

RAZONAMIENTO BASADO EN CASOS

Xavier P. Burgos-Artizzu, Jesús Conesa y Ángela Ribeiro

A11.1 INTRODUCCIÓN

A continuación se describe el ejemplo de aplicación que sirve para mostrar el funcionamiento del razonamiento basado en casos sobre imágenes descrito en el capítulo once, sección 3.1.

La aplicación, escrita en el lenguaje MATLAB <http://www.mathworks.com/>, se distribuye organizada en tres carpetas.

- 1) La carpeta “Imágenes”, contiene 23 imágenes ejemplo, organizadas en dos grupos.
 - 17 imágenes pertenecientes a los Casos ya presentes en la Base de Casos inicial, etiquetadas como CasoX, donde $X = [1 \dots 17]$.
 - 6 imágenes de prueba, que pueden ser usadas como nuevos casos de entrada para probar el funcionamiento del CBR, etiquetadas como PruebaX, siendo $X = 1, 2, 3, 4, 5, 6$.
- 2) La carpeta “CBR” que contiene tanto la Base de Casos (archivo *Bc.mat*) como todas las funciones de las que se compone el CBR.

- 3) La carpeta “Procesamiento”, que contiene todas las funciones auxiliares para el procesamiento de las imágenes.

A11.2 MANUAL DE USUARIO

A11.2.1 Función Principal (*procesaImagenCBR*)

La aplicación está escrita en el lenguaje MATLAB y su función principal es *procesaImagenCBR*. Esta función ejecuta el CBR completo sobre una imagen de entrada, pasada como parámetro, y devuelve la imagen del caso recuperado, el procesamiento usado y el valor final de cobertura de malas hierbas de la imagen de entrada.

PARAMETROS DE ENTRADA

nombre_imagen: String con ubicación en disco de la imagen a procesar

PARAMETROS DE SALIDA

imagen_recuperada: Identificador de la imagen recuperada
procesamiento_usado: Métodos y valor de los parámetros usados para el procesamiento de la imagen
mh: Cobertura estimada de malas hierbas en la imagen

USO

```
[imagen_recuperada, procesamiento_usado, mh] =  
    procesaImagenCBR(ruta_imagen)
```

Por ejemplo, si se desea procesar la imagen de prueba n° 2, basta con establecer como directorio de trabajo la carpeta CBR y escribir en la ventana de comandos de *Matlab*:

```
[imagen_recuperada, procesamiento_usado, mh] =  
    procesaImagenCBR ( '..\Imágenes/Prueba2.jpg' )
```

Tras unos segundos, la aplicación finalizará devolviendo los siguientes resultados:

```
imagen_recuperada =    5;
```

El valor de *imagen_recuperada* nos indica que se ha recuperado como Caso previo más similar el Caso nº 5.

```
procesamiento_usado = [1.0000    2.0000    2.0000
-0.7000    0.5880    0.1360   247.0000   39.0000   10.0000
15.0000];
```

El valor de *procesamiento_usado* nos muestra el procesamiento recuperado, en forma de array. Las tres primeras posiciones del array nos indican qué métodos se han usado en cada etapa (en este caso S1, E2, F2). Las siguientes posiciones muestran el valor usado para los parámetros de entrada de cada método, en orden secuencial.

Así, las primeras posiciones mostrarán el valor de los parámetros del método de segmentación (en este caso cuatro ya que se trata de S1: -0.7, 0588, 0.136, 247), las siguientes los del método de eliminación (tres al tratarse de E2: 39, 10) y por último los del método de filtrado (uno al tratarse de F2: 15).

```
mh = 8.1979;
```

El valor de *mh* indica el valor estimado de cobertura de malas hierbas sobre la imagen, en este caso del 8,2%.

La aplicación muestra también el resultado obtenido por pantalla mediante una ventana, en la que además de los valores anteriores también se puede ver el resultado del procesamiento sobre la imagen de entrada, figura A11.1.

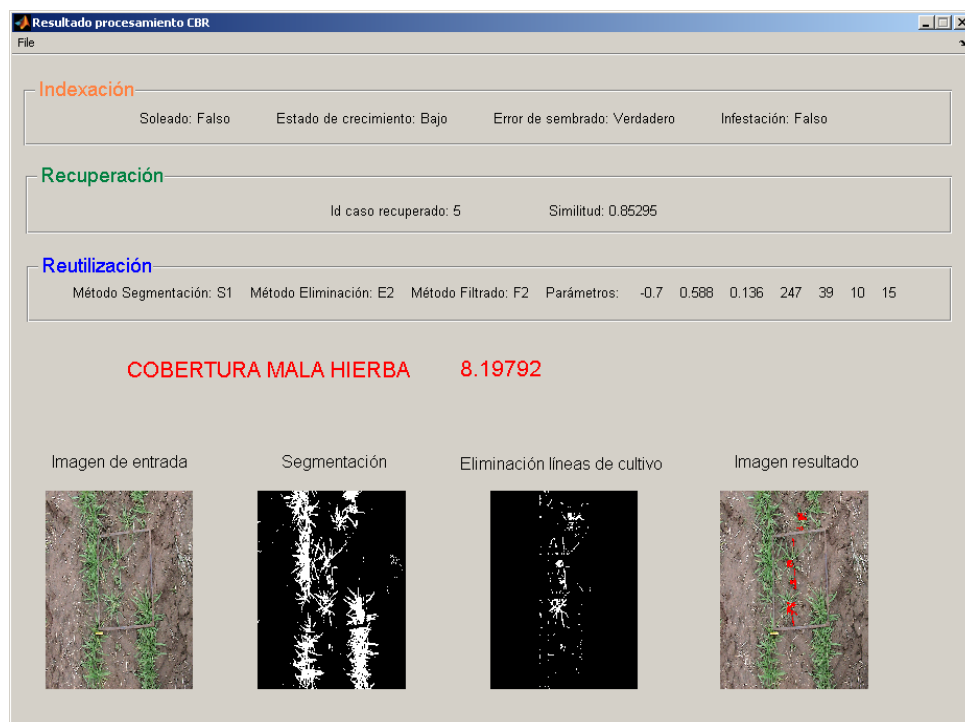


Figura A11.1. Resultados de la aplicación tras un ciclo CBR completo

A11.2.2 Funciones Auxiliares

Además de la función principal, pueden también usarse las funciones intermedias, usadas por *procesaImagenCBR* para efectuar cada una de las etapas del ciclo CBR.

Como paso previo a la ejecución de todas estas funciones, será necesaria la carga de la imagen a procesar al espacio de trabajo de *matlab* como variable. Esto se puede llevar a cabo del siguiente modo:

```
im = imread(entrada);
im = im2single(im);
im = imresize(im,[1024 768]);
```

Donde *entrada* es un “String” con la ubicación de la imagen en el disco, por ejemplo ‘../Imagenes/Prueba2.jpg’.

A11.2.2.1 INDEXACIÓN

La función principal de indexación es *indexa*. Esta función devuelve, dada la imagen de entrada *im*, su clase de pertenencia mostrando el valor de los cuatro atributos característicos (Soleado, Estado de Crecimiento del Cultivo, Presencia de errores de sembrado, Plagado). Su uso es muy sencillo:

```
[soleado, estadoCrec, errorSembrado, plagado] = indexa(im);
```

A su vez, *indexa* utiliza cuatro funciones auxiliares para calcular cada uno de los cuatro atributos, que si se desea también pueden ser usadas de forma independiente de modo muy sencillo:

```
soleado = fsoleado(im);
plagado = fplagado(im);
estadoCrec = festadoCrec(im);
errorSembrado = ferrorSembrado(im, estadoCrec);
```

A11.2.2.2 RECUPERACIÓN

La función principal de recuperación es *recupera.m*. Esta función devuelve el procesamiento a utilizar, la imagen recuperada y el valor de similitud con dicha imagen, dada la imagen de entrada *im* y su clase de pertenencia mediante los cuatro atributos característicos *soleado*, *estadoCrec*, *errorSembrado*, *plagado*, calculados en la etapa de indexación.

El procesamiento recuperado se devuelve en forma de cuatro valores, el método usado en cada etapa (*segm*, *elim.*, *filt*) y el array de parámetros de entrada a cada método *param*. Los valores de *imagen_recuperada* y *sim* indican respectivamente el número de Caso recuperado y el valor de similitud entre la imagen de entrada y la imagen perteneciente a dicho Caso.

Por lo tanto, el uso de *recupera* queda como sigue:

```
[segm, elim, filt, param, imagen_recuperada, sim] = recupera(im,
soleado, estadoCrec, errorSembrado, plagado);
```

A su vez, para calcular la similitud entre dos imágenes, *recupera* utiliza la función *similitud*, que calcula la similitud entre dos imágenes cargadas en el entorno de trabajo de *Matlab* *im1* y *im2*, según el método S2.

```
sim = similitud(im1,im2);
```

A11.2.2.3 PROCESAMIENTO IMÁGENES

Por último, en la etapa de reutilización se procesa la imagen de entrada siguiendo el procesamiento recuperado, utilizando la plataforma de procesamiento de imágenes.

Ésta puede ser usada fácilmente, indicando la imagen a procesar cargada al espacio de trabajo *im*, y el procesamiento a utilizar tal como se devuelve en la etapa de recuperación, con los cuatro valores *segm*, *elim.*, *filt* y el array *param*.

La plataforma devuelve, además de la estimación de cobertura de mala hierba *mh*, todas las imágenes resultado de cada una de las etapas intermedias del procesamiento: *ims* (imagen segmentada), *ime* (imagen con líneas de cultivo eliminadas) y, *imr* (imagen resultado final).

```
[ims,ime,imr,mh] = plataforma(im,segm,elim,filt,param);
```

Si en cualquier momento se desea visualizar cualquiera de estas imágenes bastará utilizar la función *imshow*:

```
imshow(im)
```