



Universidad Autónoma Chapingo

**Departamento de Mecánica Agrícola
Ingeniería Mecatrónica Agrícola**

Informe 3: Métodos de segmentación

Asignatura:

Visión por computadora

Nombre del profesor:

Luis Arturo Soriano Avendaño

Alumno:

Cocotle Lara Jym Emmanuel [1710451-3]

GRADO:

6°

GRUPO:

7

Fecha de entrega: 22/06/2021

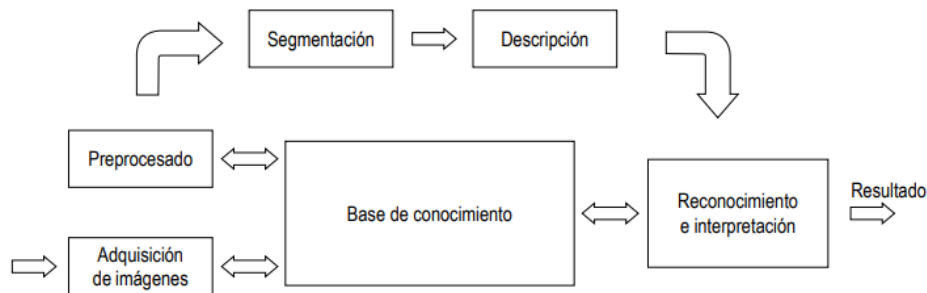
Contenido

Introducción	3
Desarrollo	4
Métodos de segmentación para el mejoramiento de imágenes.	4
Detección de bordes.....	5
Segmentación orientada a regiones	6
Visión binaria	7
Objetos binarios.....	8
Conectividad.....	8
Segmentación por histograma.	9
Umbralización	9
P-Tile Method.	9
Isodata Algorithm.....	9
Background-simmetry algorithm	10
Triangle algorithm	10
Contornos activos.....	10
Snakes	11
Snakes dinámicas y Condensación	13
Tijeras.....	14
Conjuntos de niveles	16
Dividir y combinar.....	16
Vertiente.....	16
División de regiones (agrupamiento divisivo)	17
Combinación de regiones (agrupación en clústeres aglomerativas)	17
Segmentación basada en gráficos.....	17
Cambio de media y modo de búsqueda.....	18
K-medios y mezclas de Gauss	18
Desplazamiento medio.....	20
Conclusión	23
Bibliografía	23

Introducción

La segmentación subdivide una imagen en sus partes constituyentes u objetos, con el fin de separar las partes de interés del resto de la imagen, por lo tanto, el nivel al que se lleva a cabo esta subdivisión depende del problema a resolver.

Algunas de las tareas fundamentales del procesamiento digital de imágenes son: 1) Mejoramiento de una imagen digital con fines interpretativos, y 2) Toma de decisiones de manera automática de acuerdo con el contenido de la imagen digital. Una imagen puede ser definida como una función bidimensional de intensidad de luz $f(x, y)$, donde x e y representan las coordenadas espaciales y el valor de f en un punto cualquiera (x, y) es proporcional al brillo (o nivel de gris) de la imagen en ese punto. Una imagen digital es una imagen $f(x, y)$ que se ha discretizado tanto en las coordenadas espaciales como en el brillo; y puede considerarse como una matriz cuyos índices de fila y de columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Los elementos de una distribución digital de este tipo se denominan elementos de la imagen o más comúnmente pixels, abreviatura de su denominación inglesa “picture elements” [2].



1.- Etapas del procesamiento digital de imágenes [2].

En arqueología, los métodos de procesamiento de imágenes han servido para restaurar con éxito imágenes borrosas. En la física y en campos afines, las técnicas por computador realzan de forma rutinaria imágenes de experimentos en áreas como los plasmas de alta energía y la microscopía del electrón. Igualmente, en astronomía, biología, medicina nuclear, y en aplicaciones industriales, en este último se utilizan además diversos tipos de materiales como el cuero, vidrio, metales, madera, hilos, etc.

En este informe se pretende dar a conocer diferentes métodos de segmentación de imágenes, para conocer sus características y la forma en que es posible dicho

mejoramiento, para que de esta manera emplear el adecuado método para determinada situación.

Desarrollo

Métodos de segmentación para el mejoramiento de imágenes.

La segmentación subdivide una imagen en sus partes constituyentes u objetos, con el fin de separar las partes de interés del resto de la imagen, por lo tanto, el nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. En el proceso de detectar las

partes en una imagen se identifican bordes de la imagen, o se segmenta está en regiones, líneas o curvas, etc. Otra definición considera a la segmentación como la clasificación de los puntos de la imagen (pixels), indicando las clases a la que pertenecen los diferentes pixeles. Los atributos básicos de segmentación de una imagen son: la luminancia en imágenes monocromáticas, los componentes de color en imágenes en color, textura, forma, etc.

La segmentación automática es una de las tareas más difíciles del procesamiento de imágenes, esta etapa determina el eventual éxito o fracaso del análisis, de hecho, rara vez llega a alcanzar una solución satisfactoria, se debe buscar un método alternativo de comprobación para la verificación de los resultados. Un considerable número de trabajos de investigación se centran en este problema.

Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores del nivel de gris: discontinuidad y similaridad. En la discontinuidad el método consiste en dividir una imagen basándose en los cambios bruscos del nivel de gris. Los temas más importantes en la discontinuidad son: a) detección de puntos aislados, y b) detección de líneas y c) detección de bordes de una imagen. En la similaridad, se presenta la regularidad en los valores del nivel de gris, los principales métodos están basados en a) umbralización, b) crecimiento de región, y c) división y fusión de regiones.

La segmentación de una imagen basado en la discontinuidad o en la similaridad de los valores del nivel de gris de sus pixels es aplicable tanto a las imágenes estáticas como a las dinámicas (variantes en el tiempo).

De otra manera, a lo largo de las últimas décadas se han desarrollado diversas técnicas de segmentación, las que se pueden agrupar en tres: técnicas orientadas al pixel, a los bordes y a las regiones. Dentro de ellas se pueden destacar los siguientes métodos: línea divisoria de aguas (watershed) que a partir de los mínimos en la imagen se aumenta gradualmente el nivel de gris, como si fuera agua que se vierte en un valle, hasta encontrar sus valles

vecinos; detección de bordes de las regiones mediante la búsqueda de máximos en el gradiente de la imagen o cruces por cero en la segunda derivada de la imagen; filtros en los que se optimiza una función de costo que considera la exactitud en la posición del borde y la cantidad de bordes detectados; y detección de regiones mediante agrupación de píxeles vecinos con características similares (Región Growing).

Otra corriente intenta, por medio de filtros de mediana adaptados al objeto de inspección, la estimación de una imagen libre de fallas a partir de la imagen tomada del objeto mismo. Mediante simple comparación entre la imagen real y la imagen libre de fallas estimada se segmentan las fallas. Si bien es cierto que con este último método se obtienen excelentes resultados, es necesario invertir mucho tiempo en el diseño del filtro para conseguir una adaptación al objeto. Típicamente, para el análisis de una imagen es necesario dividirla a priori en cientos de subimágenes. Cada subimagen posee un filtro morfológico distinto, configurado a partir de las características de la porción del objeto presente en ella [2].

Detección de bordes

La detección de la discontinuidad consiste en dividir una imagen basándose en los cambios bruscos del nivel de gris. Es particularmente importante porque proporciona información de los objetos de la imagen a otras tareas del procesamiento de imágenes como reconocimiento e interpretación. Los temas más importantes en la discontinuidad son: a) detección de puntos aislados, y b) detección de líneas y c) detección de bordes o contornos de una imagen. Aunque la detección de punto y línea son elementos de cualquier presentación de la segmentación de imágenes, la detección de bordes es la técnica más común para detectar discontinuidades significativas en el nivel de gris, debido a que son más frecuentes en las aplicaciones prácticas.

Los métodos de extracción de bordes de una imagen se basan en la diferencia que experimenta una característica en dos regiones adyacentes y que indican la existencia de un borde. A la vez los bordes pueden clasificarse por su anchura, ángulo de su pendiente de variación, y las coordenadas de su punto medio. En general, se identifican diferentes modelos de bordes o contornos: línea, tipo escalón, tipo rampa y tipo tejado.

Las discontinuidades son detectadas usando derivadas de primer y segundo orden, en el caso de derivadas de primer orden se utiliza el operador gradiente, mientras que en derivadas de segundo orden se utiliza el operador laplaciano [2].

Derivada de primer orden, el operador gradiente

Una de las técnicas más comunes para detectar un borde de dos dimensiones (2D), es el uso de derivadas de primer orden mediante el operador gradiente.

El gradiente de una imagen es una función $f(x, y)$ que se define como el vector bidimensional presentado en la ecuación [2].

$$G(F(x, y)) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{dF}{dx} \\ \frac{dF}{dy} \end{bmatrix}$$

Detección de bordes utilizando derivadas de segundo orden, el operador laplaciano

El laplaciano de una función bidimensional $f(x, y)$ es una derivada de segundo orden definida por la ecuación:

$$\Delta^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Enlazado de bordes y detección de límites

Si bien hay varias técnicas que detectan discontinuidades de intensidad, y que deberían dar como resultado píxeles que estuvieran en la frontera entre un objeto y su fondo, en la práctica, este conjunto de píxeles rara vez caracterizan una frontera completamente debido al ruido, interrupciones en la frontera debido a la iluminación no uniforme, y otros efectos que introducen discontinuidades de intensidad espurias. Por ello, los algoritmos de detección de bordes están seguidos por una unión y otros procedimientos de detección de frontera diseñados para reunir a los píxeles del borde en un conjunto con significado de fronteras de objetos [2].

Segmentación orientada a regiones

Se utilizan propiedades espaciales de una imagen para segmentarla por regiones, es decir la imagen es dividida en regiones conexas, en donde cada región tiene propiedades distintas que las diferencian unas de otras. En suma, se trata de extraer los objetos de una imagen, las que pueden ser procesadas de manera independiente.



2.- . Segmentación orientada a regiones [2].

Crecimiento de regiones

Es una técnica muy utilizada, que consiste en el crecimiento de regiones agrupando píxeles adyacentes que presentan características o propiedades similares. Se parte de un

conjunto de puntos “semilla” y se hace crecer las regiones, añadiendo a los puntos semilla los puntos vecinos que tengan propiedades similares, como intensidad, textura, color, etc. Por ejemplo, si la propiedad es intensidad, un criterio utilizado para incluir un píxel en una región, puede ser que la diferencia absoluta entre la intensidad del píxel y la intensidad de la semilla sea menor que un umbral, por ejemplo, $T = 8$. Los puntos semillas pueden ser escogidos manualmente o de manera automática [2].

División y fusión de regiones

Consiste en dividir inicialmente una imagen en un conjunto de regiones arbitrarias disjuntas, por ejemplo 64 divisiones, después, dependiendo del criterio de segmentación, regiones adyacentes son fusionadas si tienen propiedades similares como nivel de gris similares, o son divididos si no comparten las mismas propiedades, como variaciones considerables de niveles de gris. Finalmente, la imagen queda segmentada en un conjunto de regiones homogéneas [2].



3.- Imagen segmentada en 1405 regiones [2].

Visión binaria.

La binarización de una imagen consiste en un proceso de reducción de la información de esta, en la que sólo persisten dos valores: verdadero y falso. En una imagen digital, estos valores, verdadero y falso, pueden representarse por los valores 0 y 1 o, más frecuentemente, por los colores negro (valor de gris 0) y blanco (valor de gris 255).

En el proceso y análisis de imagen, la binarización se emplea para separar las regiones u objetos de interés en una imagen del resto. Las imágenes binarias se usan en operaciones booleanas o lógicas para identificar individualmente objetos de interés o para crear máscaras sobre regiones.

En muchos casos, una imagen binaria es el resultado de una segmentación por niveles de gris o de una segmentación por selección de un rango de color determinado. En otros casos, una imagen binaria es simplemente el resultado de una selección interactiva de regiones de interés, las cuales se utilizarán como máscaras de comparación o referencia [4].

Objetos binarios

La formación de objetos binarios en una imagen es una consecuencia directa del proceso de binarización de la misma.

Un objeto binario es una porción de la imagen formada por un conjunto de píxeles con valor verdadero (p. ej. blancos) conectados entre sí y totalmente rodeados por píxeles con valor falso (p. ej. negro).



4.- Objetos binarios [4].

Conectividad

Cuando se habla de píxeles conectados entre sí o yuxtapuestos, es necesario precisar el tipo de conexión o conectividad que se acepta como válida. En principio, un píxel de una malla rectangular puede estar conectado con los ocho píxeles que le rodean en un espacio plano o solamente con los cuatro más cercanos. Ello dará lugar a hablar de una conexión de tipo 8 o de tipo 4.

0	0	0	0	0	0
0	0	1	0	0	0
0	1	1	1	0	0
0	0	1	0	1	0
0	0	0	0	0	0

Los píxeles marcados en rojo forman un objeto con una conexión de tipo 4, mientras que el píxel marcado en verde formaría un nuevo objeto con este tipo de conexión. Por el contrario, si la conexión admitida es de tipo 8 (incluye los píxeles que se conectarían teóricamente sólo por una esquina), los cinco píxeles con valor verdadero (1) en el ejemplo formarían un solo objeto [4].

Segmentación por histograma.

Umbralización

La umbralización es uno de los métodos más importantes de la segmentación de imágenes. Se define el umbral como una función que convierte una imagen con diferentes tonalidades en una imagen en blanco y negro. Si la imagen original es $f(x, y)$, la imagen umbralizada $g(x, y)$ y se fija un umbral U ($0 < U < 255$), la operación de umbralizado se define como la ecuación:

$$g(x, y) = 255 \text{ si } f(x, y) > \text{Umbral}$$

$$g(x, y) = 0 \text{ si } f(x, y) \leq \text{Umbral}$$

Se selecciona un umbral que permita agrupar los pixels de una imagen pertenecientes a los diversos objetos de la misma imagen diferenciándolos del fondo. De esta manera la segmentación basada en el histograma se basa en la elección de uno o varios umbrales que permiten agrupar los puntos de la imagen en regiones de características similares en función de sus niveles de gris [2].

P-Tile Method.

Este método utiliza conocimiento acerca del área del histograma que ocupan los objetos que se quieren detectar. Suponiendo que, para una aplicación dada, los objetos ocupan sobre un p% del área de la imagen.

Utilizando el conocimiento de esta partición (en la imagen), uno o más umbrales pueden ser elegidos asignando un porcentaje de pixels a los objetos.

Evidentemente, este método tiene un uso muy limitado. Solamente unas pocas aplicaciones permiten estimar el área de forma general [3].

Isodata Algorithm

Es una técnica iterativa que se utiliza para la obtención del umbral correcto. El histograma es inicialmente segmentado en dos partes utilizando un umbral de comienzo tal como la mitad del máximo valor del rango dinámico.

A continuación, se computa la media de los valores asociados con cada una de las partes en que ha quedado segmentado el histograma m_1 , m_2 . Utilizando esos valores se calcula un nuevo valor umbral mediante la fórmula:

$$\alpha = \frac{m_1 + m_2}{2}$$

El proceso continuo hasta que en dos pasos consecutivos el valor umbral calculado no cambia [3].

Background-simmetry algorithm

Esta técnica asume la existencia de un pico dominante ($\max p$) para el background que además es simétrico con respecto a su valor máximo.

El algoritmo busca en la parte perteneciente al background (derecha del pico) el valor de intensidad que se corresponde con un cierto porcentaje de puntos ($p\%$).

Si tomásemos como porcentaje el 95%, el valor de luminosidad se encuentra a la derecha del pico. Para este valor de brillo (216), el 5% de los puntos se encuentran a la derecha del pico. Como suponemos que existe una simetría entorno a dicho pico se utiliza como umbral un desplazamiento a la izquierda del máximo igual al desplazamiento que existe desde el máximo hasta la localización del porcentaje marcado ($p\%$) [3].

Triangle algorithm

Esta técnica se basa en la detección del umbral correcto. La idea consiste en lo siguiente: Se construye una línea entre el valor máximo del histograma (b_{\max}) y el valor más bajo ($b_{\min}=(p=0)\%$). La distancia entre la línea y el histograma $h(b)$ se calcula para todos los valores de b , desde $b=b_{\min}$ hasta $b=b_{\max}$.

El valor de luminosidad b_0 , donde la distancia entre $h(b_0)$ y la línea es máxima es el valor umbral elegido (b_0). Esta técnica es muy efectiva cuando los pixels de los objetos producen un pico suave en el histograma.

Una técnica de clusterización (dos o más regiones) basada en umbralizaciones del histograma es la siguiente:

1. Se considera que la imagen entera es una única región y se calcula su histograma para cada componente de interés.
2. Sobre el histograma calculado se obtiene el pico mayor y se utiliza como valor umbral uno de los lados del pico. Utilizando dicho umbral se segmenta la imagen en dos regiones.
3. Se suaviza la imagen umbralizada.
4. Se repiten los pasos del 1-3 para cada región hasta que no se crean nuevas subregiones, es decir, los histogramas de cada subregión no tienen picos significativos.

El valor de luminosidad b_0 , donde la distancia entre $h(b_0)$ y la línea es máxima es el valor umbral elegido (b_0). Esta técnica es muy efectiva cuando los pixels de los objetos producen un pico suave en el histograma [3].

Contornos activos.

Mientras que las líneas, los puntos de fuga y los rectángulos son comunes en el mundo hecho por el hombre, las curvas correspondientes a los límites de los objetos son aún más

comunes, especialmente en el entorno natural. En esta sección, describimos tres enfoques relacionados para localizar dichas curvas de límite en imágenes.

La primera, originalmente llamada snakes por sus inventores (Kass, Witkin y Terzopoulos 1988), es una curva spline bidimensional que minimiza la energía y que evoluciona (se mueve) hacia características de imagen como bordes fuertes. La segunda, las tijeras inteligentes (Mortensen y Barrett 1995), permiten al usuario esbozar en tiempo real una curva que se aferra a los límites de los objetos. Por último, las técnicas de conjunto de niveles evolucionan la curva como el conjunto cero de una función característica, lo que les permite cambiar fácilmente la topología e incorporar estadísticas basadas en regiones.

Los tres son ejemplos de contornos activos (Blake e Isard 1998; Mortensen 1999), ya que estos detectores de límites se mueven iterativamente hacia su solución final bajo la combinación de imágenes y fuerzas opcionales de guía del usuario.

Snakes

Las Snakes son una generalización bidimensional de las splines que minimizan la energía.

$$\varepsilon_{int} = \int \alpha(s) \|f_s(s)\|^2 + \beta(s) \|f_{ss}(s)\|^2 ds$$

donde s es la longitud del arco a lo largo de la curva $f(s) = (x(s), y(s))$ y $\alpha(s)$ y $\beta(s)$ son funciones de ponderación de continuidad de primer y segundo orden análogas a los términos $s(x, y)$ y $c(x, y)$. Podemos discretizar esta energía muestreando la posición de la curva inicial uniformemente a lo largo de su longitud para obtener:

$$E_{int} = \sum_i \frac{\alpha(i) \|f(i+1) - f(i)\|^2}{h^2} + \frac{\beta(i) \|f(i+1) - 2f(i) + f(i-1)\|^2}{h^4}$$

Donde h es el tamaño del paso, que se puede descuidar si remuestreamos la curva a lo largo de su longitud de arco después de cada iteración. Además de esta energía spline interna, una Snake minimiza simultáneamente los potenciales externos basados en imágenes y en restricciones. Los potenciales basados en imágenes son la suma de varios términos donde el término de línea atrae a la snake a crestas oscuras, el término de borde la atrae a gradientes fuertes (bordes), y el término de término la atrae a terminaciones de línea.

Las personas también a veces extraen bordes y luego usan un mapa de distancia a los bordes como una alternativa a estos dos potenciales originalmente propuestos. En aplicaciones interactivas, también se puede agregar una variedad de restricciones colocadas por el usuario, por ejemplo, fuerzas atractivas (resorte) hacia los puntos de anclaje $d(i)$.

$$E_{spring} = k_i \|f(i) - d(i)\|^2$$

Así como fuerzas repulsivas de $\frac{1}{r}$. A medida que las Snakes evolucionan minimizando su energía, a menudo se "ingredan" y "se deslizan", lo que explica su nombre popular [1].

Redes elásticas y muelles resbaladizos

Una variante interesante sobre snakes, propuesta por primera vez por Durbin y Willshaw (1987) y más tarde reformulada en un marco de minimización de energía por Durbin, Szeliski y Yuille (1989), es la formulación neta elástica del Problema del Viajante de Vendedores (TSP). Recuerde que, en un TSP, el vendedor debe visitar cada ciudad una vez mientras minimiza la distancia total recorrida. Una Snake que está obligada a pasar por cada ciudad podría resolver este problema (sin ninguna garantía de optimalidad) pero es imposible saber de antemano qué punto de control del snake debe estar asociado con cada ciudad. En lugar de tener una restricción fija entre los nodos de la snake y las ciudades, se supone que una ciudad pasa cerca de algún punto a lo largo del recorrido. En una interpretación probabilística, cada ciudad se genera como una mezcla de gaussianos centrados en cada punto del recorrido.

$$p(d(j)) = \sum_i p_{ij}$$

Con:

$$p_{ij} = e^{-\frac{d_{ij}^2}{2\sigma^2}}$$

Donde σ es la desviación estándar del Gaussiano.

Esta energía deriva su nombre del hecho de que, a diferencia de un resorte regular, que acopla un punto de serpiente dado a una restricción dada, esta energía alternativa define un resorte resbaladizo que permite que la asociación entre las restricciones (ciudades) y los puntos de curva (recorrido) evolucione con el tiempo (Szeliski 1989).

Para calcular una buena solución para el TSP, la energía de asociación de datos de resorte resbaladizo se combina con una energía de suavidad interna regular de primer orden para definir el costo de un recorrido. El recorrido $f(s)$ se inicializa como un pequeño círculo alrededor de la media de los puntos de la ciudad y σ se reduce progresivamente. Para los valores de σ grandes, el recorrido intenta mantenerse cerca del centroide de los puntos, pero a medida que disminuye σ cada ciudad tira cada vez más fuertemente en sus puntos turísticos más cercanos (Durbin, Szeliski y Yuille 1989). En el límite como $\sigma \rightarrow 0$, cada ciudad tiene la garantía de capturar al menos un punto de recorrido y los recorridos entre las citas posteriores se convierten en líneas rectas [1].

Splines y priores de forma

Si bien las snakes pueden ser muy buenas para capturar los detalles finos e irregulares en muchos contornos del mundo real, a veces exhiben demasiados grados de libertad, lo que hace que sea más probable que puedan quedar atrapadas en mínimos locales durante su evolución. Una solución a este problema es controlar la snake con menos grados de libertad mediante el uso de aproximaciones B-spline. La Snake B resultante se puede escribir como:

$$f(s) = \sum_k B_k(s) x_k$$

Si el objeto que se rastrea o se reconoce tiene grandes variaciones en la ubicación, la escala o la orientación, estas se pueden modelar como una transformación adicional en los puntos de control, por ejemplo, $x'_k = sR x_k + t$, que se puede estimar al mismo tiempo que los valores de los puntos de control. Alternativamente, se pueden ejecutar etapas separadas de detección y alineación para localizar y orientar primero los objetos de interés. En una snake B, debido a que la snake está controlada por menos grados de libertad, hay menos necesidad de las fuerzas de suavidad interna utilizadas con las snakes originales, aunque estas todavía se pueden derivar e implementar utilizando el análisis de elementos finitos, es decir, tomando derivadas e integrales de las funciones de base B-spline [1].

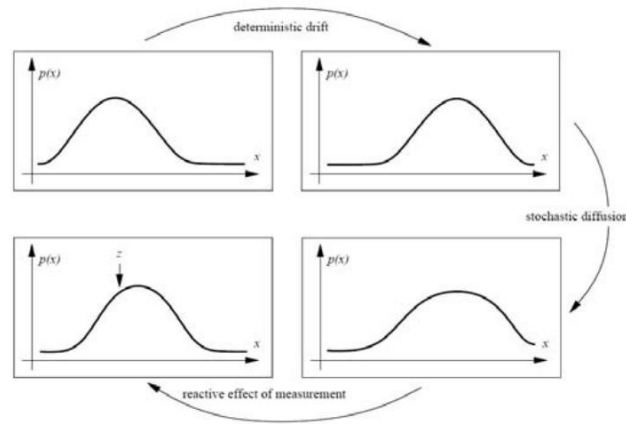
Snakes dinámicas y Condensación

En muchas aplicaciones de contornos activos, el objeto de interés se rastrea de fotograma a fotograma a medida que se deforma y evoluciona. En este caso, tiene sentido utilizar estimaciones del marco anterior para predecir y restringir las nuevas estimaciones. Una forma de hacer esto es usar el filtrado de Kalman, que resulta en una formulación llamada snakes de Kalman. El filtro de Kalman se basa en un modelo dinámico lineal de evolución de parámetros de forma:

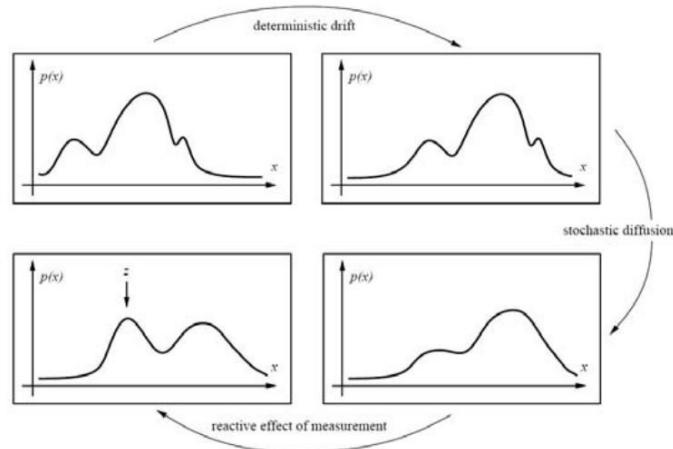
$$x_t = A x_{t-1} + w_t$$

Donde x_t y x_{t-1} son las variables de estado actual y anterior, A es la matriz de transición lineal, y w es un vector de ruido (perturbación), que a menudo se modela como gaussiano. Las matrices A y la covarianza de ruido se pueden aprender con anticipación observando las secuencias típicas del objeto que se está rastreando. El modelo dinámico lineal provoca un cambio determinista (deriva) en la estimación anterior, mientras que el ruido del proceso (perturbación) provoca una difusión estocástica que aumenta la entropía del sistema (falta de certeza). Las nuevas mediciones del marco actual restauran parte de la certeza (pico) en la estimación actualizada. En muchas situaciones, sin embargo, como cuando se rastrea en desorden, se puede obtener una mejor estimación para el contorno

si eliminamos las suposiciones de que la distribución es gaussiana, que es lo que requiere el filtro de Kalman.



5.- El filtro de Kalman modela las distribuciones como uni-modales [1].



6.- Algunas aplicaciones requieren distribuciones multimodales más generales [1].

Las técnicas de filtrado de partículas representan una distribución de probabilidad utilizando una colección de muestras puntuales ponderadas. Para actualizar las ubicaciones de las muestras según la dinámica lineal (deriva determinista), los centros de las muestras se actualizan y se generan múltiples muestras para cada punto. Estos son entonces perturbados para dar cuenta de la difusión estocástica, es decir, sus ubicaciones son movidas por vectores aleatorios tomados de la distribución de w . Finalmente, los pesos de estas muestras se multiplican por la densidad de probabilidad de medición, es decir, tomamos cada muestra y medimos su probabilidad dada las (nuevas) mediciones actuales.

Tijeras

Los contornos activos permiten a un usuario especificar aproximadamente un límite de interés y hacer que el sistema evolucione el contorno hacia una ubicación más precisa, así

como rastrearlo a lo largo del tiempo. Los resultados de esta evolución de la curva, sin embargo, pueden ser impredecibles y pueden requerir sugerencias adicionales basadas en el usuario para lograr el resultado deseado.

Un enfoque alternativo es hacer que el sistema optimice el contorno en tiempo real mientras el usuario está dibujando (Mortensen 1999). El sistema de tijeras inteligentes desarrollado por Mortensen y Barrett (1995) hace precisamente eso. A medida que el usuario dibuja un contorno aproximado, el sistema calcula y dibuja una curva mejor que se aferra a los bordes de alto contraste.

Para calcular la ruta de curva óptima (cable vivo), la imagen se preprocesa primero para asociar costos bajos con bordes (enlaces entre vecinos horizontales, verticales y diagonales, es decir, vecinos N_8) que probablemente sean elementos de límite. Su sistema utiliza una combinación de cruce cero, magnitudes de gradiente y orientaciones de gradiente para calcular estos costos.

A medida que el usuario traza una curva aproximada, el sistema vuelve a calcular continuamente la ruta de menor coste entre el punto de semilla inicial y la ubicación actual del ratón utilizando el algoritmo de Dijkstra, un algoritmo de programación dinámica de primer nivel que termina en la ubicación de destino actual.

Con el fin de evitar que el sistema salte de forma impredecible, el sistema "congelará" la curva hasta la fecha (restablecerá el punto de semilla) después de un período de inactividad. Para evitar que el cable vivo salte a los contornos adyacentes de mayor contraste, el sistema también "aprende" el perfil de intensidad bajo la curva optimizada de corriente, y lo utiliza para mantener preferentemente el cable en movimiento a lo largo del mismo límite (o de aspecto similar).

Se han propuesto varias extensiones al algoritmo básico, que funciona notablemente bien incluso en su forma original. Mortensen y Barrett (1999) utilizan el tobogán, que es una forma simple de segmentación de regiones hidrográficas, para presegmentar la imagen en regiones cuyos límites se convierten en candidatos para rutas curvas optimizadas. Los límites de región resultantes se convierten en un gráfico mucho más pequeño, donde los nodos se encuentran donde se encuentran tres o cuatro regiones. El algoritmo de Dijkstra se ejecuta en este gráfico reducido, lo que resulta en un rendimiento mucho más rápido (y a menudo más estable). Otra extensión de las tijeras inteligentes es utilizar un marco probabilístico que tenga en cuenta la trayectoria actual del límite, dando como resultado un sistema llamado JetStream.

En lugar de volver a calcular una curva óptima en cada instante de tiempo, se puede desarrollar un sistema más simple simplemente "ajustando" la posición actual del ratón al punto límite probable más cercano [1].

Conjuntos de niveles

Una limitación de los contornos activos basados en curvas paramétricas de la forma $f(s)$, por ejemplo, snakes, B-snakes y condensación, ya que es difícil cambiar la topología de la curva a medida que evoluciona. (McInerney y Terzopoulos (1999, 2000) describen un enfoque para hacer esto. Además, si la forma cambia drásticamente, también puede ser necesaria la reparificación de curvas.

Una representación alternativa para tales contornos cerrados es utilizar un conjunto de niveles, donde los cruces cero de una función característica definen la curva.

Los conjuntos de niveles evolucionan para ajustar y rastrear objetos de interés modificando la función de incrustación subyacente $\varphi(x, y)$ en lugar de la curva $f(s)$. Para reducir la cantidad de cómputo requerida, solamente una tira pequeña (frontera) alrededor de las localizaciones del cruce cero actual, lo que da lugar a lo que se llaman marchas de métodos rápidos [1].

Dividir y combinar.

Es una serie de algoritmos que proceden dividiendo recursivamente toda la imagen en partes basadas en estadísticas de región o, por el contrario, fusionando píxeles y regiones de manera jerárquica. También es posible combinar tanto la división como la fusión comenzando con una segmentación de grano medio y luego permitiendo operaciones de fusión y división [1].

Vertiente

Una técnica relacionada con el umbral, ya que opera en una imagen en escala de grises, es el cálculo de vertientes. Esta técnica segmenta una imagen en varias vertientes de captación, que son las regiones de una imagen (interpretada como un campo de altura o paisaje). Una manera eficiente de calcular tales regiones es comenzar a inundar el paisaje en todos los mínimos locales y etiquetar las crestas dondequiera que se encuentren los componentes en evolución diferente. Todo el algoritmo se puede implementar utilizando una cola de prioridad de píxeles y búsqueda de amplitud (Vincent y Soille 1991).

Dado que las imágenes rara vez tienen regiones oscuras separadas por crestas más claras, la segmentación de vertientes generalmente se aplica a una versión suavizada de la imagen de magnitud de degradado, lo que también la hace utilizable con imágenes en color. Tales técnicas terminan encontrando regiones suaves separadas por límites visibles (gradiente más alto).

Desafortunadamente, la segmentación de vertientes asocia una región única con cada mínimo local, lo que puede conducir a una sobre-segmentación. Por lo tanto, la segmentación de vertientes se utiliza a menudo como parte de un sistema interactivo, donde el usuario primero marca las ubicaciones de las semillas (con un clic o un trazo corto) que corresponden a los centros de los diferentes componentes deseados. También muestra el resultado de una versión mejorada de la vertiente que utiliza la morfología local para suavizar y optimizar los límites que separan las regiones [1].

División de regiones (agrupamiento divisivo)

Dividir la imagen en regiones sucesivamente más finas es una de las técnicas más antiguas en visión artificial. Ohlander, Price y Reddy (1978) presentan una técnica de este tipo, que primero calcula un histograma para toda la imagen y luego encuentra un umbral que separa mejor los picos grandes en el histograma. Este proceso se repite hasta que las regiones son bastante uniformes o por debajo de un cierto tamaño.

Combinación de regiones (agrupación en clústeres aglomerativas)

Las técnicas de fusión de regiones también se remontan a los inicios de la visión por computadora. Brice y Fennema (1970) utilizan una cuadrícula dual para representar los límites entre los píxeles y las regiones de fusión en función de sus longitudes de límite relativas y la fuerza de los bordes visibles en estos límites.

En el clustering de datos, los algoritmos pueden vincular clústeres entre sí en función de la distancia entre sus puntos más cercanos (clustering de enlace único), sus puntos más lejanos (clustering de enlace completo) o algo intermedio.

Una versión muy simple de la fusión basada en píxeles combina regiones adyacentes cuya diferencia de color promedio está por debajo de un umbral o cuyas regiones son demasiado pequeñas. Segmentar la imagen en tales super-píxeles, que no son semánticamente significativas, puede ser una etapa de preprocesamiento útil para hacer algoritmos de mayor nivel como la coincidencia estéreo, flujo óptico, y reconocimiento tanto más rápido como más robusto [1].

Segmentación basada en gráficos

Mientras que muchos algoritmos de fusión simplemente aplican una regla fija que agrupa píxeles y regiones, Felzenszwalb y Huttenlocher (2004) presentan un algoritmo de fusión que utiliza diferencias relativas entre regiones para determinar cuáles deben fusionarse; produce un algoritmo que optimiza de forma demostrable una métrica de agrupación global. Comienzan con una medida de disimilitud píxel a píxel $w(e)$ que mide, por ejemplo, las diferencias de intensidad entre los vecinos N_8 .

Para cualquier región R , su diferencia interna se define como el mayor peso de borde en el árbol de expansión mínimo de la región.

$$Int(R) = \max_{e \in MST(R)} w(e)$$

Para dos regiones adyacentes cualesquiera con al menos una arista que conecta sus vértices, la diferencia entre estas regiones se define como la arista de peso mínimo que conecta las dos regiones.

$$Dif(R_1, R_2) = \min_{e = (v_1, v_2) | v_1 \in R_1, v_2 \in R_2} w(e)$$

Su algoritmo combina dos regiones adyacentes cualesquiera cuya diferencia es menor que la diferencia interna mínima de estas dos regiones.

Al fusionar regiones en orden decreciente de los bordes que las separan (que se pueden evaluar de manera eficiente utilizando una variante del algoritmo de árbol de expansión mínimo de Kruskal), producen segmentaciones que no son ni demasiado finas (existen regiones que podrían haberse fusionado) ni demasiado gruesas (hay regiones que podrían dividirse sin ser fusionables). Para las vecindades de píxeles de tamaño fijo, el tiempo de ejecución de este algoritmo es $O(N \log N)$, donde N es el número de píxeles de imagen, lo que lo convierte en uno de los algoritmos de segmentación más rápidos [1].

Cambio de media y modo de búsqueda.

Las técnicas de búsqueda de medias y modos, como k-medias y mezclas de gaussianos, modelan los vectores de características asociados con cada píxel (por ejemplo, color y posición) como muestras de una función de densidad de probabilidad desconocida y luego intentan encontrar clústeres (modos) en esta distribución. Las k-medias y mezclas de gaussianos utilizan un modelo paramétrico de la función de densidad para responder a esta pregunta, es decir, asumen que la densidad es la superposición de un pequeño número de distribuciones más simples (por ejemplo, gaussianos) cuyas ubicaciones (centros) y forma (covarianza) se pueden estimar. El desplazamiento medio, por otro lado, suaviza la distribución y encuentra sus picos, así como las regiones del espacio de entidades que corresponden a cada pico. Puesto que una densidad completa se está modelando, este acercamiento se llama no-paramétrico [1].

K-medios y mezclas de Gauss

Mientras que k-means modela implícitamente la densidad de probabilidad como una superposición de distribuciones esféricamente simétricas, no requiere ningún razonamiento probabilístico o modelado. En su lugar, al algoritmo se le da el número de clústeres k que se supone que debe encontrar; a continuación, actualiza de forma iterativa

la ubicación del centro del clúster en función de los ejemplos más cercanos a cada centro. El algoritmo se puede inicializar muestreando aleatoriamente k centros de los vectores de entidad de entrada. También se han desarrollado técnicas para dividir o fusionar centros de clúster basados en sus estadísticas, y para acelerar el proceso de encontrar el centro medio más cercano.

En mezclas de gaussianos, cada centro de conglomerado se incrementa con una matriz de covarianza cuyos valores se reestiman a partir de las muestras correspondientes. En lugar de utilizar vecinos más cercanos para asociar muestras de entrada con centros de clúster, se utiliza una distancia de Mahalanobis.

$$d(x_i, \mu_k; \Sigma_k) = \|x_i - \mu_k\|_{\Sigma_k^{-1}} = (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)$$

Donde x_i son las muestras de entrada, μ_k son los centros del clúster y Σ_k son sus estimaciones de covarianza. Las muestras se pueden asociar con el centro de clúster más cercano (una asignación difícil de pertenencia) o se pueden asignar suavemente a varios clústeres cercanos.

Este último enfoque, más comúnmente utilizado, corresponde a la reevaluación iterativa de los parámetros para una mezcla de función de densidad gaussianos.

$$p(x|\{\pi_k, \mu_k, \Sigma_k\}) = \sum_k \pi_k N(x|\mu_k, \Sigma_k)$$

Donde π_k son los coeficientes de mezcla, μ_k y Σ_k son las medias y covarianzas gaussianas, y N es la distribución normal (gaussiana).

Para calcular iterativamente (una estimación local) máxima probable para los parámetros de mezcla desconocidos $\{\pi_k, \mu_k, \Sigma_k\}$, el algoritmo de maximización de expectativas (EM) procede en dos etapas alternas:

1. La etapa de expectativa (paso E) estima las responsabilidades

$$z_{ik} = \frac{1}{Z_i} \pi_k N(x_i|\mu_k, \Sigma_k)$$

Con:

$$\sum_k z_{ik} = 1$$

Que son las estimaciones de la probabilidad de que se generó una muestra x_i a partir del k -ésimo cúmulo gaussiano.

2. La etapa de maximización (paso M) actualiza los valores de los parámetros.

$$\mu_k = \frac{1}{N_k} \sum_i z_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_i z_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

Donde:

$$N_k = \sum_i z_{ik}$$

es una estimación del número de puntos de muestra asignados a cada clúster [1].

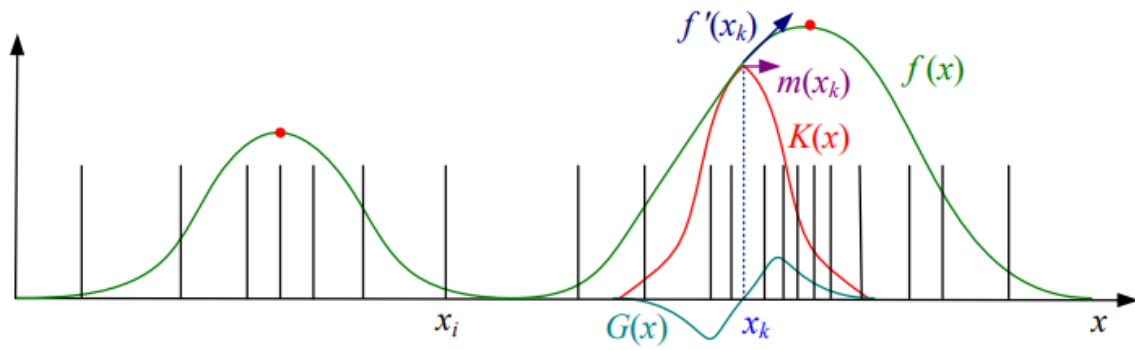
Desplazamiento medio

Mientras que k-medias y mezclas de gaussianos utilizan una forma paramétrica para modelar la función de densidad de probabilidad que se está segmentando, el desplazamiento medio modela implícitamente esta distribución utilizando un modelo no paramétrico continuo suave. La clave para el cambio medio es una técnica para encontrar eficientemente picos en esta distribución de datos de alta dimensión sin calcular nunca la función completa explícitamente.

La primero, entonces, es cómo estimar la función de densidad dado un conjunto escaso de muestras. Uno de los enfoques más simples es simplemente suavizar los datos.

$$f(x) = \sum_i K(x - x_i) = \sum_i \frac{k(\|x - x_i\|^2)}{2}$$

Donde x_i son las muestras de entrada y $k(r)$ es la función del núcleo (o ventana de Parzen). Este enfoque se conoce como estimación de la densidad del núcleo o la técnica de la ventana de Parzen. Una vez que hemos calculado $f(x)$, podemos encontrar sus máximos locales utilizando el ascenso de gradiente o alguna otra técnica de optimización.



7.- Visualización unidimensional de la estimación de la densidad del núcleo, su derivada y un desplazamiento medio [1].

El problema con este enfoque de "fuerza bruta" es que, para dimensiones más altas, se vuelve computacionalmente prohibitivo evaluar $f(x)$ sobre el espacio de búsqueda completo. En su lugar, el desplazamiento medio utiliza una variante de lo que se conoce en la literatura de optimización como descenso de gradiente de reinicio múltiple. Comenzando con alguna conjetura para un máximo local, y_k , que puede ser un punto de datos de entrada aleatorio x_i , el desplazamiento medio calcula el gradiente de la estimación de densidad $f(x)$ en y_k y da un paso cuesta arriba en esa dirección. El gradiente de $f(x)$ está dado por:

$$\nabla f(x) = \sum_i (x_i - x)G(x - x_i) = \sum_i (x_i - x)g\left(\frac{\|x - x_i\|^2}{h^2}\right)$$

Donde:

$$g(r) = -k'(r)$$

Y $k'(r)$ es la primera derivada de $k(r)$. Podemos reescribir el gradiente de la función de densidad como

$$\nabla f(x) = \left[\sum_i G(x - x_i) \right] m(x)$$

Donde el vector

$$m(x) = \frac{\sum_i x_i G(x - x_i)}{\sum_i G(x - x_i)} - x$$

Se llama desplazamiento medio, ya que es la diferencia entre la media ponderada de los vecinos x_i alrededor de x y el valor actual de x

En el procedimiento de desplazamiento medio, la estimación actual del modo y_k en la iteración k se sustituye por su media ponderada localmente.

$$y_{k+1} = y_k + m(y_k) = \frac{\sum_i x_i G(x - x_i)}{\sum_i G(x - x_i)}$$

Los núcleos derivados correspondientes $g(r)$ son una bola unitaria y otra gaussiana, respectivamente. El uso del núcleo de Epanechnikov converge en un número finito de pasos, mientras que el núcleo gaussiano tiene una trayectoria más suave (y produce mejores resultados), pero converge muy lentamente cerca de un modo.

La forma más sencilla de aplicar el desplazamiento medio es iniciar una estimación de modo de desplazamiento medio independiente y en cada punto de entrada x_i e iterar para un número fijo de pasos o hasta que la magnitud del desplazamiento medio esté por debajo de un umbral. Un enfoque más rápido es submuestrear aleatoriamente los puntos de entrada x_i y realizar un seguimiento de la evolución temporal de cada punto. Los puntos restantes se pueden clasificar en función de la trayectoria de evolución más cercana. Paris y Durand revisan una serie de otras implementaciones más eficientes del desplazamiento medio, incluyendo su propio enfoque, que se basa en el uso de una estimación eficiente de baja resolución del espacio multidimensional completo de $f(x)$ junto con sus puntos estacionarios.

Por lo general, se pueden obtener mejores resultados mediante la agrupación en el dominio conjunto del color y la ubicación. En este enfoque, las coordenadas espaciales de la imagen $x_s = (x, y)$, que se denominan dominio espacial, se concatenan con los valores de color x_r , que se conocen como el dominio de rango, y el clustering de desplazamiento medio se aplica en este espacio de cinco dimensiones x_j . Dado que la ubicación y el color pueden tener diferentes escalas, los núcleos se ajustan en consecuencia, es decir, usamos un núcleo de la forma:

$$K(x_j) = k\left(\frac{\|x_r\|^2}{h_r^2}\right)k\left(\frac{\|x_s\|^2}{h_s^2}\right)$$

Donde se utilizan parámetros separados h_s y h_r para controlar los anchos de banda espaciales y de rango de los núcleos de filtro.

La diferencia entre el desplazamiento medio y el filtrado bilateral, sin embargo, es que en el desplazamiento medio las coordenadas espaciales de cada píxel se ajustan junto con sus valores de color, de modo que el píxel migra más rápidamente hacia otros píxeles con colores similares y, por lo tanto, se puede usar más adelante para la agrupación en clústeres y la segmentación.

Determinar los mejores parámetros de ancho de banda h para usar con el desplazamiento medio sigue siendo algo así como un arte, aunque se han explorado varios enfoques. Estos incluyen la optimización del equilibrio sesgo varianza, la búsqueda de rangos de

parámetros donde el número de clústeres varía lentamente, la optimización de algún criterio de agrupación en clústeres externos o el uso de conocimientos de arriba hacia abajo (dominio de aplicación). También es posible cambiar la orientación del núcleo en el espacio de parámetros conjuntos para aplicaciones como segmentaciones espaciotemporales.

El cambio medio se ha aplicado a una serie de problemas diferentes en la visión por computadora, incluido el seguimiento de la cara, la extracción de formas 2D y la segmentación de texturas, y más recientemente en el emparejamiento estéreo, la representación no fotorrealista y la edición de video. Paris y Durand proporcionan una buena revisión de tales aplicaciones, así como técnicas para resolver de manera más eficiente las ecuaciones de cambio de media y producir segmentaciones jerárquicas [1].

Conclusión

A través de este informe se mencionaron diferentes métodos de segmentación para el mejoramiento de imágenes, a través de la investigación se comprendió de mejor manera dichos métodos y se pudo identificar posibles áreas de aplicación.

El mejoramiento de imágenes es importante en la visión por computadora ya que es a través del mejoramiento que se pueden tratar imágenes para cumplir ciertos requerimientos.

Bibliografía

1. SZELISKI, R. (2020). COMPUTER VISION (pp. 267-304). [Place of publication not identified]: SPRINGER NATURE.
2. La Serna Palomino, N., & Román Concha, U. (2021). Técnicas de Segmentación en Procesamiento Digital de Imágenes [Ebook]. Retrieved from https://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/risi/2009_n2/v6n2/a02v6n2.pdf
3. METODOS DE SEGMENTACION BASADOS EN EL ANALISIS DEL HISTOGRAMA. (2021). Retrieved 21 June 2021, from http://www.varpa.org/~mgpenedo/cursos/Ip/Tema6/nodo6_2.html
4. Sugrañes, L. (2021). Imagen binaria. Retrieved 21 June 2021, from <http://www.dimages.es/Tutorial%20A.l/segmentacion/binaria.htm>