

LIBRERÍA PARA EL PROCESAMIENTO DE SEÑALES DIGITALES CON COMPUTADORA

Francisco J. Madrid Cuevas, Rafael Medina Carnicer, Ángel Carmona Poyato
y Nicolás L. Fernández García
UNIVERSIDAD DE CÓRDOBA

Departamento de Informática y Análisis Numérico

malmacuf@uco.es

"Enseñar haciendo que lo complicado parezca sencillo es la labor más
gratificante que un ser humano puede desear"



LIBRERÍA PARA EL PROCESAMIENTO DE SEÑALES
DIGITALES CON COMPUTADORA

Procesamiento de Señales Digitales
y su Aplicación en Computadoras
Departamento de Ingeniería y Ciencias
Eléctricas



LIBAVA PARA EL PROCESAMIENTO DE SEÑALES EXTRAS CON COMPUTADORA

Francisco J. Madrid Cuevas, Rafael Medina Carnicer,

Ángel Carmona Poyato y Nicolás L. Fernández García

Universidad de Córdoba

Resumen

Palabras clave

Introducción

LIBAVA: Criterios de diseño y desarrollo

LIBAVA: Una nueva forma de generar ayuda para autoaprendizaje de forma automática

Resultados

Conclusiones

Agradecimientos

Bibliografía



LIBRERÍA PARA EL PROCESAMIENTO DE SEÑALES DIGITALES CON COMPUTADORA

Francisco J. Madrid Cuevas, R. Medina
Carnicer, Ángel Carmona Poyato y Nicolás
L. Fernández García

Fecha de entrega: 23 marzo 2004

Fecha de aceptación: 20 abril 2004

RESUMEN

En este trabajo se exponen las características básicas del desarrollo y la funcionalidad de un software específico para la enseñanza en cualquier materia que incluya entre sus contenidos el "procesamiento digital de señales". El software ha permitido que los alumnos de la Universidad de Córdoba que cursan los estudios de Ingeniero en Automática y Electrónica Industrial, Ingeniería Técnica en Informática de Sistemas e Ingeniería Técnica en Informática de Gestión, puedan simular los contenidos teóricos correspondientes a materias con la temática comentada anteriormente, sin necesidad de requerir adicionales medios informáticos de la Universidad.

La principal ventaja del producto desarrollado estriba en la limitación del tiempo requerido para su aprendizaje. En el ámbito de la enseñanza práctica, ha sido comprobado que este tiempo no excede las tres horas, teniendo en cuenta que para estos fines el software puede ser utilizado sin necesidad de tener conocimientos de programación. En el ámbito de la investigación, puede ser utilizado como plataforma de desarrollo, siendo necesario tener conocimientos de programación en el lenguaje C++.

ABSTRACT

In this work the basic characteristics of the development and the functionality of a specific software are exposed for the teaching in any matter that includes among their contents the "digital signal processing." The software has allowed that the students of the University of Córdoba that study the studies of Engineer in Automatic and Electronic Industrial, Technical Engineering in Computer science of Systems and Technical Engineering in Computer science of Administration, they can simulate the theoretical contents corresponding to matters with the thematic one commented previously, without necessity of requiring additional computer means of the University.

The main advantage of the developed product rests in the limitation of the time required for its learning. In the environment of the practical teaching, it has been proven that this time doesn't exceed the three hours, keeping in mind that for these ends the software can be used without necessity of having programming knowledge. In the environment of the investigation, it can be used as development platform, being necessary to have programming knowledge in the language C++.

PALABRAS-CLAVE

Software docente
Análisis de señales digitales

KEYWORDS

Software to Teach
Digital Signal Processing

INTRODUCCIÓN

El informe sobre la adaptación de los Estudios de las Ingenierías en Informática a la Declaración de Bolonia efectúa una propuesta de refundición de los actuales estudios de Ingenierías en Automática y Electrónica, Telecomunicaciones, e Informática en un primer ciclo de 4 años de duración, y un segundo ciclo de duración 1-2 años.

En el mismo informe, sobre la base del supuesto anterior, se propone una estructura porcentual sobre las materias de “Redes de Computadoras” del 4% y en el caso de las materias denominadas “Sistemas de Percepción”, englobadas en la temática de la Inteligencia Artificial, de un 2% sobre el total. Asimismo, en el informe se propone un segundo ciclo, inspirado en la propuesta del Carrer Space, en el que se incluye la temática del “Diseño de aplicaciones de Procesamiento Digital de la señal” como una de las posibles especializaciones a establecer.

Los contenidos incluidos en las materias anteriores tienen de común que en todos los casos se necesitan conocimientos, con una alta componente práctica, de lo que se suele denominar “Tratamiento y estudio de señales digitales por computadora”. Distintos campos científicos, como pueden ser la Comunicación de Datos o la Visión por Computadora, requieren también de este tipo de conocimientos.

Tradicionalmente, la enseñanza práctica de este tipo de materias se ha hecho utilizando software específico para cada una de ellas. Desde un punto de vista histórico el alumno debía hacer directamente programas de ordenador para ejercitar los conocimientos teóricos o, en casos específicos como el tratamiento digital de imágenes, utilizar paquetes de software que eran gratuitos para usos académicos.

A lo largo de la última década se ha comprobado que este tipo de organización docente no es la ideal porque el tiempo necesario para que un alumno desarrolle programas de ordenador para practicar los conceptos teóricos es excesivo y además estos desarrollos no cumplen ningún objetivo marcado en estas materias (no son asignaturas de programación).

Como efecto añadido debemos citar, que incluso en aquellos casos en los que es posible utilizar un software específico ya desarrollado (por ejemplo Matlab), existen pocas alternativas de esta temática que sean

gratuitos y en general los mismos son de aprendizaje y uso complicado. Además estos paquetes gratuitos suelen funcionar sobre computadoras con Sistema Operativo Unix, necesitan muchos recursos de las mismas, y como consecuencia limitan el número de instancias en ejecución concurrentes.

Todos los factores indicados se agravan si pensamos que en el contexto de la Declaración de Bolonia se pretende “enseñar a aprender” en lugar del modelo clásico de aprendizaje basado en la explicación del profesor. En este nuevo contexto aparece una necesidad acuciante de generar materiales que permitan el autoaprendizaje del alumno, cubriendo sólo los objetivos específicos de una materia, con el menor tiempo posible.

En este trabajo exponemos los conceptos básicos de diseño y desarrollo de una herramienta software construida bajo forma de librería de operadores. La herramienta la hemos denominada LIBAVA y hace posible el tratamiento de una señal digital en una computadora, independientemente del campo aplicativo, de una forma muy intuitiva, sin necesidad de programar nada. En la sección 2 mostramos los conceptos básicos de LIBAVA, ejemplos de su utilización y las características fundamentales de la misma. En la sección 3 mostramos las características de la documentación de ayuda que posee la librería, fundamental para su aprendizaje, así como la estructura de la misma. Finalmente, en las secciones 4 y 5 citamos los resultados obtenidos en una experiencia docente realizada con la librería así como las conclusiones y principales aportaciones de este trabajo.

LIBAVA: CRITERIOS DE DISEÑO Y DESARROLLO

Puesto que el concepto de señal digital es independiente de un dominio específico LIBAVA trata dicho concepto de forma genérica. En este sentido para LIBAVA una señal puede ser una imagen o bien un fichero de texto en donde se incluyen una serie de datos que provienen de las mediciones de un sensor.

LIBAVA define un formato interno de datos (reconocido por la extensión *ava*). La propia librería contiene un operador para transformar un fichero de datos en formato *ava*. También se dispone de otro operador que

transforma una imagen en formato .png (formato muy popular y que la mayoría de las herramientas gráficas reconocen) al formato interno .ava.

A partir de esta generalización LIBAVA se estructura en módulos (pequeños algoritmos) ejecutables que manipulan las entradas en formato .ava para múltiples propósitos específicos. La utilización de estos módulos ejecutables se realiza sobre el propio sistema operativo a nivel de comando. Todos los comandos incluyen parámetros de entrada (ficheros a procesar), parámetros de salida (ficheros obtenidos) y parámetros internos. Por ejemplo, el comando "istats -i gaviota.ava -o estadistica.txt -whole" toma el fichero denominado "gaviota.ava" y obtiene estadísticos sobre él, tales como media, mediana, varianza, curtosis, etc..., en el fichero estadistica.txt. El comando "imk_convolve -i1 gaviota.ava -i2 gauss.ava -o gaviota2.ava" permite convolucionar el fichero gaviota.ava con el fichero denominado gauss.ava obteniendo como fichero de salida el llamado gaviota2.ava.

LIBAVA descarga los procesos de visualización de resultados (imágenes, histogramas etc...) en otras herramientas. Para ello dispone de dos conversores (uno para imágenes en formato .ava y otro para ficheros de texto en formato .ava) a archivos de tipo png ó .txt. De esta forma, el usuario puede utilizar las herramientas de visualización de resultados que le sean más usuales sin necesidad de aprender nada nuevo.

En las figuras 1 y 2 se muestran dos ejemplos que muestran las capacidades de tratamiento de la librería. En la primera se muestra un resultado clásico del procesamiento de imágenes: como eliminar ruido en una imagen. En el segundo se muestra el mismo concepto, pero en este caso aplicado a una señal digital cualquiera.



figura 1: EJEMPLO DE ELIMINACIÓN DE RUIDO EN UNA IMAGEN

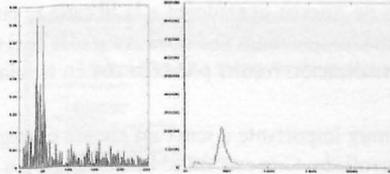


figura2: EJEMPLO DE ELIMINACIÓN DE RUIDO EN UNA SEÑAL

Desde el punto de vista de la programación de la librería se ha utilizado el paradigma del análisis y diseño de software orientado a objetos (Meyyer 1997). Para la implementación se ha utilizado el lenguaje C++ (Stroustrup, 1997). Se ha seleccionado este paradigma porque tiene una serie de ventajas frente al paradigma tradicional de descomposición funcional.

La librería de software para el tratamiento de señales digitales con computadora desarrollada, incorpora como características principales:

a) Un conjunto de operadores que permita la aplicación directa de técnicas de procesamiento digital de la señal sin necesidad de programar nada. Consta de los siguientes módulos:

- 1) Módulo para representación de señales digitales en una y dos dimensiones.
- 2) Módulo para generar señales digitales unidimensionales y bidimensionales según diferentes esquemas de codificación analógica y digital.
- 3) Módulo para el análisis y procesamiento espacio/temporal de de señales digitales.
- 4) Módulo para el análisis y procesamiento espacio/frecuencial de señales digitales.

b) Facilidad de extensión de la librería con nuevas funciones.

El diseño realizado ha conseguido que la librería sea en realidad una plataforma para el desarrollo de programas óptimos para el tratamiento digital de la señal.

c) Facilidad de uso e independencia de la plataforma. Puede ser ejecutada bajo S.O. Unix o MsWindows.

La incorporación de nuevos operadores a la librería es muy asequible a cualquier programador que tenga nociones avanzadas de C++, por lo que su ampliación/actualización resulta poco costosa en términos de desarrollo.

Un aspecto muy importante a tener en cuenta es asegurar la calidad del software desarrollado. Con este fin se han utilizado dos herramientas:

1) Se ha utilizado el paradigma de la programación por contrato (Meyyer 1997) en el desarrollo de las interfaces de las clases.

2) Se han implementado pruebas de regresión para asegurar que la refactorización/adición de funcionalidad a la librería no afecta a lo ya desarrollado. Para ello se ha utilizado la herramienta CppUnit.

LIBAVA: UNA NUEVA FORMA DE GENERAR AYUDA PARA AUTOAPRENDIZAJE DE FORMA AUTOMÁTICA

Con relativa frecuencia, poderosas herramientas software no son utilizadas demasiado debido a su dificultad de aprendizaje y a la difícil ayuda en línea que poseen. Ambos aspectos han sido considerados en el diseño y desarrollo realizado, debido al carácter fuertemente docente en el que se pretende utilizar LIBAVA.

A estos efectos se ha utilizado la herramienta de auto documentación Doxygen (Heesch). Esta herramienta constituye una poderosa forma de generar documentación de ayuda en línea a partir del código desarrollado. La documentación que se genera puede ser visualizada a través de cualquier navegador WEB, puesto que está en formato html.

La FIGURA 3 muestra la página inicial de la documentación.

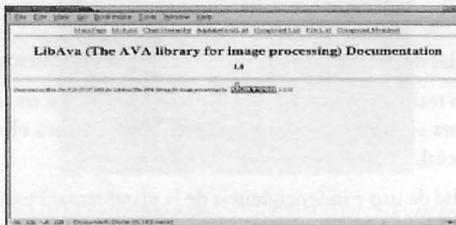
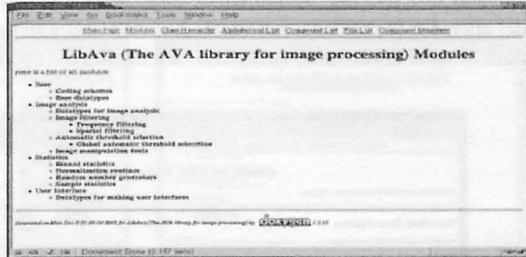


figura 3: PÁGINA PRINCIPAL DE LA AYUDA

figura4: CLASIFICACIÓN POR CATEGORÍAS



La documentación se estructura en 6 partes diferentes, denominadas Modules, Class Hierarchy, Alphabetical List, Compound List, File List y Compound Members. En el apartado Modules, como puede verse en la figura4, se muestran las 4 categorías principales, y las subcategorías respectivas, en las que se encuentran todos los operadores de la librería. Los títulos de las mismas expresan claramente la funcionalidad de los operadores que en ellas se incluyen.

Cuando se activa una cualquiera de las categorías o subcategorías aparece la ayuda correspondiente a la misma. En la figura5 se muestra como ejemplo la pantalla de ayuda correspondiente a la categoría "Spatial filtering". Como puede observarse, en la misma aparecen relacionados todos los operadores (con extensión cc) que constituyen la categoría junto a una breve descripción de su funcionalidad.



figura5: OPERADORES INCLUIDOS EN LA CATEGORÍA SPATIAL FILTERING

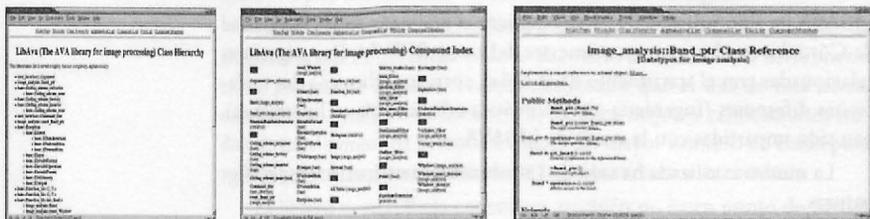


figura 8: JERARQUÍA DE CLASES, RELACIÓN ALFABÉTICA DE CLASES Y EJEMPLO DE CONSULTA DE UNA CLASE CONCRETA

En cualquier momento se puede acceder a la jerarquía de clases construidas, a partir de la página principal, o a una relación alfabética de todos los operadores. También es posible acceder a la información exacta de una clase (miembros y procedimientos) consultando si se desea cualquiera de ellos. La figura 8 muestra las correspondientes pantallas.

Hay que resaltar que toda la documentación y la estructuración de la misma es generada de forma automática por la herramienta doxygen. Sólo hay que seguir las pautas de generación de código C++ que pueden consultarse en cualquiera de los operadores existentes en la librería para que doxygen incorpore de forma automática la documentación del nuevo operador, y lo clasifique alfabéticamente y dentro de una determinada categoría. Este paradigma de documentación favorece enormemente el aprendizaje de la librería.

Aunque el entorno de desarrollo utilizado ha sido un PC estándar con sistema operativo GNU/Linux, distribución Debian, para facilitar su portabilidad a distintos entornos de desarrollo se han utilizado la herramienta autoconf. Además la portabilidad a las plataformas Unix y MsWindows es muy simple utilizando el entorno Cygwin.

Finalmente indicar que la librería se ha desarrollado bajo licencia GPL (The General Public License) por lo que se da permiso para su instalación, distribución, modificación y estudio bajo los términos indicados por esta licencia.

RESULTADOS

LIBAVA ha sido utilizada en una experiencia realizada en la Universidad de Córdoba en el primer cuatrimestre del curso 2003/04. Dos asignaturas relacionadas con el tratamiento de la señal, correspondientes a dos titulaciones diferentes (Ingeniería en Automática e Ingeniería en Informática), han sido impartidas con la ayuda de LIBAVA.

La muestra utilizada ha sido de 150 alumnos matriculados en las asignaturas.

Nuestra experiencia comenzó destinando un total de 3 horas para explicar los conceptos básicos de LIBAVA y la forma de utilizar los operadores ya implementados. A partir de aquí se le suministro a cada alumno la librería para que fuera instalada por ellos mismos en sus ordenadores personales.

Las clases teóricas fueron desarrolladas con la ayuda de un ordenador portátil mostrando a todos los alumnos ejemplos reales de los conceptos que debían asimilar. Se propusieron un total de 25 ejercicios diferentes, no explicados en clase, a lo largo del cuatrimestre. Posteriormente se propuso un examen de 5 ejercicios diferentes.

Las calificaciones obtenidas por los alumnos en el examen, con respecto a las obtenidas en los dos cursos anteriores en las mismas asignaturas y con un número similar de alumnos, muestran que en media obtuvieron un 20% de calificación superior. La materia impartida en ambas asignaturas ha abarcado un 15% más de lo que se pudo impartir en los cursos precedentes, y finalmente se ha incrementado sustancialmente el número de alumnos que optan por realizar el Proyecto Fin de Carrera (obligatorio en ambas titulaciones) en una temática relacionada con las asignaturas objeto del estudio.

CONCLUSIONES

En el ámbito del desarrollo de herramientas software que favorezcan el autoaprendizaje de los alumnos universitarios se ha diseñado y desarrollado un producto, que bajo forma de librería de operadores, permite el

autoaprendizaje práctico de cualquier materia que incluya entre sus contenidos el tópic denominado “tratamiento de señales digitales”.

Como consecuencia del diseño realizado se ha contrastado que un alumno es capaz de aprender todo lo necesario para utilizar la herramienta en un máximo de 3 horas. Hemos constatado que el uso de esta herramienta como ayuda en la docencia a) implica mejores calificaciones medias de los alumnos, b) permite explicar un mayor número de conceptos, y c) en general, motiva mucho mas a los alumnos en esta temática.

La librería desarrollada constituye también un buen punto de partida como herramienta software para ser utilizada en investigación. La implementación de nuevos algoritmos (en forma de operadores) es extremadamente simple para un programador que tenga unos conocimientos mínimos de C++ y del paradigma de programación orientada a objetos, por lo que su ampliación es en términos generales poco costosa en tiempo.

El paradigma empleado para generar documentación (ayuda) de la librería de forma automática favorece mucho la aplicabilidad docente de la misma, frente a otro tipo de herramientas.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto 02NP068 correspondiente a la Convocatoria de Ayudas a la Innovación y Mejora de la Calidad Docente de la Universidad de Córdoba.

BIBLIOGRAFÍA



Autoconf - Produces shell scripts which automatically configure source code. <http://www.gnu.org/software/autoconf.html>.

Cppunitdocumentation. <http://cppunit.sourceforge.net/doc/1.8.0/index.html>.

Cygwin. <http://www.cygwin.com>.

Debian GNU/Linux distribution. <http://www.debian.org>.

HEESCH, V., *Doxygen manual*. <http://www.stack.nl/~dimitri/doxygen/manual.html>.

MEYER, B. (1997), *Construcción de Software Orientado a Objetos*. Prentice Hall.

STROUSTRUP, B. (1997), *The C++ programming language*. Addison-Wesley.

The General Public License. <http://www.fsf.org/licenses/licenses.html>.