

ALGORITMOS INTERACTIVOS PARA LA SEPARACIÓN DE FUENTES Y DECONVOLUCIÓN DE CANAL MEDIANTE JAVA

David Jiménez i Reifs(*), Jordi Solé i Casals(*), Christian Jutten(**)

(*) Departament de Teoria del Senyal - Universitat de Vic - Sagrada Família, 7, 08500, Vic (Catalunya, España)
david.jimenez@uvic.es

(**) Laboratoire des Images et des Signaux - Institute National Polytechnique de Grenoble - 38, av. Felix Viallet,
38000 Grenoble (France)

RESUMEN

El objetivo de estos programas es crear una herramienta que nos permita, de una manera fácil, entender mejor la separación de fuentes y la deconvolución de canal. Por eso se presenta el diseño, mediante Java, de una página web [1]:

<http://www.uvic.es/proyectos/SeparationSources>
con carácter marcadamente didáctico para el estudio y evaluación de diferentes algoritmos propuestos en la bibliografía.

1. INTRODUCCIÓN

En una definición general, la separación de fuentes consiste en recuperar unas señales desconocidas, las fuentes, observando solamente una transformación de dichas fuentes.

La deconvolución de canal consiste en anular el efecto producido por un canal de transmisión en una señal conociendo solamente la señal filtrada.

Uno de los objetivos funcionales que se plantearon en el momento de escoger el lenguaje de programación fue la multiplataforma. Es por ello que los hemos implementado en Java 'puro', para que el usuario pueda ejecutarlos desde la red Internet sea cual sea su plataforma de conexión.

Para poder ejecutar la aplicación se requiere un visualizador de *applets* o un navegador equivalente a Netscape 4.5, Pentium MMX-166 y una resolución de 1024x768.

2. CARACTERÍSTICAS DE PROGRAMACIÓN

Las principales características son:

- Multiplataforma: Los algoritmos se pueden ejecutar en cualquier máquina, sin restricción de sistema operativo o fabricante (PC, MAC...)
- Accesibilidad: De una manera rápida el usuario puede acceder, o a través de internet o ejecutando un intérprete de *applets*, a los programas de simulación.
- Implementación WWW: La ventaja de implementar los algoritmos en *applets* es la difusión que podemos hacer de ellos. Mediante una página web, este software es accesible a cualquier persona. De una manera fácil podemos añadir un pequeño tutorial que ayudará al usuario a entender mejor el funcionamiento de cada programa.
- Velocidad y seguridad: Al ejecutar *applets*, el usuario sólo descarga el código (relativamente

pequeño) a través de la red, y la máquina del usuario es la encargada de hacer todos los cálculos. La seguridad en Java nos garantiza que en la máquina del usuario sólo habrán los archivos compilados, respetando de esta forma la confidencialidad y la integridad del código.

3. CARACTERÍSTICAS FUNCIONALES

Hemos implementado un total de 4 simulaciones que básicamente las podemos agrupar en dos: separación de fuentes y deconvolución de canal.

Separación de fuentes: mezcla lineal instantánea [2, 3, 4] y mezcla post no-lineal [5].

Deconvolución de canal: Deconvolución Lineal e Inversión de sistemas de Wiener no lineales [6, 7].

A través de las diferentes pantallas de simulación el usuario puede modificar algunos de los parámetros de los algoritmos. También se puede observar la evolución de los resultados durante la simulación, ayudando así a la comprensión de su funcionamiento. Hemos creado, así mismo, un asistente que nos servirá de ayuda para la configuración de los diferentes algoritmos.

Previo al entorno de simulación, el usuario podrá navegar por unas páginas web en las cuales se presenta, de forma resumida, el fundamento teórico de cada uno de los algoritmos, y sus referencias bibliográficas.

3.1 Separación de fuentes

Caso lineal:

- Fuentes: Selección de las fuentes originales y de su frecuencia.
- Matriz de mezcla: Selección de la matriz, aleatoriamente o de forma manual.
- Ganancia de adaptación: Modificación del paso de adaptación.
- Épocas: Control del número de épocas que se ejecuta el algoritmo (el algoritmo trabaja con un bloque de datos de forma iterativa).
- Algoritmo: A través de esta opción podemos escoger el tipo de algoritmo de separación que deseamos. Así pues se puede elegir el algoritmo de minimización de la información mutua (MIM), el algoritmo Héroult-Juttén (H-J) o bien H-J modificado, en el cual la ganancia de adaptación depende de la correlación cruzada de las señales en la salida.

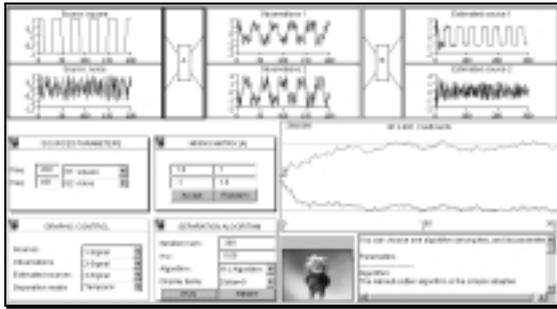


Figura 1: Mezcla lineal instantánea.

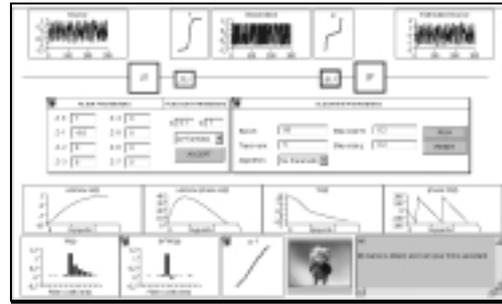


Figura 2: Inversión de Sistemas de Wiener

- Visualización: El usuario puede cambiar la visualización de las señales. Puede visualizarlas en forma temporal o como la composición de las dos señales.

Caso post no-lineal:

- Funciones no lineales: Nos modelarán comportamientos no lineales. Se dispone de dos tipos básicos de funciones no lineales: una función saturante $\tanh(\cdot)$ y una distorsión cúbica $(\cdot)^3$. Mediante dos parámetros podemos modificar el nivel de distorsión de estas funciones.

3.2 Deconvolución de canal

Caso lineal:

- Coeficientes del filtro: En el caso de la deconvolución, el usuario puede modificar los ocho coeficientes que conforman el filtro 'desconocido'.
- Filtro estimado: También se puede elegir el número de coeficientes del filtro que aproximamos para la deconvolución (*Taps*).
- Épocas: Control del número de épocas que se ejecutará el algoritmo.

Caso no lineal:

Aparte de las opciones anteriormente citadas en el 'caso lineal' la simulación no lineal añade las siguientes opciones.

- Funciones no lineales: Se dispone, como en el caso de separación post no-lineal, de dos tipos de funciones comentadas anteriormente: $\tanh(\cdot)$ y $(\cdot)^3$.
- Algoritmo: Podremos elegir el tipo de algoritmo que queremos utilizar cuando buscamos la función inversa $g(\cdot)$. El método paramétrico usa un polinomio para caracterizar la función, mientras que el no paramétrico no tiene ninguna función que lo caracterice. Este último nos permite invertir cualquier función biyectiva.

5. CONCLUSIONES

Se ha presentado una herramienta didáctica para el estudio y evaluación de diferentes algoritmos de separación de fuentes y deconvolución de canal.

Todos estos algoritmos están íntegramente realizados en Java, lo cual nos permite poder ejecutarse desde cualquier lugar del mundo con conexión a Internet, y sin importancia de la plataforma usada.

Los resultados obtenidos en las diferentes simulaciones permiten entender fácilmente en qué consiste cada una de las simulaciones, y gracias a la facilidad de uso, con ayuda del asistente, creemos que va a ser una herramienta muy útil para la difusión de las técnicas de separación de fuentes a diferentes aplicaciones dentro del procesamiento de señal.

AGRADECIMIENTOS

Este trabajo no hubiera sido posible sin la inestimable ayuda de Christian Jutten, director del Laboratoire des Images et des Signaux (LIS, INPG), lugar donde se realizó el proyecto, y Jordi Solé, profesor de la Universitat de Vic y director de este proyecto.

6. REFERENCIAS

- [1] D. Jiménez, "Algorithms interactifs de Separation Sources en JAVA". TFC Universitat de Vic, Marzo 2000.
- [2] J. Héroult, C. Jutten. "Blind separation of sources, Part I : An adaptive algorithm based on neuromimetic architecture". *Signal Processing*, Vol. 24, pp. 1-10, 1991
- [3] J. Solé, E. Monte. "Uso de redes neuronales para la separación de señales de música". URSI 97, Bilbao (España), septiembre 1997
- [4] D. T. Pham, P. Garat, C. Jutten. "Separation of mixtures of independent sources through a maximum likelihood approach" EUSIPCO 92, Bruxelles (Belgium), september 1992
- [5] A. Taleb, C. Jutten "Source Separation in Post Non Linear Mixtures" *IEEE Transactions on Signal Processing*, Vol. 47, n° 10, pp. 2807-20, October 1999
- [6] J. Solé, A. Taleb, C. Jutten. "Parametric Approach to Blind Deconvolution of Non-linear Channels". ESANN 2000, Bruges (Belgium), abril 2000.
- [7] A. Taleb, J. Solé, C. Jutten. "Quasi-Nonparametric Blind Inversion of Wiener Systems". Submitted to *IEEE Transactions on Signal Processing*, 1999.