

ACTUALIZACIÓN CARTOGRÁFICA A PARTIR DE IMÁGENES DIGITALES

S. FERNÁNDEZ, P. GARCÍA Y V. MORENO.
INISEL ESPACIO. Madrid

RESUMEN

La Actualización Cartográfica a partir de Imágenes Digitales se presenta como alternativa a los métodos tradicionales. El algoritmo elegido para implementar es el Nevatia y Medioni. Dicho algoritmo está orientado a líneas y el proceso se realiza por una relajación que computa la similitud de estructuras geométricas.

ABSTRACT

Cartography updating from Digital Images is a choice to the traditional methods. Chosen algorithm to implement is the Nevatia and Medioni. This algorithm is line-based and the process is doing by a relaxation which computes most geometrical structures.

1. INTRODUCCIÓN

Uno de los mayores problemas que se plantea a la hora de manejar Cartografía, es el grado de actualización y exactitud de esta. La revisión de la cartografía existente, con una cierta frecuencia, supone un gasto importante tanto de tiempo como dinero. La Actualización Cartográfica a partir de Imágenes Digitales, es la alternativa propuesta en este trabajo, a los métodos tradicionales empleados normalmente.

Como paso previo a este proceso, siempre tendremos que efectuar una Detección y Extracción de Objetos mediante algoritmos de Detección de Bordes y de Segmentación. Los objetos extraídos de la Imagen Digital nos servirán para comparar con los elementos del mapa que queremos actualizar, este proceso es lo que en la literatura anglosajona se denomina "matching".

Hay muchos métodos desarrollados para solucionar el problema de "matching". Estos métodos se pueden clasificar según el nivel al cual se aplica el proceso. En los casos más simples este proceso se aplica a nivel de píxel y los elementos a describir serán las propiedades del píxel, tales como intensidad y color. Esta técnica es muy sensible a las variaciones de las propiedades del píxel; podemos esperar cambios en la intensidad cuando las condiciones de la imagen cambien (diferente ángulo del sol, diferentes condiciones ambientales, cambios de estaciones, etc.).

En el otro extremo, hay técnicas de "matching" que utilizan descripciones de objetos, es decir, utilizan estructuras semánticas tales como carreteras, edificios y efectúa el "matching" con sus homólogos. Estos métodos son muy resistentes a los cambios en las imágenes, pero el estado del arte en Visión Artificial es insuficiente para extraer estas descripciones de las imágenes.

2. DESCRIPCIÓN DEL ALGORITMO UTILIZADO

El algoritmo elegido para efectuar el proceso de Actualización Cartográfica es el desarrollado por R. Nevatia y G. Medioni (1984). Este algoritmo está orientado a líneas y el proceso de "matching" se realiza por una relajación que computa la similitud de estructuras geométricas.

Antes de entrar en la descripción del algoritmo en sí, es conveniente aclarar algunos términos que se utilizarán de aquí en adelante:

- **Objeto:** Cualquiera de los segmentos detectados en la imagen digital de la cual partimos. Los denotaremos como a_j .

- **Etiqueta:** Cualquiera de las primitivas del mapa que queremos actualizar. Lo denotaremos como l_j .

- **Escena:** Conjunto de objetos.

- **Modelo:** Conjunto de etiquetas.

Como paso previo tenemos que computar la posibilidad de que un objeto a_j tenga su correspondiente etiqueta l_j en el mapa modelo; esta posibilidad la denotaremos $p(i,j)$.

Este primer parámetro tiene las propiedades siguientes:

- Es posible que para un objeto a_j no haya una etiqueta l_j , por lo tanto la suma en j de todos los $p(i,j)$ será igual a 0.

- Varios objetos pueden compartir una etiqueta común; esto es debido a que la primitiva del modelo está fraccionada en el proceso de segmentación de la imagen. Según esto la suma en i de todos los $p(i,j)$ puede ser > 1 .

- Un objeto a_j puede tener varias etiquetas; por lo tanto la suma en j de $p(i,j)$ será > 1 .

Las Figuras 1 y 2 ilustran estas propiedades.

El núcleo del método se podría resumir en que cuando asignamos una etiqueta l_j a un objeto a_j , esperamos encontrar

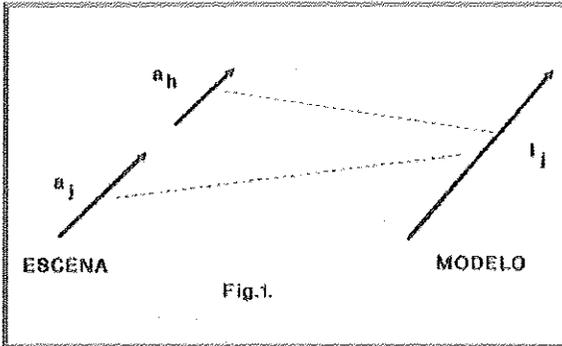


Fig. 1.

Figura 1.

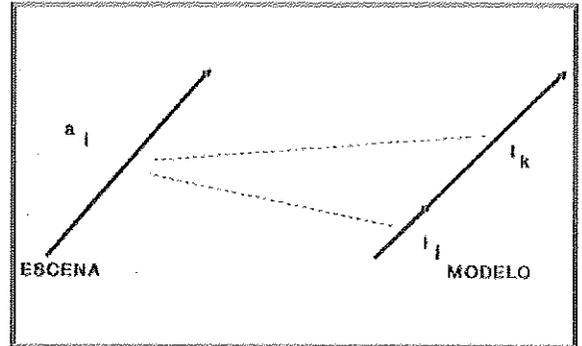


Figura 2.

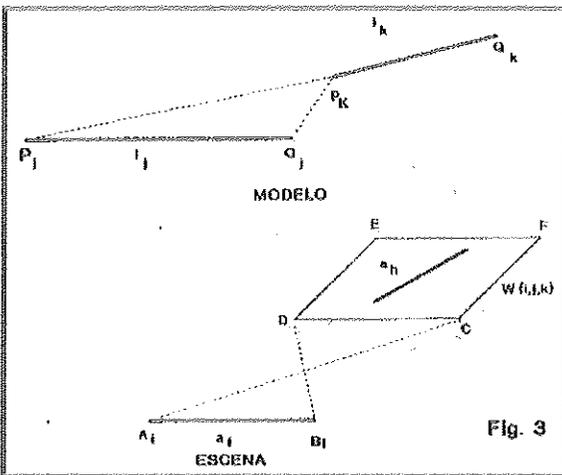


Fig. 3

Figura 3.

trar un objeto a_j con una etiqueta l_k en una cierta área dependiente de i,j,k ; este área se denota $W(i,j,k)$ y se llama ventana (i,j,k) .

Cada objeto a_i se puede representar como un vector bidimensional $A_i B_i$ y cada etiqueta l_j como vector $P_j Q_j$. Lo que esperamos encontrar es un objeto a_j con una etiqueta l_k ($P_k Q_k$). Si deslizamos l_j sobre a_i y hacemos lo propio con l_k , el área definida durante esta operación nos definirá la ventana $W(i,j,k)$ (ver Figura 3).

Finalmente necesitamos definir la relación C "es compatible con" entre dos pares (i,j) y (h,k) . $C(h,k) \iff a_j$ está en $W(i,j,k)$ y a_i en $W(h,j,k)$. Hay que comprobar ambos predicados ya que la relación no es simétrica.

Una vez definido todo lo anterior podemos hacer una descripción formal del algoritmo:

- **Asignación Inicial de Posibilidades.** Si a_j y β_j es la orientación angular de l_j , entonces tenemos lo siguiente:

$$\text{para todo } (i,j), p^0(i,j)=1 \text{ si } |\alpha_j - \beta_j| \leq 15^\circ$$

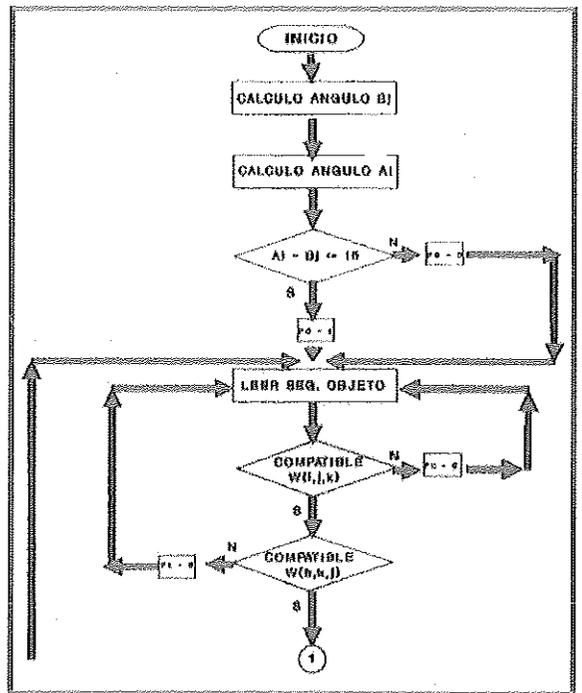


Figura 5.

en el caso contrario será 0.

La relación $\alpha_j - \beta_j \leq 15^\circ$ es lo que de ahora en adelante llamaremos criterio angular.

- **Fórmula de Iteración.** Para todo (i,j) , $p^{t+1}(i,j) = 1$ si $p^t=1$ y existe un subconjunto S de $[1,m]$ con q elementos tal que para todo s en S existe un k en $[1,n]$ tal que $p^t(k,j)=1$ y $(i,j) C(k,s)$.

Es decir, tendremos que estudiar primero la asignación de posibilidad por medio del criterio angular y posteriormente observar si q de los m posible elementos, están en $W(i,j,k)$ y en $W(h,j,k)$ y que su posibilidad en el momento t y en el mo-

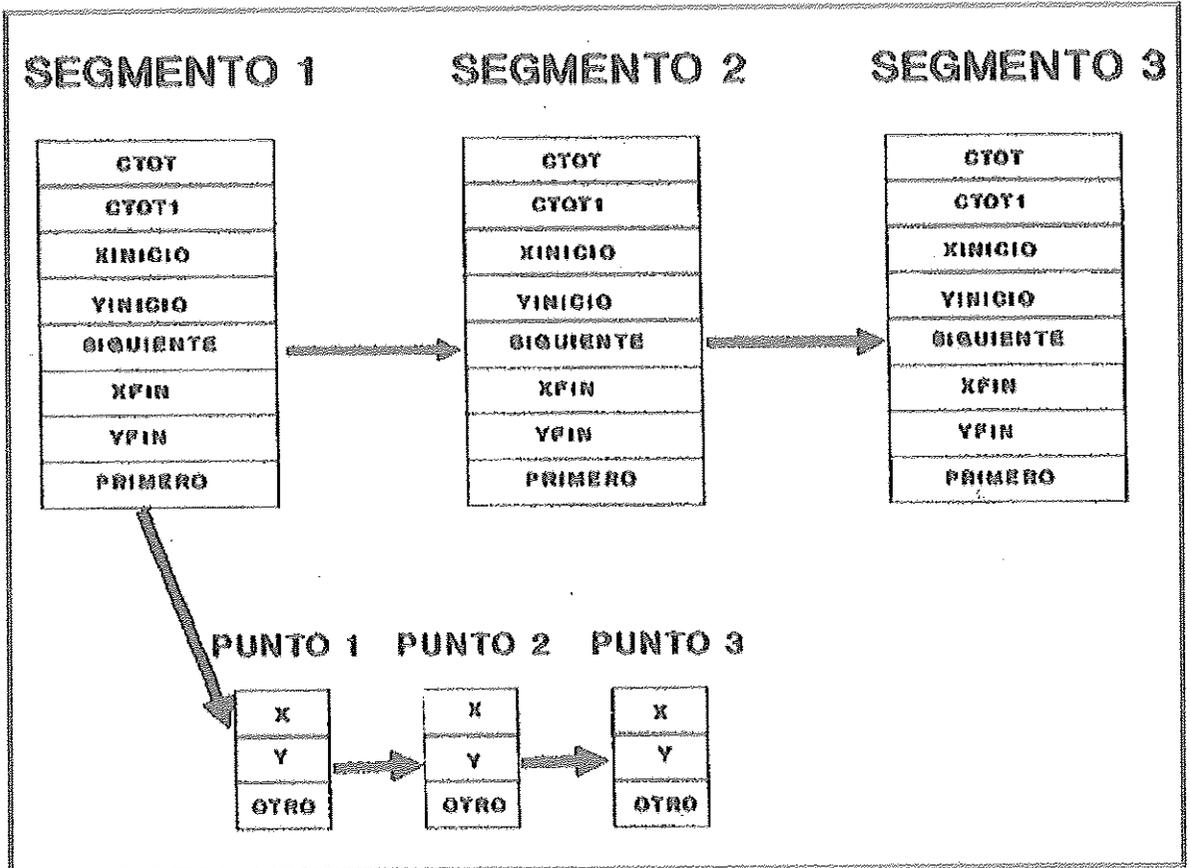


Figura 4.

mento $t+1$ sea la misma. Se considera que el proceso de "matching" es aceptable cuando q de las m posibles etiquetas están asignadas.

En la práctica el algoritmo presenta dos casos degenerados que requieren especial atención:

- la longitud del vector $A_i B_j$ y $P_i Q_j$ es igual.
- $-j_i$ y h_k tienen exactamente la misma orientación.

En estos dos casos, la ventana (normalmente un paralelogramo) se convierte en una línea, que llamaremos UV, haciendo que el estudio de compatibilidad sea erróneo. Cuando ocurra esto, Nevatia y Medioni proponen generar una ventana rectangular en la cual uno de los lados es UV y el otro es un vector normal a UV de longitud 4.

3. IMPLEMENTACIÓN DEL ALGORITMO

El desarrollo de la implementación del algoritmo se realizó en una estación de trabajo SUN SPARCSTATION IPC, bajo el sistema operativo UNIX Versión 4.1.1 y todo él integrado en el Sistema de Información Geográfica GRASS.

El lenguaje de programación elegido fue el lenguaje C

(Kernighan y Ritchie, 1988) y también se usaron las librerías de desarrollo integradas en GRASS (USA-CERL, 1988).

La estructura de la aplicación efectuada con dicho algoritmo se puede dividir en tres módulos:

- **Módulo Principal.** Controla todo el flujo de procesos y llama a las funciones necesarias de los módulos correspondientes.

- **Módulo de Funciones de los Segmentos del Modelo.** Es el conjunto de funciones que gestionan los procesos referentes a los segmentos del modelo.

- **Módulo de Funciones de los Segmentos Objeto.** Hace lo mismo que el módulo anterior pero aplicado a los segmentos objeto.

La estructura de datos básica de la aplicación consiste en una lista enlazada de segmentos y otra lista enlazada de puntos. Según esto, cada segmento tiene un puntero al siguiente segmento (además de otros campos que almacenan distintos valores) y otro puntero al primer punto de la lista de puntos (ver Figura 4).

El ordinograma de toda la aplicación está representado en las Figuras 5 y 6.

Es muy recomendable que antes de proceder al "matching" se proceda a un aligerado previo de los segmentos detectados en la imagen digital. Este aligerado consiste en borrar los puntos no significativos o redundantes de cada segmento. Para realizar este proceso hay que definir un umbral (threshold) a partir del cual consideraremos que un punto no puede borrarse. Con todo esto conseguimos que el posterior proceso de "matching" sea mucho más rápido y el resultado será idéntico que si se hubieran mantenido los puntos.

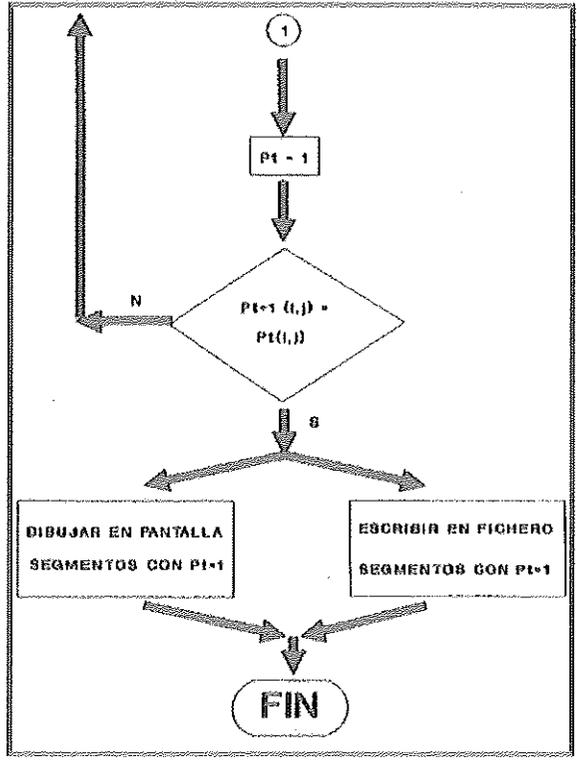


Figura 6.

4. BIBLIOGRAFIA

- ✓ KERNIGHAN, W.; RITCHIE, M., (1988): *The C Programming Language*. 2ª ed. Prentice-Hall New Jersey.
 - ✓ MEDIONI, G.; NEVATIA, R. (1984): Matching Images using Linear Features. *IEEE. Transactions on Pattern Analysis and Machine Intelligence*. PAMI-6.6 675-685.
 - ✓ USA-CERL (1988): *GRASS user's Manual*. U.S. Army Engineering and Housing Support Center. Champaign Illinois.
 - ✓ USA-CERL (1988): *GRASS programmer's Manual*. U.S. Army Engineering and Housing Support Center. Champaign Illinois.
-