

Contribuciones al Procesado Simbólico de Segmentos en Visión por Computador*

Jorge Cabrera-Gómez[†]
e-mail: jcabrera@dis.ulpgc.es

Antonio Falcón Martel[†]
e-mail: afalcon@dis.ulpgc.es

Francisco M. Hernández Tejera[†]
e-mail: fhernandez@dis.ulpgc.es

Juan Méndez Rodríguez[†]
e-mail: jmendez@dis.ulpgc.es

Resumen

Se presenta una metodología de proceso a nivel de segmentos en el contexto de los sistemas de visión basados en conocimiento. Se muestra la utilización de conceptos de Lógica Difusa en las operaciones que tienen lugar en dicho procesador y para establecer el paso entre los dominios de representación numérico y simbólico.

Palabras clave: Sistemas de Visión, Sistemas Basados en Conocimiento, Segmentación de Imágenes.

1 Introducción.

La Visión por Computador está dedicada al estudio de teorías y al desarrollo de métodos y algoritmos en sistemas de percepción visual artificial. En los últimos años se ha producido un cambio sustantivo en las metodologías de diseño y desarrollo de tales sistemas, por un lado, debido a la integración de tecnologías y métodos de la Ingeniería del Conocimiento y, por otro lado, en la consideración básica de que un sistema de visión no es un elemento pasivo o aislado que recibe información y la procesa en algún sentido, sino más bien que forma parte indisoluble con un entorno concreto y que, para su estudio, es necesario la consideración del sistema interactuando con el medio mediante un proceso de percepción-acción [1], [2]. Además, la evolución se manifiesta en los planteamientos de resolución del compromiso de la flexibilidad o adaptabilidad de los sistemas desarrollados respecto a las modificaciones del entorno frente a la calidad o robustez de los resultados. Por otra parte, así como en muchas áreas de la Ciencia y de la Tecnología se tratan problemas para los que se dispone de modelos útiles y válidos que proporcionan resultados concretos, en Visión por Computador los problemas carecen de modelos plenamente válidos, salvo en circunstancias muy concretas. La resolución de tales problemas requiere en muchos casos de la utilización de procedimientos con múltiples heurísticas y la integra-

ción de resultados de hipótesis de validez parcial con distintos grados y tipos de incertidumbres. Dicha integración puede hacerse de manera adecuada utilizando métodos de Inteligencia Artificial y Conjuntos Fuzzy para generar conclusiones y resultados.

Los sistemas de visión por computador involucran procesos que actúan sobre elementos de diferente tipología, esto es, a bajo nivel o pixels, a nivel intermedio o segmentos y a alto nivel u objetos. En cada uno de los niveles se definen categorías simbólicas de objetos y en ellas aparecen, en general incertidumbre. Particularmente, la segmentación de imágenes puede considerarse como un proceso de simbolización, entendida en el contexto de lo que así se considera en Reconocimiento de Formas o en Visión por Ordenador [3]. En este caso, el grano de información que se maneja es el segmento o agregación de pixels, definiendo un determinado nivel de representación en intenso de mayor complejidad que el nivel de pixels pero de inferior del de objetos o estructuras relacionales. En este contexto, un segmento de una escena puede pasar de una descripción numérica a una simbólica con ciertos coeficientes de certeza y es aquí donde las técnicas de la Lógica Fuzzy permiten una aproximación práctica plausible [4]. Además en este nivel la definición de categorías visuales no es dura sino más bien de variación suave [5]. De otro lado, las particularidades de los procesos de refinamiento que se producen en una segmentación, donde las condiciones que afectan a cada segmento juegan un papel importante en el resultado final, nos hacen plantearnos una vía de estudio del proceso de evaluación de condiciones y de reglas de control dentro del marco de los Conjuntos Fuzzy. La incertidumbre se manifiesta en los diferentes niveles [6] y adquiere características específicas según sean estos (pixels, segmentos, objetos, ...) fundamentadas en las definiciones de clases y en las ambigüedades de las interpretaciones motivadas por preguntas mal planteadas desde niveles más altos. En este trabajo se expone la arquitectura que debería tener un procesador de segmentos que produzca particiones estables hacien-

* Este trabajo ha sido parcialmente financiado por la Consejería de Educación del Gobierno de Canarias (Proy. 93/069)

[†] Grupo de Inteligencia Artificial y Sistemas de la Universidad de Las Palmas de Gran Canaria

do uso de mecanismos de Ingeniería del Conocimiento y de Conjuntos Fuzzy, así como el análisis de los elementos necesarios para el control en este nivel, todo ello dentro del marco de los Sistemas de Visión Basados en Conocimiento.

2 El Nivel de los Segmentos.

La organización en niveles es una herramienta que suele utilizarse con relativa frecuencia al tratar de analizar o sintetizar sistemas complejos. Así, en el caso de análisis, el concepto de nivel responde a una forma de ordenar en jerarquías una realidad existente en hechos. Mientras en el caso de síntesis, constituye una forma de simplificar el proceso en base a construir capas de complejidad reducida. En un sentido no formal consideraremos a un nivel como un conjunto de símbolos y una serie de operadores sobre estos. En esencia, lo que define un nivel es la elección de un conjunto de símbolos que representan las entidades de su dominio y un álgebra o conjunto de formalismos u operadores que realizan las transformaciones de los símbolos.

En el nivel de los segmentos el objetivo es proporcionar la partición de la imagen en un conjunto de segmentos o agregaciones de píxeles con sus respectivos grados de pertenencia a clases de segmentos ("cuadrado_rojo", "línea_recta", ...) y esto se establecerá desde un conjunto de transformaciones sobre los segmentos y sus símbolos asociados. Consideramos al segmento como una agregación conexa de píxeles definida por criterios de forma o de uniformidad en propiedades. Para clarificar la terminología, denominaremos partición a la lista de segmentos que constituyen una teselación completa de la imagen y que incluye las asignaciones simbólicas de los segmentos a clases y las relaciones espaciales entre segmentos.

A partir de la aplicación de procedimientos de diferente naturaleza es posible hacer explícitas ciertas categorías o clases visuales relacionadas bien con los píxeles o con agregaciones de estos. En general, la definición de estas categorías no es dura, sino más bien de variación suave o gradual [7],[5] y existe ambigüedad visual, bien a nivel de la definición geométrica de clases de segmentos o de la definición de categoría tonal de los mismos, o bien a nivel de la definición de las relaciones geométricas entre segmentos. Además de la citada aplicación directa de procedimientos, los diagnósticos simbólicos sobre los segmentos pueden obtenerse por combinación lógica de diagnósticos previos, donde también se manifiesta la incertidumbre.

Es necesario hacer notar que la utilización del segmento como grano de información en este nivel plantea dos dificultades que no se presentan en el nivel de los píxeles. De una parte, la necesidad de obtener la definición espacial de los segmentos a partir de los mapas de diagnóstico o clases de píxeles obtenidos desde niveles inferiores como paso previo a su descripción simbólica en términos de clases de segmentos. De otra, en el procesamiento de segmentos es muy interesante permitir que las entidades puedan sufrir cambios durante el proceso de

refinamiento de la segmentación como consecuencia de la fusión o división de segmentos.

En este nivel, para los segmentos se consideran los siguientes elementos, que permiten el procesamiento simbólico:

a) *Propiedades* o Características, estas representan el valor numérico de una propiedad, $f_i(s)$, que se evalúa desde el segmento, s , en cuestión a través de los *Procedimientos* que son un tipo particular de *Operadores* de naturaleza numérica. Estas características definen el dominio numérico y reflejan ciertas cualidades del segmento en cuestión, bien en su aspecto morfológico (forma) o referida a propiedades intrínsecas (tonal) de los píxeles que lo componen tales como brillo, color o textura.

b) *Operadores*, que además de los Procedimientos incluyen a los elementos destinados a la obtención de clases asociadas a cada segmento desde propiedades, o desde condiciones a partir de otras clases de segmentos. Los primeros son los *Clasificadores* que transforman un dominio borroso las propiedades en diagnósticos o *Clases de Segmentos*. Existen otro tipo de operadores que están orientados a la definición de clases desde combinaciones, esto es las *Reglas*, que implican condiciones que verifican ciertas clases y cuya verificación en un dominio borroso se articula para la generación de nuevas clases.

c) *Clases de Segmentos*, que representan lo que podemos decir en un dominio simbólico acerca de la asignación de un segmento a una determinada clase o tipología de segmentos en un intervalo [0-100].

d) *Transformaciones*. Estas representan un tipo especial de operadores que afectan al dominio espacial más que al simbólico. Surgen de la necesidad de alterar la composición espacial de los segmentos al verificarse ciertas condiciones en un proceso de generación de la partición. Funcionalmente se articulan como *Reglas de Control* que, desde la verificación en un dominio borroso de ciertas condiciones, se producen acciones de control que modifican la partición actual para alcanzar otra que se acerque más a un cierto objetivo.

La estructura de clases de segmentos y su evaluación se define desde mapas de píxeles expresados en forma simbólica, esto es imágenes simbólicas generadas por procesos de simbolización sobre mapas de características. En estos mapas de píxeles el valor de cada píxel es el grado de pertenencia a una tipología o clase de píxel [3] representa la asignación de un píxel a una determinada clase de píxel. Con estos elementos, es posible realizar computación simbólica, dado que la representación se basa en clases y tipos que se pueden considerar como símbolos más que números. Como consecuencia se pueden utilizar operaciones relacionadas con jerarquías de clases y relaciones entre ellas tales como reglas o producciones. La utilización de grados de pertenencia a clases posibilita la utilización de clasificadores borrosos y coeficientes de certeza en las reglas.

Los procedimientos permiten la obtención de las características de los segmentos desde mapas de diag-

nósticos de pixels o desde otras características. En estos casos nos movemos en un dominio claramente numérico. Los clasificadores son básicamente funciones discriminantes y de decisión y aportan una mayor riqueza de significado. Utilizan características de tipo numérico para computar una de tipo simbólico en el rango[0-100]. En esencia este operador es una implementación de procedimientos de clasificación incluyendo decisiones borrosas mediante el cual se pueden implementar decisiones más complejas [8]. Sea $f_i(s)$, $i=1, \dots, n$, un conjunto de características evaluadas sobre el segmento s , $F(s)$ un funcional computado desde ellas y $C(s)$ el resultado suministrado por el clasificador. Se tiene:

$$C(s) = D_M(F(s))$$

Donde $D_M(u)$ es una función de decisión. Esta definición queda abierta a distintos tipos de funcionales (unitarios, lineales, cuadráticos ...). Así por ejemplo el funcional cuadrático será de la forma:

$$F(s) = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} [f_i(s) - a_i] b_{ij} [f_j(s) - a_j]$$

Donde $\{a_i\}$ y $\{b_{ij}\}$ son conjuntos de coeficientes numéricos. En este contexto son posibles distintos tipos de funciones de decisión, en general estos vendrán expresados en la forma $D_M(u; p_1, \dots, p_m)$, donde M representa el modelo (borroso, duro, ...). Por ejemplo, si consideramos la función tipo-S de conjuntos borrosos definida por $S(u; \alpha, \beta)$, [9], la función de decisión $D_{SS}(u; \alpha, \beta)$, queda como:

$$D_{SS}(u; \alpha, \beta) = 100 S(u; \alpha, \beta)$$

Donde los parámetros α y β definen el intervalo de borrosidad entre la decisión nula (0) y total (100). El esquema permite otros posibles modelos para las funciones de decisión [10].

La definición de una clase de segmentos puede establecerse desde clasificadores y características permitiendo la generación de símbolos asociados a los segmentos.

Las reglas como operadores para obtener diagnósticos son construcciones lógicas, similares a las cláusulas de Horn. Una regla de este tipo define una contribución a una clase de segmentos basada en los valores de una serie de condiciones que pueden admitir cualificadores lógicos y operadores lingüísticos borrosos "very" y "more-or-less"[9],[11]. La incertidumbre se transforma y asigna desde las clases, clasificadores y reglas hacia otras clases. El valor intrínseco de cada clase se determina desde las fuentes de datos, tales como clasificadores y reglas. Si la clase posee particularizaciones, entonces el citado valor intrínseco debe normalizarse para obtener el valor final. Esta normalización está basada en considerar los casos particulares como conjuntos borrosos. La combinación de evidencias para definir una clase de segmentos desde clasificadores, reglas y particularizaciones utiliza el siguiente esquema:

Paso 1: Computar todos los clasificadores, casos

particulares y clases incluidas en las condiciones de las reglas de una clase.

Paso 2: Computar todos los predicados en las condiciones de reglas.

Paso 3: Computar el resultado de la reglas, en base a los predicados y a la certeza de la misma. Si $C_{pk}(s)$, $k=1, \dots, h$, es la certeza de cada predicado, y C es la certeza de la regla, entonces el resultado de la regla, $C_r(s)$, es:

$$C_r(s) = \frac{C}{100} \text{Min}_{k=1..h} [C_{pk}(s)]$$

Paso 4: Combinar los resultados de clasificadores y reglas, proporcionando el valor intrínseco. El valor intrínseco $IC(s)$ se evalúa desde las certezas $C_{lk}(s)$, $k=1, \dots, m$, de los m clasificadores y reglas en conjunto. Esta evaluación se realiza incrementalmente en la forma siguiente:

$$IC(s) := 0; \text{ for } k=1 \text{ to } m \text{ } IC(s) := \text{COM}(IC(s), C_{lk}(s));$$

La función $COM()$ se define como sigue según el tipo de combinación elegido, una propuesta es:

$$COM(u, v) = \begin{cases} u + v - \frac{uv}{100} & \text{si Logic} \\ 50 \left[EV\left(\frac{u}{50} - 1, \frac{v}{50} - 1\right) + 1 \right] & \text{si Evidence} \end{cases}$$

Donde $EV(a, b)$ es:

$$EV(a, b) = \begin{cases} a + b - ab & a \geq 0 \wedge b \geq 0 \\ a + b + ab & a \leq 0 \wedge b \leq 0 \\ \frac{a + b}{1 - \min(|a|, |b|)} & (a > 0 \wedge b < 0) \vee (a < 0 \wedge b > 0) \end{cases}$$

La ley "Evidence" utiliza el dominio [-1,+1] y está basada en la Teoría de la Evidencia, en la forma en que es utilizada por algunos sistemas basados en conocimiento [14].

Paso 5: Normalizar el valor intrínseco utilizando los valores de los casos particulares. Sea $C(s)$ el valor de la clase, y $C_{rk}(s)$, $k=1, \dots, n$, la certeza de los n casos particulares. La normalización es de la forma siguiente:

$$C(s) = \text{Max} (IC(s), \text{Max}_{k=1..n} [C_{rk}(s)])$$

Con objeto de estudiar mejor como se realizan los procesos en el Nivel de Segmentos, se introduce el concepto de Procesador de Segmentos [13] que define la arquitectura de procesos en este nivel y que se incluye resumidamente a continuación. Sobre el se ha definido un lenguaje que permite el desarrollo de aplicaciones.

2.1 Arquitectura del Procesador

Desde el punto de vista funcional, la arquitectura del Procesador de Segmentos sobre la que hemos desarrollado el trabajo (figura 1), se organiza en tres grandes bloques con cometidos específicos. El primero está dedicado a la definición de la partición inicial y es el cometido del módulo de presegmentación integrado en el denominado procesador Bottom-Up (BU). Este módulo está concebido como una parte intercambiable dentro del Procesador de Segmentos, pues en principio son posibles diferentes aproximaciones a la hora de establecer una partición inicial o segmentación primaria de la imagen. El segundo tiene como cometido la definición de la estructura computacional de obtención de los diagnósticos de segmentos. Está basado en la distinción de un dominio numérico o de características y otro simbólico o de clases. El tercer bloque funcional se ubica en el denominado procesador Top-Down (TD) y gestiona la atención de peticiones de diagnósticos y el control del Procesador de Segmentos. Éste incluye la evaluación del estado de la partición desde un conjunto de reglas de control que pueden producir acciones que modifiquen la definición espacial de los segmentos.

La unidad BU del Procesador de Segmentos tiene dos cometidos fundamentales: la definición del nuevo conjunto de entidades de este nivel a partir de los símbolos definidos para el nivel anterior y la atribución simbólica de las entidades así definidas. El Procesador de Segmentos opera según dos fases diferenciadas. Una directa, ascendente, ejecutada por la unidad BU en la que se calculan los grados de pertenencia a clases de segmentos de aquellos que están presentes en la partición inicial. La otra queda bajo la supervisión del procesador TD y permite desarrollar estrategias que modifican la partición de una imagen desde un conjunto de reglas de control.

La segmentación inicial producida en el presegmentador se trata como un mapa de colores y se transforma posteriormente en una partición inicial mediante un análisis de componentes conectadas, que extrae del mapa de color información espacial exhaustiva que utiliza durante la descripción simbólica de los segmentos.

3 Las Condiciones.

Una *Condición* en el marco del Procesador de Segmentos se entiende como el encadenamiento de un conjunto de premisas conectadas por conjunciones lógicas tipo "And". Las condiciones así definidas se utilizan en el Procesador de Segmentos en los diversos tipos de reglas, tanto aquellas que definen la estructura de clases, como en reglas de control o en acciones de control. Las premisas utilizadas en este nivel se enriquecerán con la inclusión de relaciones espaciales. Para clarificar el ámbito de estas condiciones, detallamos a continuación la definición sintáctica de la premisa de una condición, que se establece simbólicamente como sigue:

premise::=prem_atom_true | prem_list

```

prem_atom_true::=Is ( True, [qualifier1] x )
prem_list::= prem_atom { And prem_atom }
prem_atom::=Is( property { And property }, [qualifier1] x )
property::=[qualifier2] [order] sclassname
qualifier1::=[neig][qualifier2][spatial]
neig::=All | Any
qualifier2::=[Not][linguistic]
linguistic::=Very | MoreLess
spatial::=Above|Below|Left|Right|Beside|InBeside|Neib|
ExNeib | InNeib | Contains | user_defined_localizer
order::= Gt (number) | Eq (number) | Lt (number)

```

Los cualificadores All y Any hacen referencia al entorno y operan en conjunción con los cualificadores de relaciones espaciales que analizaremos posteriormente. Se incluyen además relaciones de orden (Gt, Eq, Lt) que afectan al grado de la propiedad sobre la cual se establece el diagnóstico. Se permite asimismo la inclusión de listas de propiedades And-conectadas. La sintaxis de escritura de una premisa de esta naturaleza se fundamenta en la expresión elemental:

Is(property , x)

Donde *x* denota el segmento foco de atención (cuya referencia puede omitirse, pero que mantendremos por claridad en la escritura) y *property* es el diagnóstico sobre el que se condiciona a *x*. Así la premisa:

Is(Green , x)

Sería la expresión formal del hecho "El segmento foco de atención es verde", y la evaluación de esta premisa devolvería el grado en que el segmento foco de atención satisface el diagnóstico asociado a la clase "Green". El sentido de una premisa puede modificarse mediante el uso de los modificadores Not, Very y MoreLess. Es fundamental en el marco de este procesador establecer premisas sobre la vecindad del segmento foco de atención, para obtener una descripción del papel particular que juegan las relaciones espaciales. La sintaxis básica para incorporar relaciones espaciales es:

Is(property , spatial x)

Donde *spatial* representa a un elemento del lenguaje que se asocia con una relación espacial de los tipos: Above, Below, Left, Right, Beside, Neib, InBeside, ExNeib, InNeib y Contains. Así la expresión :

Is(Green, Below x)

Se correspondería con la condición "Algún segmento de los que están debajo del foco de atención es verde". Por otra parte, algunos localizadores tienen acotado su ámbito de aplicación. Así, localizadores como Above, Below, Left, Right, Beside, ExNeib y Contains se evalúan sólo sobre los vecinos "externos" de un segmento dado. Por el contrario, los localizadores InBeside e InNeib producen la evaluación exclusivamente sobre los segmentos

contenidos dentro de uno dado, es decir, sobre sus vecinos "interiores". Finalmente, Neib hace que se evalúe la condición sobre cualquier vecino del segmento foco de atención. También resulta útil establecer la condición, no sobre algún vecino (opción de defecto) sino sobre todos los adyacentes, lo que se hará utilizando el modificador All. Así, si deseamos expresar condiciones como: "Todos los segmentos contenidos en el segmento foco de atención no son rojos" o " Todos los segmentos que no están debajo del foco de atención son muy cuadrados", tendríamos respectivamente las premisas:

$$Is(Not Red , All InNeib x)$$

$$Is(Very Square , All Not Bellow x)$$

La evaluación del predicado "A Above B" devuelve la medida, en el intervalo [0,100], en que el segmento A, vecino del B, se sitúa "por encima" de éste.

La utilización de predicados que incluyen relaciones espaciales entre segmentos permiten establecer también relaciones de orden, esto es comparaciones entre vecinos para una propiedad o diagnóstico dado. Las relaciones de orden permitidas son: Gt (Mayor), Eq (Igual) y Lt (Menor), cuya evaluación, establecida de manera difusa, trataremos a continuación.

Para ello, considérense dos segmentos A y B y sea SC un determinado diagnóstico en base al cual se desea establecer una relación de orden del tipo GT entre A y B, que designaremos por la expresión $SC_A Gt SC_B$, siendo SC_A y SC_B los valores que toma el diagnóstico sobre A y B respectivamente. Sea u una variable definida como la diferencia entre estos dos valores y sea δ un parámetro indicativo de la semianchura del intervalo en u que separa un resultado verdadero (100) de otro falso (0). Entonces el juicio sobre " $SC_A Gt(\delta) SC_B$ " se resuelve de acuerdo con:

$$R_{SC_A Gt(\delta) SC_B} = 100 S(u; -\delta, \delta)$$

El parámetro δ controla la incertidumbre en el juicio y es dependiente del diagnóstico utilizado para establecer la relación de orden. De esta forma será posible expresar hechos del tipo "Algún vecino del foco de atención es más negro con incertidumbre 10", como:

$$Is(Gt(10) Black , Neib x)$$

Donde Black es una clase o diagnóstico representativo de la clase segmentos negros. En análogo sentido se definen las evaluaciones de las relación del tipo "Eq" o "Lt".

Para la evaluación de las premisas se utiliza el esquema propuesto en [15], mientras la evaluación de una condición se realiza mediante una operación que devuelva el mínimo de los resultados obtenidos para cada una de las premisas que la integran.

Las condiciones también se utilizan en las reglas de control. Estas reglas de control, son diferentes de las utilizadas en la generación de clases de pixels o de seg-

mentos y se estructuran a partir de una condición o lista de condiciones, una acción de control a ejecutar si se dispara la regla y, si se requiere, una condición adicional que actúa como selector de algún segmento vecino.

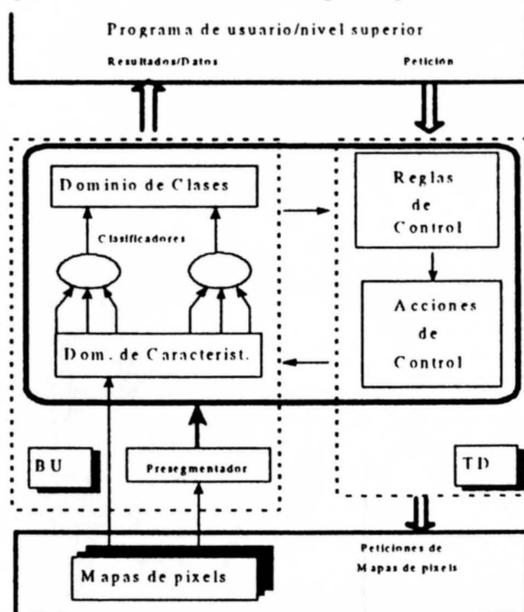


Figura 1. Estructura de procesadores y dominios en el nivel de los segmentos.

4 Aplicaciones.

Las aplicaciones han sido desarrolladas en el contexto del sistema SVEX [10]. Para este sistema se ha desarrollado un lenguaje de propósito específico orientado a objetos. Un programa escrito para este sistema se compone de módulos que afectan a cada uno de sus procesadores. La aplicación que se incluye está inspirada en un problema real tomado del dominio de la sedimentología. Este consiste en la obtención de la forma aproximada de las partículas presentes en una muestra como la que se presenta en la figura 2.a. La dificultad radica, en la adyacencia de partículas. La estrategia se articula en dos etapas. La primera y más sencilla está destinada a aislar el conjunto de partículas del fondo de la imagen. Una vez obtenida esta primera segmentación se convierte la partición en un nuevo mapa de pixels y se activa un segundo programa en el Procesador de Segmentos. En este caso el Presegmentador obtiene el perfil a partir del gradiente morfológico de las zonas que en la partición inicial corresponden a las partículas. Durante el proceso de inundación se hace crecer las partículas desde los mínimos del gradiente compitiendo con el fondo. Las zonas peor delimitadas son las que corresponden a la frontera entre dos partículas, donde evidentemente la incertidumbre acerca de la forma original es muy alta. Finalmente, (figura 2.c) se computan un conjunto de descriptores relativos a la forma geométrica y se seleccionan aquellas partículas que se clasifican como de aspecto redondeado..

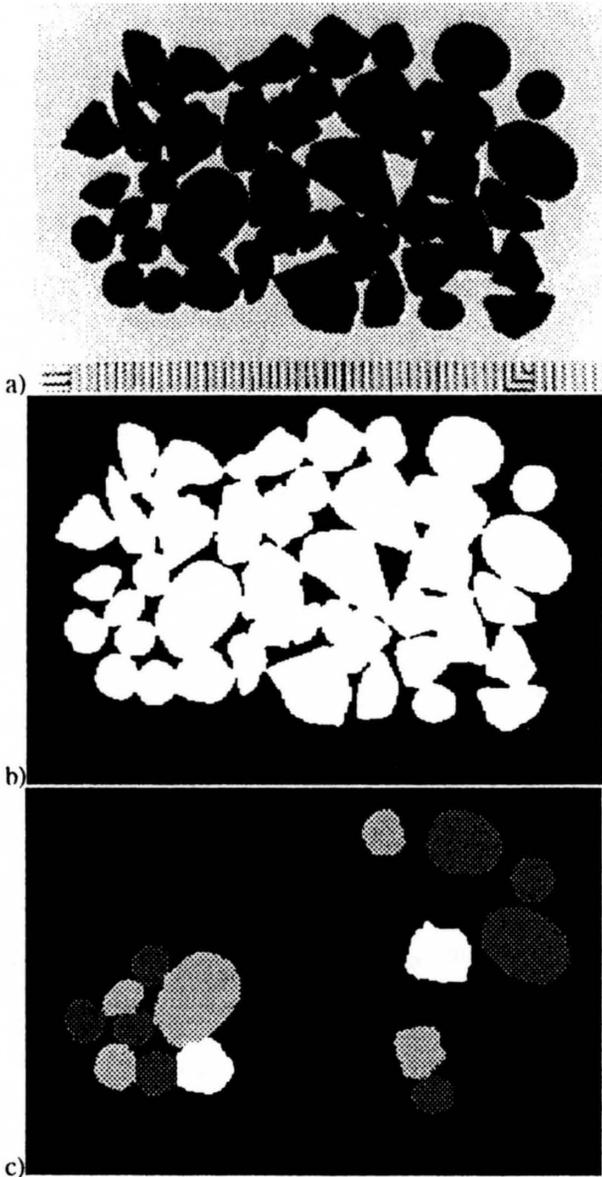


Figura 2. a) Imagen original del caso de estudio. b) Primera Partición. c) Partición final con identificación de partículas redondeadas

5 Conclusiones.

La naturaleza difusa presente en muchas de las tareas que desarrolla un sistema de visión (reconocimiento, clasificación, evaluación de relaciones espaciales, fusión de información, etc.) hacen que estos sean un marco adecuado para la utilización de técnicas de lógica difusa. Por ello, el objeto de los experimentos no ha sido el demostrar la superioridad del esquema propuesto frente a otros sistemas o algoritmos que trabajan a nivel de segmentos, sino el ilustrar aplicaciones de un nuevo esquema que permite resolver problemas de segmentación en contextos muy diferentes mediante mecanismos sencillos fundamentados en la utilización de conceptos de lógica borrosa y la utilidad de estos en el

contexto de los Sistemas de Visión por Computador Basados en Conocimiento.

Referencias

- [1] Aloimonos J., Weis I., Active Vision, *Int. J. of Comp. Vision*, 2, (1988), 333-356.
- [2] Bajcsy R., Active Perception, *Proc of the IEEE*, 76, 8, (1988), 996-1005.
- [3] Wilson R., Spann M., *Image Segmentation and Uncertainty*, Research Studies Press Ltd., 1988.
- [4] Niemann H., Brüning H., Salzbrunn R., Schröder S., A Knowledge-Based Vision System for Industrial Applications, *Machine Vision and Applications*, 3, (1990) 201-229.
- [5] Dubois D., Prade H., Yager R.R., Introduction en *Reading in Fuzzy Sets for Intelligent Systems*, Dubois D., Prade H., Yager R.R. Eds, Morgan Kaufmann Pub. Inc., pp 1-20, 1993.
- [6] Krisnapuram R., Keller J.M., Fuzzy Set Theoretic Approach to Computer Vision: An Overview, en *Fuzzy Logic Technology an Applications*, IEEE Tech. Activities Board, Marks R.J. (ed), pp 25-32, 1994.
- [7] Huntsberger T.L., Rangarajan C., Jayaramamurthy S.N., Representation of Uncertainty in Computer Vision Using Fuzzy Sets, *IEEE Trans. Comput.*, 35, 2 (1986) 145-156.
- [8] Riseman E.M., Hanson A.R., A Methodology for the Development of General Knowledge-Based Vision Systems, en *Vision, Brain and Cooperative Computation*, Ed. Arbib y Hanson, MIT Press., Cambridge Mass., pp. 285-328, 1987.
- [9] Zadeh L.A., PRUF-a meaning representation language for natural languages, en *Fuzzy Reasoning and its Applications*, Mandani y Gaines (Eds.), Academic Press, pp. 1-66, London, pp. 1-39, 1981.
- [10] Mendez J, Falcón A., Hernández F., Cabrera J., A Development Tool for Computer Vision Systems at Pixel Level, *Cybernetic & Systems* 25, 2 (1994) 289-316.
- [11] Zadeh L.A., Commonsense Knowledge Representation based on Fuzzy Logic, *IEEE Computer*, 16, 10 (1983), 61-65.
- [12] Gordon J., Shortliffe E.H., The Dempster-Shafer Theory of Evidence, en *Rule-Based Expert Systems*, Ed. Buchanan y Shortliffe, Addison-Wesley, pp. 272-292, 1984.
- [13] Cabrera J., *Sistema Basado en Conocimiento para Segmentación de Imágenes. Desarrollos y Aplicaciones*, Tesis Doctoral, Universidad de Las Palmas de Gran Canaria, 1994.