con la de una cámara artificial, fue el problema más importante que se abordó en el proyecto. Tal fue la complejidad, que se planteó la posibilidad de crear un pequeño motor de render. Debido a las expectativas que ofrecía Inventor, se insistió y al final se consiguió este objetivo.

....

En definitiva, el desarrollo de este proyecto ha significado un gran objetivo a nivel personal, he aprendido y comprendido todos los conceptos y problemas que rodean la combinación de imágenes reales y sintéticas en una secuencia vídeo.

Capítulo 12

Trabajo Futuro

Durante este proyecto fin de carrera se nos han planteado otros trabajos que se podían desarrollar con el fin de mejorar los resultados parciales y finales, pero que por cuestiones de tiempo y esfuerzo no hemos podido realizar. Entre estos podemos destacar:

Ampliar los formatos del vídeo. En el proyecto se considera el vídeo como una secuencia de imágenes separadas y en formato imagen. Sería deseable que AMICam soportara un mayor número de formatos de vídeo (tanto nuevos formatos de imágenes como formatos de vídeo digital). En especial, el trabajo futuro debería estar orientado a dar soporte de vídeo digital. El único cambio a realizar en la aplicación sería el de incorporar librerías que leyesen el vídeo. Se debe dar soporte, al menos, a los formatos más conocidos como son el *.avi* y el *.mpeg*.

Se espera que la próxima versión de la librería gráfica *Inventor*, de soporte para grabar secuencias de vídeo en formato *mpeg*.

Apéndice A

Detalles técnicos sobre la implementación del proyecto

En este capítulo se expone todo lo relacionado con los diferentes detalles técnicos de la implementación del proyecto. Se comienza introduciendo las decisiones más importantes que se han tomado a lo largo de las fases de diseño, desarrollo y pruebas.

Dada la naturaleza gráfica del proyecto, se escogió para la implementación el sistema operativo Linux y su entorno *X Window*, frente por ejemplo al entorno MS-Windows, tanto por la disponibilidad de máquinas sobre las que trabajar, de las cuales he obtenido gran experiencia, como de las ventajas en cuanto a fiabilidad y posibilidad de portar el código a otras implementaciones de S.O. que a su vez ofrecen, en su conjunto, un amplio abanico de plataformas hardware.

El XMegaWave es un entorno, con una interfaz gráfica, para el procesamiento gráfico de imágenes. El usuario puede interactuar con la interfaz mediante ratón y pop-up menús. Además se puede incluir en el entorno funciones creadas por propio usuario. El XMegaWave puede trabajar con vídeos (en un formato especial) y con imágenes, tanto en color como en escala de grises. Está orientado a estaciones UNIX con X11 y librerías gráficas Motif.

A.1. Uso de las STL (Standard Template Library)

La STL es la librería estándar de plantillas. Sus principales componentes son contenedores, algoritmos, iteradores y objetos funciones. Esta librería forma parte del Standard ISO C++. En la librería que implementa el seguimiento de puntos se ha utilizado los contenedores típicos (listas y vectores)

y los iteradores que acceden a los datos (similar a los punteros).

Las principales estructura de almacenamiento se ha utilizado ya que aporta muchas ventajas y pocos inconvenientes:

Ventajas:

1. Facilidad de uso y flexibilidad.

Las operaciones de inserción, extracción y modificación son intuitivas y con una interfaz de programación muy sencilla. Permite una gran variedad de operaciones y búsquedas de alto nivel dentro de la estructura.

2. Optimización de recursos.

Aleja al programador de toda la complejidad del manejo y conocimiento de la estructura subyacente. Por ello, se puede optimizar las operaciones sobre la estructura de manera transparente al programador.

3. Fiabilidad.

Se trata de una estructura ampliamente probada y que permite al programador centrarse en otros problemas que no sean el almacenamiento de datos, por lo que los esfuerzos de desarrollo y depuración irán destinados en otro sentido.

Inconvenientes:

1. Necesario un compilador de C++.

Las STL son unas librerías desarrolladas en C++, por lo que cualquier programa que las utilice también debe estar escrito en C++ y necesita un compilador de C++.

Bibliografía

- [1] Faugeras, Olivier, *Three-Dimensional Computer Vision* The MIT Press. (1993).
- [2] Faugeras, Olivier and Quang-Tuan Luong and Theo Papadopoulos, *The geometric of multiples images: the law that govern the formation of multiple images of a scene and some of their applications* The MIT Press. (2001).
- [3] Kenichi Kanatani, *Geometric Computation for Machine Vision* Oxford University Press. (1995).
- [4] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision* The MIT Press. (2001).
- [5] Alvarez, Luis and Cuenca, Carmelo and Mazorra, Luis, *Signal Processing, Pattern Recognition and Applications. IASTED*. Morphological Corner Detector. Application to Camera Calibration. (2001).
- [6] Roger S. Pressman, *Ingeniería del Software, quinta edición* McGraw-Hill Companies. (2001).
- [7] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns* Addison Wesley Publishing Company. (1994).
- [8] Gregory L. Heileman, *Estructuras de datos, algoritmos y programación orientada a objetos*. McGraw-Hill/Interamericana de España S.A.U. 1998.
- [9] Scott Meyers, *Effective C++* Addison Wesley Publishing Company. (1997).
- [10] Josie Wernecke, *The Inventor Mentor* Addison Wesley Publishing Company. (1994).

- [11] Josie Wernecke, *The Inventor Toolmaker* Addison Wesley Publishing Company. (1994).
- [12] Richard S. Wright Jr., *Programación en OpenGL* Anaya Multimedia. (1997).
- [13] Grady Booch, James Rumbaugh and Ivar Jacobson, *El lenguaje unificado de modelado* Addison Wesley Publishing Company. (2000).
- [14] Martin Fowler and Kendall Scott, *UML gota a gota* Addison Wesley Publishing Company. (1997).
- [15] Antoni Diller, *Latex line by line* Wiley Professional Computing. (1993).