

# ALGORITMO ADAPTATIVO CON EL GRADO DE MOVIMIENTO PARA EL DESENTRELAZADO DE VÍDEO

P. Brox<sup>1</sup>, I. Baturone<sup>1</sup>, S. Sánchez-Solano<sup>1</sup>, J. Gutiérrez-Ríos<sup>2</sup> and F. Fernández-Hernández<sup>2</sup>

<sup>1</sup> Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC)

Avda. Reina Mercedes S/N. Edificio CICA. 41012 Sevilla

e-mail:brox@imse.cnm.es

<sup>2</sup> Dpto. Tecnología Fotónica. Facultad de Informática de la Universidad Politécnica de Madrid

Campus de Montegancedo S/N. 28660 Boadilla del Monte (Madrid)

e-mail:jgr@fi.upm.es

## Resumen

En esta comunicación se presenta un algoritmo adaptativo con el movimiento para el desentrelazado de vídeo. Se basa en un sistema de inferencia difuso, que realiza una interpolación entre dos técnicas lineales en función del grado de movimiento. Se ha realizado un estudio de diferentes sistemas difusos con distinto número de funciones de pertenencia, analizándose el grado de complejidad de los mismos frente a su eficacia desentrelazando varias secuencias de vídeo.

**Palabras Clave:** Desentrelazado de vídeo, Movimiento adaptativo, Sistemas de inferencia difusos, Técnicas de aprendizaje supervisado.

## 1 INTRODUCCIÓN

Los principales formatos de transmisión de señales de televisión (NTCS, PAL, SECAM) utilizan una señal de vídeo entrelazada, donde sólo se transmiten alternativamente las líneas pares e impares de cada fotograma. De este modo, el ancho de banda de la transmisión se reduce a la mitad de una forma muy efectiva ya que, debido a las características del sistema de visión humano, el parpadeo provocado por la eliminación de líneas es prácticamente inapreciable [3]. No obstante, el auge de dispositivos que requieren un barrido progresivo de la señal de vídeo (televisores de alta definición, DVDs, proyectores, etc.) ha fomentado el desarrollo de algoritmos de desentrelazado que realizan algún tipo de interpolación espacio-temporal para calcular las líneas no transmitidas.

Entre los algoritmos de desentrelazado pueden distinguirse globalmente aquellos que utilizan un vector re-

presentativo del movimiento de la imagen para interpolar las líneas ausentes y los que no lo hacen [4]. Los primeros realizan una interpolación más precisa a costa de un elevado coste computacional requerido para calcular dicho vector. Los diferentes algoritmos pueden clasificarse atendiendo a si interpolan siempre los mismos píxeles (técnicas lineales) [6] [11], o si la interpolación se adapta a las características de la imagen (técnicas no-lineales) [1] [5]. Entre los algoritmos adaptativos se distinguen a su vez dos grupos: aquellos que tratan de adaptar la interpolación a la presencia de bordes en la imagen [5]; y aquellos otros que evalúan la cantidad de movimiento en la imagen adaptando la interpolación a ésta [1].

Cuando se emplean técnicas de movimiento adaptativo es fundamental realizar una buena estimación del grado de movimiento. Básicamente los detectores de movimiento evalúan la diferencia de valores de luminancia entre píxeles de campos consecutivos. No obstante, esta medida no siempre es fiable debido a la presencia de bordes o detalles con grandes contrastes de valores de luminancia en la dirección vertical de la imagen, y a que la señal puede contener ruido. Para aumentar la robustez de los detectores de movimiento algunos autores proponen conectar varios en cascada, de modo que solo si todos ellos detectan movimiento la señal se active [3]. También se han propuesto distintos algoritmos basados en lógica difusa para realizar un desentrelazado adaptativo con el grado de movimiento obteniendo mejoras significativas. Esto se debe a la capacidad de las técnicas difusas para realizar interpolaciones en zonas donde la información es imprecisa y, por tanto, la decisión no es trivial [10]. La técnica propuesta en [10] obtiene buenos resultados pero emplea una base de reglas compleja que requiere un coste computacional considerable.

En esta comunicación se propone un nuevo algoritmo adaptativo para el desentrelazado de vídeo que emplea un sistema de inferencia difuso para determinar, en función del movimiento, la interpolación entre los

píxeles de las líneas transmitidas. El algoritmo es descrito en detalle en la sección 2. Su validez es analizada en la sección 3 desentrelazando varias secuencias de imágenes. Finalmente, las conclusiones del trabajo son resumidas en la sección 4.

## 2 DESCRIPCIÓN DEL ALGORITMO

El algoritmo de desentrelazado adaptativo en función del movimiento que presentamos se basa en la siguiente heurística: si el píxel a calcular corresponde a un área donde no existe movimiento, el resultado mejor se obtiene realizando una interpolación entre píxeles del campo anterior (interpolación temporal). Si por el contrario el píxel corresponde a un área donde el grado de movimiento es elevado, lo más adecuado es realizar una interpolación entre distintos píxeles del campo actual (interpolación espacial). Entre las técnicas lineales temporales y espaciales se han elegido las más básicas: la técnica de inserción del píxel del campo anterior como temporal ( $I_T$ ) y la media aritmética de los píxeles de las líneas superior e inferior como espacial ( $I_S$ ). El grado de movimiento es evaluado procesando la señal que se obtiene al realizar la convolución bi-dimensional de la diferencia de luminancia en valor absoluto de dos campos que contienen líneas de la misma paridad. Matemáticamente puede expresarse como:

$$mov(x, y, t) = \sum_{i=1}^3 (\sum_{j=1}^3 H_{ij} C_{ij}) \quad (1)$$

donde  $H_{ij}$ ,  $C_{ij}$ , vienen dadas por las siguientes matrices:

$$C = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (2)$$

$$H^{xyt} = \begin{pmatrix} H^{x-1y-1t-1} & H^{x-1yt} & H^{x-1y+1t-1} \\ H^{xy-1t-1} & H^{xyt} & H^{xy+1t-1} \\ H^{x+1y-1t-1} & H^{x+1yt} & H^{x+1y+1t-1} \end{pmatrix} \quad (3)$$

siendo  $H^{xyt}$  la diferencia en valor absoluto de la luminancia de los píxeles que pertenecen a dos campos que contienen líneas de la misma paridad:

$$H^{xyt} = H(x, y, t) = \frac{|I(x, y, t-1) - I(x, y, t+1)|}{2} \quad (4)$$

La notación (x,y,t) significa que el píxel tiene una ubicación espacial determinada por las coordenadas (x,y)

y corresponde a un campo determinado (t) de la secuencia de video. Atendiendo al tamaño de las matrices H y C, se observa que se emplea una ventana de convolución bi-dimensional de tamaño 3x3. La idea de utilizar técnicas de convolución para evaluar el movimiento fue introducida en [7]. La principal ventaja de éstas es que permiten tener en cuenta la contribución de los vecinos espacio-temporales al estimar el movimiento en el píxel actual. De este modo, puede minimizarse la influencia de los eventuales errores en la detección de movimiento debido a la presencia de ruido, bordes o detalles con contrastes de luminancia elevados. Además es posible asignar un peso ponderando cada uno de los píxeles vecinos mediante los coeficientes de la matriz C, tal y como indica la expresión (2). Se ha realizado un estudio para evaluar distintas posibilidades de la ventana de convolución analizando distintas dimensiones y valores de los coeficientes que la componen [2]. Como conclusión se ha seleccionado la indicada en la expresión (2).

Las técnicas de movimiento adaptativo fueron originalmente introducidas en [1]. El grado de movimiento se evaluaba comparando el valor de la señal correspondiente a la diferencia de luminancia entre campos consecutivos con un valor umbral constante. El objetivo del trabajo descrito en esta comunicación ha sido emplear una técnica de movimiento adaptativo que emplea un sistema difuso para realizar la transición entre las dos técnicas de interpolación ( $I_S$ ,  $I_T$ ) de manera más suave. De este modo, en las zonas donde el grado de movimiento es medio y por tanto, la decisión no es trivial, se realiza una interpolación no-lineal entre  $I_S$  e  $I_T$ .

### 2.1 DESCRIPCIÓN DEL SISTEMA DE INFERENCIA DIFUSO

El conocimiento heurístico empleado por las técnicas de movimiento adaptativo es modelado mediante un sistema de inferencia difuso. En primer lugar, se ha empleado un sistema con solo dos reglas, donde los conceptos SMALL y LARGE se representan mediante los conjuntos difusos de la Figura 1(a). No obstante, podemos aprovechar la capacidad de interpolación de la lógica difusa considerando la posibilidad de ampliar el número de conjuntos difusos. De este modo, sería posible contemplar un nuevo conjunto difuso (representado con la etiqueta MEDIUM en la Figura 1(b)). La base de reglas se amplía considerando una nueva regla que, en el caso de activarse, implementa una combinación lineal de las técnicas  $I_S$  y  $I_T$ .

Este razonamiento puede extenderse aumentando el número de conjuntos difusos considerados a los cuatro (SMALL, SMALL-MEDIUM, MEDIUM-LARGE, LARGE) o cinco (SMALL, SMALL-MEDIUM, ME-

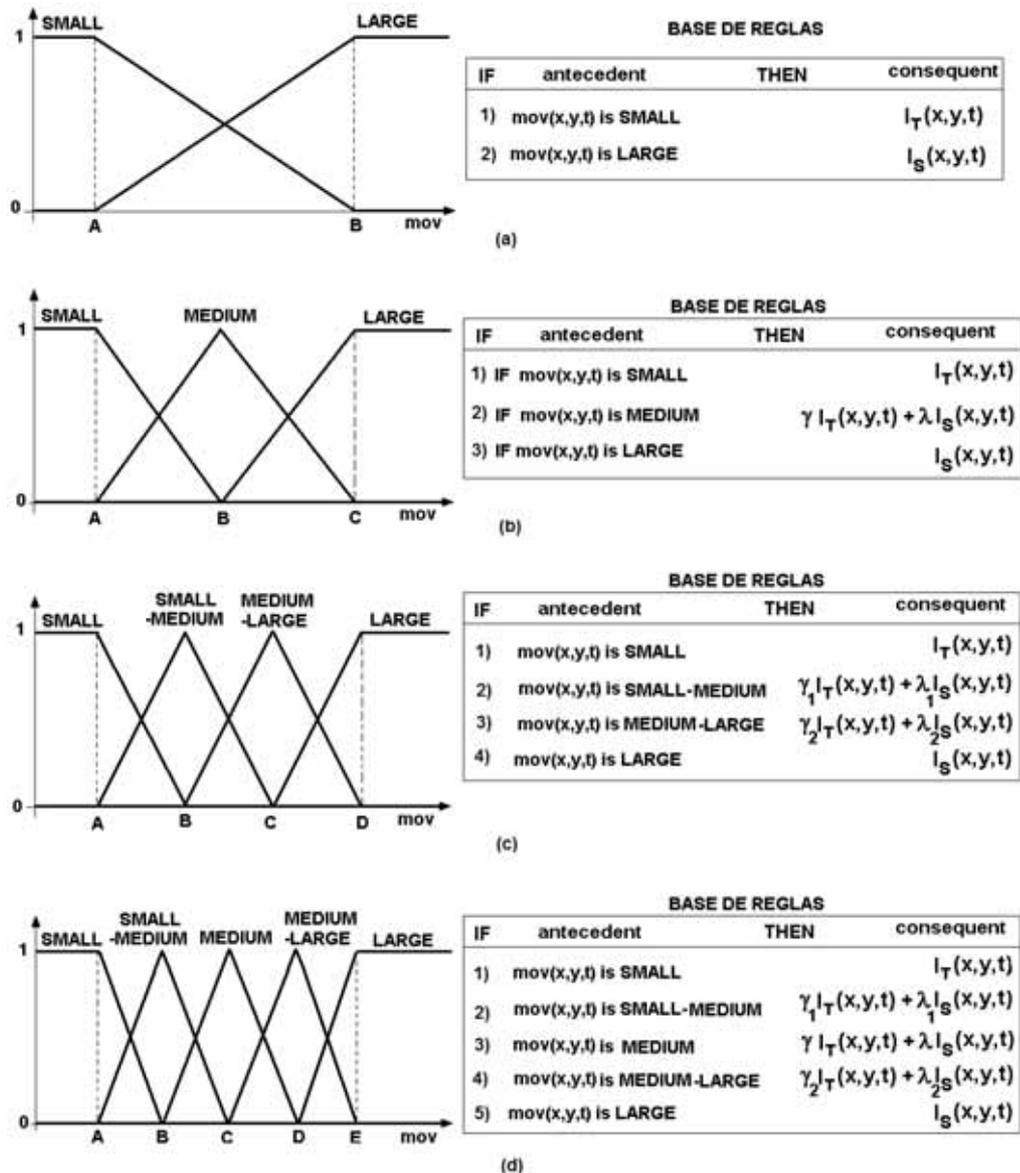


Figura 1: Funciones de pertenencia utilizadas por los distintos sistemas de inferencia

DIUM, MEDIUM-LARGE, LARGE) representados en las Figura 1(c) y 1(d) respectivamente. El número de reglas de la base de reglas aumenta del mismo modo. El problema de esta técnica es que "a priori" no existe ninguna indicación para fijar las constantes de los consecuentes de las bases de reglas con más de dos reglas, ni tampoco para determinar las constantes (A, B, C, D, E) que definen los conjuntos difusos asociados a las distintas etiquetas lingüísticas. Para fijar estos valores podemos entrenar los sistemas difusos empleando técnicas de aprendizaje supervisado. El apartado siguiente describe en detalle dicho proceso.

## 2.2 PROCESO DE APRENDIZAJE SUPERVISADO

Los sistemas han sido implementados en el entorno de desarrollo de sistemas difusos Xfuzzy [8]. Este entorno facilita el diseño de sistemas de inferencia basados en lógica difusa al incluir distintas herramientas de CAD que cubren las etapas de descripción, identificación, simplificación, verificación, ajuste automático y síntesis.

La etapa de ajuste constituye habitualmente una de las tareas más complejas del diseño de sistemas difusos. La herramienta que se encarga de implementar esta etapa en Xfuzzy se denomina *xfit* [9]. Esta herramienta permite aplicar algoritmos de aprendizaje

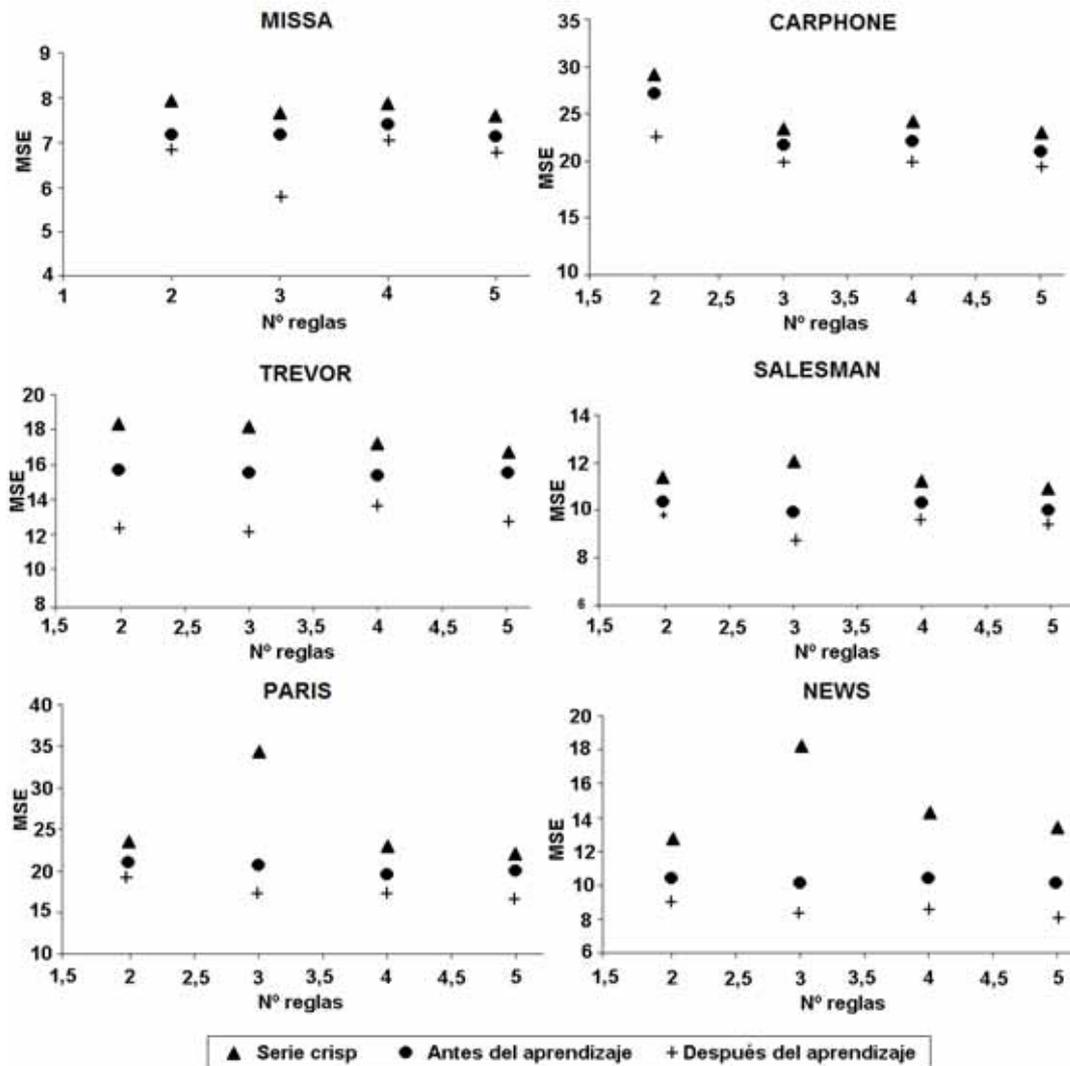


Figura 2: Valores de MSE obtenidos por los distintos sistemas de inferencia difusos al desentrelazar las secuencias de vídeo

supervisado donde el comportamiento deseado del sistema es descrito mediante un conjunto de patrones de entrenamiento. Los sistemas han sido entrenados utilizando como patrones de entrenamiento un conjunto de fotogramas de vídeo progresivo. De este modo, el algoritmo de aprendizaje supervisado seleccionado (Marquardt-Levenberg en nuestro caso) intenta minimizar una función de error que evalúa la diferencia entre el comportamiento actual y el deseado (determinado por los patrones de entrada/salida). La herramienta *xfl* permite aplicar el proceso de ajuste a los distintos parámetros de los sistemas de inferencia difusos implementados. La utilidad de esta etapa del proceso de diseño se ha verificado desentrelazando varias secuencias de vídeo, y se explica en detalle en la sección 3.

### 3 RESULTADOS DE SIMULACIÓN

El algoritmo propuesto ha sido probado simulando distintas secuencias de vídeo estándares ampliamente utilizadas por la comunidad científica y accesibles a través de la página web: <http://decsai.ugr.es>. Las secuencias utilizadas se encuentran originalmente en un formato de vídeo progresivo por lo han sido desentrelazadas artificialmente, es decir, eliminado líneas de cada uno de los fotogramas que las componen. Los datos del fichero de entrenamiento se obtiene seleccionando un conjunto de imágenes progresivas de cada una de las secuencias.

La Figura 2 muestra el error cuadrático medio (MSE) obtenido al desentrelazar seis secuencias de vídeo. Este valor corresponde al valor medio de las imágenes desentrelazadas (aproximadamente se han simulado unas

Tabla 1: Valor medio de PSNR (en dBs) obtenido al desentrelazar distintas secuencia con diferentes algoritmos.

<b>Secuencia</b>	<b>Missa</b>	<b>Salesman</b>	<b>Carphone</b>	<b>Paris</b>	<b>Trevor</b>	<b>News</b>
<b>Formato</b>	<b>CIF</b>	<b>CIF</b>	<b>QCIF</b>	<b>CIF</b>	<b>CIF</b>	<b>QCIF</b>
<b>RL</b>	36.44	29.75	28.25	23.61	31.05	25.18
<b><math>I_S</math></b>	40.47	33.53	32.61	26.67	35.04	29.25
<b><math>I_T</math></b>	38.36	36.17	30.34	29.86	34.36	33.13
<b>VT-2fields</b>	40.25	36.54	34.08	30.73	36.61	35.46
<b>VT-3fields</b>	40.52	36.95	34.54	31.37	37.16	35.67
<b>Técnica [10]</b>	40.01	37.62	32.27	33.12	35.38	34.73
<b>Propuesta 2 reglas</b>	40.18	38.29	34.78	35.28	36.69	37.51
<b>Propuesta 3 reglas</b>	40.51	38.44	34.83	35.78	37.49	38.68
<b>Propuesta 4 reglas</b>	39.65	38.23	34.94	35.55	36.77	38.65
<b>Propuesta 5 reglas</b>	39.67	38.21	34.94	35.93	37.16	39.15

50 imágenes de cada secuencia). Las gráficas de la Figura 2 muestran los resultados obtenidos al implementar un algoritmo donde los conceptos, SMALL, SMALL-MEDIUM, MEDIUM, MEDIUM-LARGE y LARGE están definidos mediante valores umbrales, es decir, determinados por un valor numérico constante. También se muestran los resultados obtenidos mediante la implementación de los sistemas difusos con distinto número de reglas (con y sin aprendizaje). Comparando las tres series de resultados puede deducirse que los algoritmos que implementan los sistemas difusos obtiene los errores más pequeños, reduciéndose aún más estos valores si las funciones de pertenencia y los consecuentes se modifican mediante el proceso de aprendizaje. Finalmente, analizando el número de reglas empleadas y el valor de MSE obtenido se deduce que si se utilizan tres reglas se obtienen mejores resultados que con dos. No obstante, las mejoras introducidas con cuatro y cinco funciones de pertenencia no son significativas con respecto a la propuesta que utiliza tres. Es más, en determinados casos incluso dan lugar a errores ligeramente superiores.

El algoritmo propuesto también ha sido comparado con otras técnicas de desentrelazado. La Tabla 1 muestra el valor medio en PSNR obtenido al desentrelazar distintas secuencias de vídeo aplicando una serie de algoritmos. Concretamente se han analizado las técnicas lineales más simples: duplicación o repetición del píxel de la línea anterior (RL) y el valor medio de las líneas superior e inferior ( $I_S$ ) como técnicas espaciales y la inserción del píxel del campo anterior ( $I_T$ ) como temporal. También se han considerado en el estudio técnicas lineales espacio-temporales actualmente utilizadas en chips comerciales [5], [10]. Finalmente, hemos considerado una técnica de movimiento adaptativo que también emplea un sistema difuso para realizar la interpolación [9]. Analizando los resultados mostrados en la Tabla 1 se observa que los resultados más altos de

PSNR y por tanto, los errores más bajos corresponden al algoritmo propuesto (se indican los valores obtenidos con las distintas funciones de pertenencia tras realizarse el proceso de aprendizaje). Esto también puede ser corroborado analizando las imágenes desentrelazadas de la Figura 3.

Finalmente, se ha realizado un análisis del coste computacional involucrado en la implementación de cada uno de los algoritmos. Para ello todos los algoritmos han sido ejecutados en la misma plataforma (un PC con procesador Pentium IV y sistema operativo MSWindow XP) determinándose el tiempo empleado por cada uno de ellos en procesar una misma secuencia. Los resultados se muestran en la Tabla 2. Puede comprobarse cómo las técnicas lineales son la más rápidas aunque los resultados obtenidos por ellas se ven ampliamente mejorados por nuestra propuesta.

## 4 CONCLUSIONES

En esta comunicación se ha presentado un sistema difuso que en función del grado de movimiento implementa distintas combinaciones entre dos técnicas lineales. Está basado en las técnicas clásicas de movimiento adaptativo pero utiliza deficiones difusas en lugar de crisp para determinar el grado de movimiento. Se han implementado distintos sistemas difusos con distinto grado de complejidad analizando la eficacia de cada uno de ellos para realizar la interpolación. Los parámetros que definen el sistema de inferencia difuso han sido determinados mediante un proceso de ajuste automático implementando un proceso de aprendizaje supervisado. En función de los resultados obtenidos se deduce que un sistema que evalúa el grado de movimiento con tres funciones de pertenencia es eficiente tanto por los resultados que consigue como por su coste computacional.

Tabla 2: Tiempo de ejecución para desentrelazar una de las secuencias.

Algoritmo	RL	$I_S$	$I_T$	VT 2fields	VT 3fields	Técnica [10]	Propuesta 2-3-4-5 reglas
Tiempo(s)	2.03	2.05	3.28	10.62	14.65	143.03	29.23-30.95-31.76-32.65



Figura 3: Imágenes desentrelazadas obtenidas aplicando: (a) RL, (b)  $I_S$ , (c)  $I_T$ , (d) VT2fields, (e) VT3fields, (f) técnica [10], (g) propuesta de 2 y (h) 3 reglas

### Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos TEC2005-04359/MIC del Ministerio español de Educación y Ciencia, y TIC2006-635 de la Junta de Andalucía. El primer autor forma parte del programa de formación para estudiantes de doctorado F.P.U., del Ministerio español de Educación y Ciencia.

### Referencias

[1] A. M. Bock. Motion adaptive standards conversion between formats of similar field rates. *Signal*

*Processing: Image Communication*, Vol. 6, no. 3, Pág.275-280, 1994.

- [2] P. Brox, I. Baturone, S. Sánchez-Solano. A Fuzzy Motion Adaptive Algorithm for Interlaced-to-Progressive Conversion. *It will be published in Proc. of the Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'2006)*, 2006.
- [3] G. De Haan. Video Processing. University Press, Eindhoven, 2004.
- [4] G. De Haan and E. B. Bellers. De-interlacing: An overview. *Proc. of the IEEE*, Vol. 86, Pág.1839-1857, 1998.
- [5] T. Doyle and M. Looymans. Progressive scan conversion using edge information. *Signal Processing of HDTV*. Ed. Elsevier Science Publishers, Vol. II, Pág.711-721, 1990.
- [6] Genesis Microchip, Inc., Preliminary data sheet of Genesis gmVLD8, 8 bit digital video line doubler, versión 1, 1996.
- [7] J. Gutiérrez-Ríos, F. Fernández-Hernández, J. C. Crespo and G. Triviño. Motion adaptive fuzzy video de-interlacing method based on convolution techniques. *Proc. of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2004.
- [8] F. J. Moreno-Velo, I. Baturone, S. Sánchez-Solano and A. Barriga. Rapid design of complex fuzzy systems with XFUZZY. *Proc. IEEE Int. Conf. on Fuzzy Systems*, Págs.342-347, 2003.
- [9] F. J. Moreno-Velo, I. Baturone, R. Senhadji y S. Sánchez-Solano. Tuning complex fuzzy systems by supervised learning algorithms. *Proc. IEEE Int. Conf. on Fuzzy Systems*, Págs. 226-231, 2003.
- [10] D. Van de Ville, B. Rogge, W. Philips and I. Lemahieu. De-interlacing using fuzzy-based motion detection. *Proc. 3rd Int. Conf. on Knowledge-Based Intelligent Information Engineering Systems*, Pág.263-267, 1999.
- [11] M. Weston. Interpolating lines of video signals. US-patent 4, Pág.789-893, 1998.