Una Comparación Entre Métodos de Segmentación en Imágenes de Escenas de Interiores de Inmuebles

Salvador Cervantes Álvarez y Raúl Pinto Elías

Departamento de Ciencias Computacionales
Centro Nacional de Investigación y Desarrollo Tecnológico
[scervantes, rpinto]@cenidet.edu.mx
Paper received on 09/08/10, Accepted on 20/09/10

Resumen. En este trabajo se presenta una comparación perceptual entre las segmentaciones producidas por los algoritmos de segmentación Mean Shift, gPb-owt-ucm y la técnica Basada en Grafos de Felzenszwalb aplicados en imágenes de escenas de interiores de inmuebles no controladas. La comparación del desempeño de los algoritmos se realizó considerando los siguientes aspectos: sobre segmentación (partición de una región homogénea en 2 o más regiones), segmentación burda (unión de 2 o más regiones no homogéneas dentro de una sola región) y segmentación consistente. Los algoritmos fueron aplicados sobre las colecciones de imágenes de [17] y PASCAL 2007 [20].

Palabras clave: Segmentación, escenas de interiores.

1 Introducción

El objetivo de las técnicas de segmentación es obtener una partición de la imagen en regiones con apariencia homogénea dada una cierta medida de similitud aplicada sobre un conjunto de descriptores. Varias técnicas de segmentación actuales ([2], [4], [5], [9]) intentan segmentar la imagen en regiones homogéneas que contengan un significado perceptual, donde las regiones aíslen a un objeto o partes de objetos presentes en una imagen (por ejemplo, el respaldo o el descansa brazos de una silla de escritorio).

La segmentación es una etapa indispensable en varios modelos que utilizan información de alto nivel ([1], [3], [10], [11], [18], [19]), sin embargo, la tarea de obtener una segmentación fiable sigue siendo un problema desafiante, y el buscar un algoritmo de segmentación con el mejor desempeño ha sido el objetivo de varios trabajos que realizan estudios comparativos entre los algoritmos de segmentación más ampliamente usados en el análisis de imágenes de escenas ([7], [8], [13], [14]); estos estudios son realizados sobre la colección de imágenes Berkeley [14], la cual, está compuesta principalmente por imágenes de escenas de exteriores. En la literatura no se han encontrado comparaciones de desempeño, realizadas con imágenes de escenas de interiores, motivo por el cual se desarrolló la actual investigación.

El resto de este documento está estructurado de la siguiente forma: en la Sección 2 se explica el funcionamiento de los tres algoritmos de segmentación analizados; en la Sección 3 se describen las características de las imágenes de escenas de interiores de las colecciones de imágenes de [17] y [20] utilizadas en el experimento; en la Sección 4 se presentan los resultados obtenidos con los métodos de segmentación bajo estudio y en la Sección 5 se presentan las con-

Advances in Intelligent and Information Technologies.
Research in Computing Science 50, 2010, pp. 213-222

clusiones así como las sugerencias para obtener una comparación más precisa de los resultados.

2 Algoritmos de segmentación de escenas

Los algoritmos de segmentación comparados, fueron seleccionados debido a que son los más ampliamente usados en aplicaciones recientes, proporcionan un desempeño razonable ([2], [4], [9]) y sus implementaciones están disponibles públicamente. A continuación se explica en forma general el funcionamiento de cada uno de los métodos de segmentación analizados.

2.1 Mean Shift

En el método de [4], los píxeles de una imagen son descritos en un espacio de características multidimensional; la implementación utiliza el espacio de color CIE L*u*v*. El algoritmo busca las características que corresponden a las regiones más densas en el espacio de características, es decir a los cluster que corresponden a regiones en la imagen. El objetivo del análisis del espacio de características multidimensional, es delinear estos clusters.

El método Mean Shift se ejecuta en forma recursiva y converge hacia el punto con mayor densidad de acuerdo a una *función de densidad fundamental* detectando las *modas* de las densidades, para lo cual, se utiliza la técnica de ventana de Parzen [6] como kernel. Se comienza analizando una región inicial (ventana) y la media local es cambiada hacia la subregión en la que reside la mayoría de los puntos, de esta forma el vector Mean Shift puede definir el camino que lleva a puntos estacionarios de *densidad estimada*. El procedimiento Mean Shift ejecutado en forma sucesiva, calcula el vector Mean Shift y la translación del kernel (ventana) y garantiza la convergencia hacia puntos cercanos estacionarios (ver figura 1).

En [4], se realiza una teselación con ventanas iniciales en el espacio de características y se realiza una ejecución en forma paralela del método Mean Shift para cada ventana. Todos los puntos en el espacio de características evaluados por ventanas que convergen hacia un mismo punto estacionario, pertenecen a una misma región.

A la representación de la imagen en el espacio de características CIE L*u*v*, se incorporan las coordenadas de un píxel, generando una representación de *dominio mixto*. El *dominio mixto* está compuesto por el *dominio espacial* (coordenadas x y y) y el *dominio de rango* (espacio CIE L*u*v*), para los cuales se utilizan kernels por separado y se define el ancho de la ventana de Parzen h_s y h_r, para el domino espacial y de rango respectivamente. La selección del ancho del kernel no es algo trivial ya que un valor muy grande de h puede llevar a obtener una segmentación densa y un valor pequeño de h puede generar una sobre segmentación.

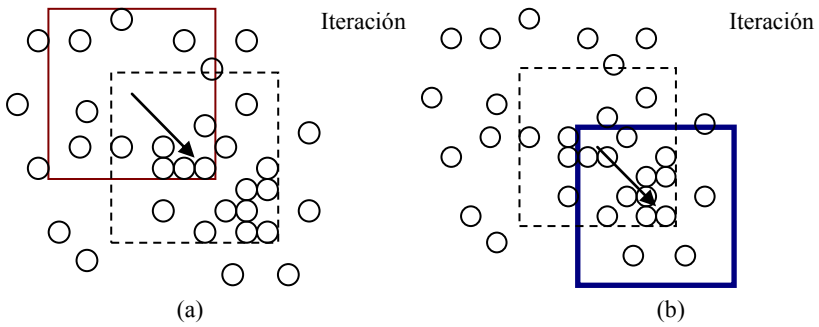


Figura 1. En la figura (a) se muestra el resultado del desplazamiento de la ventana de Parzen en la iteración 1 a partir de una posición inicial hacia una posición intermedia donde se encontró la mayor densidad en el espacio analizado. En la figura (b) se muestra el resultado de la iteración 2 donde a partir de una posición intermedia se alcanza el punto estacionario local.

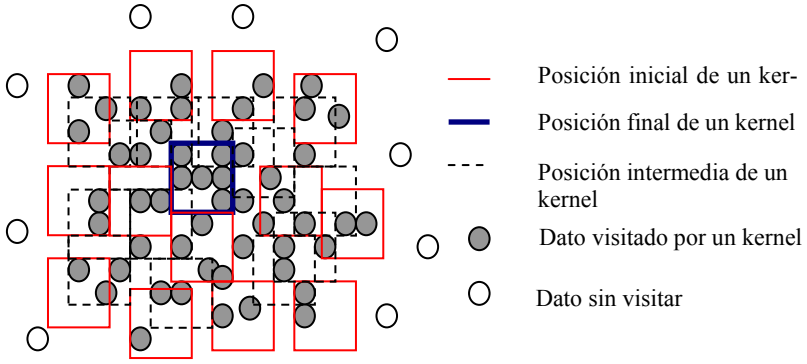


Figura 2. Se presenta una teselación del espacio de características. Todos los puntos que son visitados por ventanas de Parzen que convergen hacia el mismo punto pertenecen a una misma región.

2.2 Técnica basada en grafos de Felzenszwalb

El método de [9] considera a una imagen como un grafo dirigido y cada uno de sus píxeles son vértices conectados con sus píxeles adyacentes (considerando una vecindad cuatro u ocho píxeles) a través de aristas que miden la disimilaridad que existe entre los píxeles. Una descripción formal del método es: Sea $G = (V, E)$ un grafo no dirigido con vértices $u_i \in V$ (el conjunto de elementos a segmentar) y las aristas $(u_i, u_j) \in E$ correspondan a pares de vértices vecinos. Cada arista $(u_i, u_j) \in E$ tiene su correspondiente peso $w((u_i, u_j))$, el cual, es una medida no negativa de la disimilaridad entre los elementos vecinos u_i y u_j. Los elementos de V son píxeles de la imagen y los pesos w de cada arista es alguna medida de disimilaridad entre dos píxeles conectados por dicha arista.

En el enfoque basado en grafos, una segmentación S es una partición de V dentro de varios componentes, tal que cada componente (o región) $C \in S$ corresponde a componentes conectados en un grafo $G' = (V, E')$, donde $E' \subseteq E$. Cualquier segmentación es inducida por un subconjunto de aristas en E. En la segmentación, las aristas de dos vértices en el mismo componente tienen pesos relativamente bajos, y las aristas entre vértices en diferentes componentes tienen pesos altos.

Para realizar la segmentación, se tiene un predicado D, que compara las diferencias entre componentes (regiones) con las diferencias dentro de cada componente y es por lo tanto adaptativo a las características locales de los datos. La *diferencia interna de un componente* $C \in V$, es el peso más grande en el *árbol de expansión mínimo* (MST por sus siglas en inglés) de el componente, MST(C, E). Esto es:

$$Int(C) = \max_{e \in MST(C,E)} w(e) \tag{1}$$

La *diferencia entre dos componentes* C_1, $C_2 \in V$ es la arista de menor peso que conecta los dos componentes. Esto es:

$$Dif(C_1, C_2) = \min_{u_i \in C_1, u_j \in C_2, (u_i, u_j) \in E} w((u_i, u_j)) \tag{2}$$

Si no hay arista conectado a C_1 y C_2 se estable $Dif(C_1,C_2) = \infty$. El predicado de comparación de regiones D evalúa si existe evidencia para un límite entre un par de componentes revisando si la diferencia entre los componentes, $Dif(C_1,C_2)$, es relativamente grande con respecto a la diferencia interna dentro de al menos uno de los componentes, $Int(C_1)$ y $Int(C_2)$. Una función de umbral es usada para controlar el grado al cual la diferencia entre componentes debe ser más grande que la diferencia mínima interna. El predicado de comparación es definido como:

$$D(C_1, C_2) = \begin{cases} verdad & Si \quad Dif(C_1, C_2) > MInt(C_1, C_2) \\ falso & deotraforma \end{cases} \tag{3}$$

Donde la diferencia mínima interna, *MInt*, es definida como:

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)) \tag{4}$$

Donde la función de umbral τ controla el grado a el cual la diferencia entre dos componentes debe ser más grande que las diferencias internas con el objetivo de que exista evidencia de la existencia de un límite entre ellos (que D sea verdadero). Sin embargo, para pequeños componentes, $Int(C)$ no es una buena estimación de las características locales de los datos. En el caso extremo, cuando $|C| = 1$, $Int(C) = 0$. Por lo tanto, se usa una función de umbral basada en el tamaño del componente:

$$\tau(C) = k / | C | \tag{5}$$

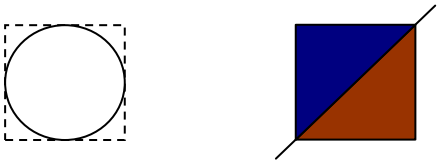
Donde $|C|$ denota el tamaño de C, y k es un parámetro constante. Para pequeños componentes se requiere de una fuerte evidencia para establecer la existencia de un límite. En la práctica k establece una escala de observación, en la

que valores grandes de *k* causan una preferencia por componentes grandes. Sin embargo, *k* no es un tamaño de componente mínimo.

2.3 gPb-owt-ucm

En [2] se genera una segmentación jerárquica de una imagen partiendo de los contornos de la imagen, para lo cual se utiliza el método para obtener de contornos de *probabilidad global de límites* (*gPb*), aunque puede utilizarse cualquier otro método para obtener los contornos. A partir de los contornos se aplica la *transformada watershed orientada* (*owt*) para producir un conjunto de regiones iniciales, con las cuales se construye un *mapa de contornos ultra métrico* (*ucm*). La secuencia de operaciones *owt-ucm* produce un árbol de regiones jerárquico.

El método *gPB* [12] está basado en el detector de contornos *Pb* [13], en el cual se calcula una señal de gradiente orientada $G(x, y, \theta)$ a partir de una imagen de intensidades *I*. El cálculo se realiza mediante colocar un disco en la posición (x, y) y dividirlo en dos mitades en el ángulo θ, aplicado cada uno de los canales del espacio de color CIE Lab y utilizando textones para la textura.



(a) (b)

Figura 4. (a) El disco es reemplazado por una ventana que lo contiene. (b) División de la ventana en un ángulo θ.

En cada mitad del disco se calcula un histograma de intensidad de los píxeles de *I*. La magnitud del gradiente *G* en la posición (x, y) está definida por la distancia X^2 entre los histogramas de las dos mitades del disco. Se realizan tres convoluciones en los canales de brillo (canal *L*), color (canales *a* y *b*) y textura (mapa de textones) con tamaños de disco iguales a $[\sigma/2, \sigma, 2\sigma]$, donde σ es la escala del detector *Pb*.

Los resultados con los diferentes tamaños de disco y en los diferentes canales son combinados en forma lineal con (6).

$$mPb(x, y, \theta) = \sum_{s} \sum_{i} \alpha_{i,s} G_{i,\sigma(i,s)} \quad (x, y, \theta) \tag{6}$$

donde *s* es el índice de las escalas, *i* el índice de los canales de características y $G_{i,\sigma(i,s)}(x, y, \theta)$ mide la diferencia de histogramas en el canal *i* entre dos mitades de un disco de radio $\sigma(i, s)$ centrado en (x, y) y divididos en el ángulo θ. Tomando la repuesta máxima sobre las orientaciones, se obtiene una medida de la fuerza del límite en cada píxel.

$$mPb(x, y) = \max_{\theta} \{mPb(x, y, \theta)\} \tag{7}$$

Para obtener *gPb* se necesita una etapa de agrupamiento espectral, donde se construye una matriz de afinidad *W* donde el máximo valor de *mPb* a lo largo de una línea conecta dos puntos.

$$W_{ij} = \exp\left(- \max_{p \in ij} \{mPb(p)\} / \sigma \right) \tag{8}$$

Donde *ij* es un segmento de línea que conecta a *i* y *j*, y σ es una constante. Se define $D_{ii} = \Sigma_j W_{ij}$ y se resuelve para los eigenvectores generalizados $\{v_0, v_1, \ldots, v_k\}$ del sistema $(D - W)v = \lambda Dv$. Tratando cada eigenvector $\mathbf{v_k}$ como una imagen, se realiza una convolución con *filtros derivados direccionales Gaussianos* en múltiples orientaciones θ, obteniendo una señal $sPb_v(x, y, \theta)$. La información de los diferentes eigenvectores es combinada para obtener el componente espectral del detector de contornos *gPb*.

$$sPb(x, y, \theta) = \sum_{k=1}^{n} \frac{1}{\sqrt{\lambda_k}} . sPb_v(x, y, \theta) \tag{9}$$

La probabilidad global del límite es entonces escrita como una suma ponderada de las señales locales y espectrales:

$$gPb(x, y, \theta) = \sum_s \sum_i \beta_{i,s} G_{i,\sigma(i,s)} (x, y, \theta) + \gamma . sPb(x, y, \theta) \tag{10}$$

El método *gPb* produce contornos cerrados, lo cual es necesario para evitar que el método *owt* fusione regiones que están separadas por un límite. Los valores de $\beta_{i,s}$ y γ son aprendidos utilizando como entrenamiento las imágenes de la base de imágenes Berkeley.

(a) (b) (c)

Figura 5. Imagen original de [17] de 256x256 píxeles. (b) Contornos de la imagen. (c) Segmentación de la imagen.

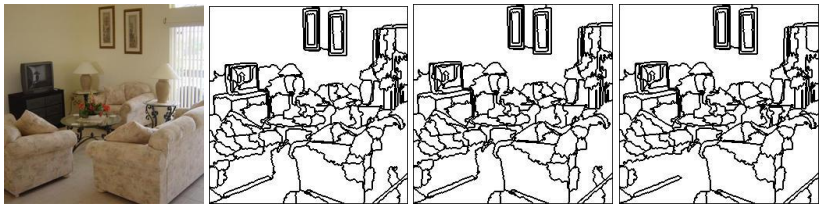
En la figura 5b se muestran los contornos cerrados obtenidos del procesamiento la imagen de la figura 5. Los contornos con mayor gradiente tienen una mayor intensidad y son representados en forma de árbol mediante el proceso *owt-ucm* en donde en el nivel más bajo del árbol (las hojas) se genera una sobre segmentación de la imagen equivalente a las regiones separadas por todos los contornos y en los niveles más altos del árbol se producen segmentaciones burdas (ver figura 5c) en donde los contornos con mayor gradiente segmentan la imagen.

3 Colecciones de imágenes de escenas de interiores

Las colecciones utilizadas para probar el desempeño de los algoritmos de segmentación en escenas de interiores son: imágenes de las categorías de sala, comedor y oficina de la colección generada en [17] e imágenes de interiores de la colección de entrenamiento PASCAL 2007 [20]. Las imágenes de las colecciones son no controladas y las dimensiones de las imágenes varían.

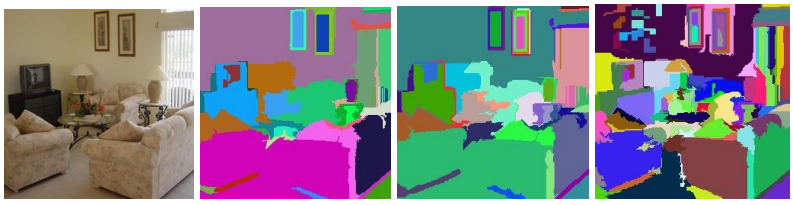
4 Resultados

Para mostrar el comportamiento de los métodos de Mean Shift y de Felzenszwalb, en las figuras 6 y 7 se presentan respectivamente resultados obtenidos con diferentes valores para sus parámetros, intentando evitar segmentaciones burdas. En ambos métodos el parámetro *mínimo* permite establecer la mínima cantidad de píxeles que debe tener una región, en caso de que la región sea menor es fusionada con la región adyacente con mayor similitud. En todos los casos se utilizó una máquina con 1.9 GHz y 3GB en RAM.



(a) (b) (c) (d)

Figura 6. Imagen original de [17] de 256x256 píxeles. (b) hs=1, hr=1.4, mínimo=160 (c) hs=8, hr=1.4, mínimo=160 (d) hs= 8, hr=1.5, mínimo=160.

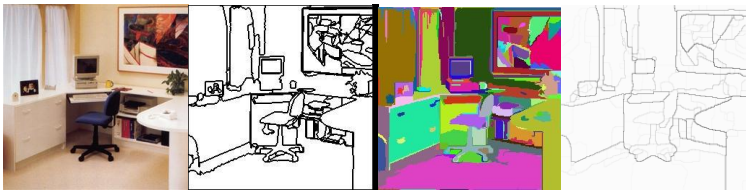


(a) (b) (c) (d)

Figura 7. Imagen original de [17] de 256x26 píxeles. (b) $\sigma = 0.8$, $k = 300$, mínimo=100. (c) $\sigma = 0.5$, $k = 300$, mínimo=100. (d) $\sigma = 0.5$, $k = 100$, mínimo=100.

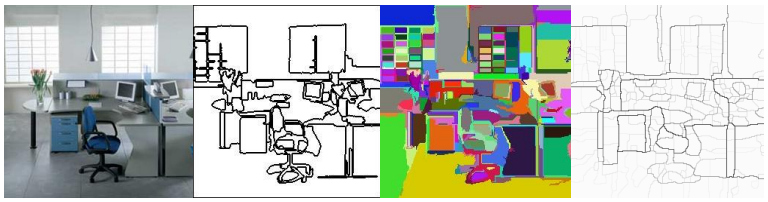
En el método Mean Shift por lo general se obtuvieron sobre segmentaciones de la imagen y en la figuras 6b y 6c se puede observar que se pueden obtener segmentaciones idénticas de la imagen con valores distintos para los parámetros *hs* y *hr*. A diferencia de Mean Shift el método de Felzenszwalb presentó una tendencia a generar segmentaciones burdas. En las figuras 8, 9, 10 y 11 se muestran los resultados de los tres métodos de segmentación; la imagen de la izquierda es la imagen original y las siguientes imágenes son los resultados de los métodos Mean Shift, Felzenszwalb y *gPb-owt-ucm* respectivamente. Se realizó

una selección manual de los parámetros de los métodos Mean Shift y Felzenszwalb intentando que produjeran segmentaciones consistentes; en el caso del método *gPb-owt-ucm* se utilizaron los valores de [2] para el tamaño de disco (σ=5 para el canal L y σ=10 para los demás canales).



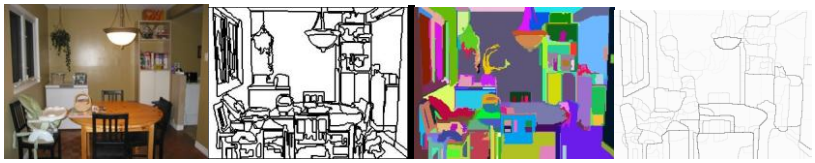
(a (b (c (d

Figura 8. (a) Imagen original de [17] de 256x256 píxeles. (b) hs=10, hr=9.2, mínimo=20. (c) σ =0.5, *k* =200, mínimo=20. (d) Contornos cerrados generados por el método *gPb-owt-ucm*.



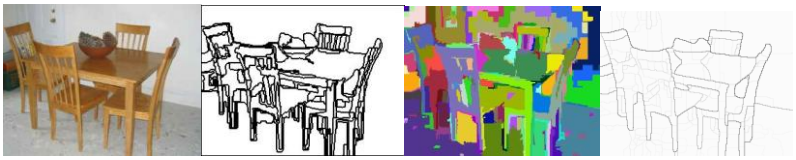
(a) (b) (c) (d

Figura 9. (a) Imagen original de [17] de 256x256 píxeles. (b) hs=4, hr=6.2, mínimo=100. (c) σ =0.5, *k* =200, mínimo=20. (d) Contornos cerrados generados por el método *gPb-owt-ucm*.



(a) (b) (c) (d)

Figura 10. (a) Imagen original de PASCAL 2007 [20] de 310x233 píxeles. (b) hs=1, hr=3.8, mínimo=120. (c) σ =0.5, *k* =200, mínimo=20. (d) Contornos cerrados generados por el método *gPb-owt-ucm*.



(a) (b (c) (d)

Figura 11. (a) Imagen original de [17] de 300x225 píxeles. (b) hs=10, hr=4.8, mínimo=80. (c) σ =0.5, *k* =200, mínimo=20. (d) Contornos cerrados generados por el método *gPb-owt-ucm*.

..El método gpb-owt-ucm requirió 1 hora 38 minutos para procesar una imagen de 321x481 píxeles siendo un método computacionalmente caro, mientras que los métodos Mean Shift y Felzenszwalb requirieron unos cuantos segundos, siendo este último el más rápido. Para lograr una segmentación consistente, los parámetros de los métodos Mean Shift y Felzenszwalb cambiaron en cada imagen, sin embargo, se mantuvo un comportamiento de sobre segmentación en Mean Shift y de segmentación burda en Felzenszwalb, mientras que los resultados de gPb-owt-ucm son dependientes de la elección del nivel del árbol (mediante la definición de un umbral), por tal razón solamente se muestra la imagen de los contornos en las figuras 8d, 9d, 10d y 11d.

5 Conclusiones

Siguiendo la idea de [2] es mejor obtener una sobre segmentación de la imagen y después fusionar las regiones en forma jerárquica basándose en otros descriptores y métricas. Tomando en cuenta lo anterior, la técnica de Felzenszwalb es la de peor desempeño, aunque es de los algoritmos más rápidos según [16].

El método *gPb-owt-ucm* es computacionalmente caro y la calidad de la segmentación depende de la elección del nivel del árbol generado por el proceso *owt-ucm*, sin embargo, tiene la ventaja de poder establecer relaciones de pertenencia entre regiones debido a su estructura de árbol, lo cual, puede ser útil para la representación de objetos como en [1]. Los métodos Mean Shift y Felzenszwalb pueden ser mejorados agregando descriptores de textura en el proceso de segmentación y en el caso del método Mean Shift se puede seguir un proceso de mezcla de regiones en la sobre segmentación, buscando generar una segmentación jerárquica como en [2].

Los resultados de la comparación pueden ser útiles para futuros trabajos de investigación en escenas de interiores, que requieran del empleo de un algoritmo de segmentación tomando en cuenta el tiempo de procesamiento requerido y el desempeño. Sin embargo, para poder obtener una evaluación más precisa del desempeño de los algoritmos de segmentación en imágenes de escenas de interiores, se debe realizar una comparación cuantitativa de los algoritmos de segmentación, para lo cual, se requerirá que varias personas realicen segmentaciones manuales de las imágenes como en la base de imágenes de Berkeley [14] y aplicar una medida cuantitativa a las segmentaciones resultantes como en [7], [8], [13], [14].

Referencias

1. Ahuja, N., Todorovic, S.: Connected Segmentation Tree – A Joint Representation of Region Layout and Hierarchy. Computer Vision and Pattern Recognition, (2008).
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From Contours to Regions: An Empirical Evaluation. Computer Vision and Pattern Recognition, (2009).
3. Christoudias, C., Georgescu, B., Meer, P.: Synergism in Low Level Vision. International Conference on Pattern Recognition, vol. 4, (2002).
4. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 603-619, (2002).
5. Cour, T., Benezit, F., Shi, J.: Spectral Segmentation with Multiscale Graph Decomposition. Computer Vision and Pattern Recognition, (2005).

6. Duda, R. O., Hart, P. E., Store, D. G.: Pattern Classification, Second Edition. Wiley, (2000).
7. Estrada, F, Jepson, A.: Quantitative Evaluation of a Novel Image Segmentation Algorithm. Computer Vision and Pattern Recognition, vol. 2, pp. 1132-1139, (2005).
8. Estrada, F, Jepson, A.: Benchmarking Image Segmentation Algorithms. International Journal of Computer Vision, vol. 85, pp. 167-181, (2009).
9. Felzenszwalb, P. F., Huttenlocher, D.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision, (2004).
10. Gu, C., Lim, J., Arbelaez, P., Malik, J.: Recognition using Regions. Computer Vision and Pattern Recognition, (2009).
11. Li, L., Socher, R., Fei-Fei, L.: Towards Total Scene Understanding: Classification, Annotation and Segmentation in an Unsupervised Framework, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2009).
12. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using Contours to Detect and Localize Junctions in Natural Images, Computer Vision and Pattern Recognition, (2008).
13. Martin, D., Fowlkes C., Malik, J.: Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues. IEEE Transactions on Pattern Analysis and Machine Learning, vol. 26, pp. 530-549, (2004).
14. Martin, D., Fowlkes C., Tal, D., Malik, J.: A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. On Proceedings of the International Conference on Computer Vision, vol. 2, (2001).
15. Meer, P. Georgescu, B.: Edge detection with embedded confidence. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 1351-1365, (2001).
16. Paris, S., Durand, F.: A Topological Approach to Hierarchical Segmentation Using Mean Shift. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, (2007).
17. Quattoni A., Torralba, A.: Recognizing Indoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition, (2009).
18. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in Context. International Conference on Computer Vision, (2007).
19. Russell, B. C., Efros, A. A., Sivic, J., Freeman, W.T., Zisserman, A.: Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. Computer Vision and Pattern Recognition, vol. 2, pp. 1605-1614, (2006).
20. Visual Object Classes Challenge, http://pascallin.ecs.soton.ac.uk/challenges/VOC/.