Detección de texto en imágenes digitales como estrategia para mejorar la recuperación de imágenes por contenido

Manuel Mejía-Lavalle[1], Mathias Lux[2], Carlos Pérez[1], Alicia Martínez[1]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor, México

[2] Klagenfurt University, Klagenfurt am Wörthersee, Austria

{mlavalle, carlospl, amartinez}@cenidet.edu.mx, mlux@itec.aau.at

Resumen. Los avances en los algoritmos para la detección de texto están permitiendo detectar de mejor manera el texto que aparece en imágenes digitales de escenas naturales. En particular la Transformada del Ancho del Trazo (*Stroke Widht Transform*) ha mostrado buenos resultados. Sin embargo, en el área de Recuperación de Imágenes Basada en Contenido, en donde se manejan características globales para eficientar el proceso de búsqueda, en general no se está usando la información de alto nivel del texto. Típicamente se está empleando la textura o el color para detectar las regiones de texto. En este trabajo se investiga el impacto de detectar texto usando características globales en beneficio de la Recuperación de Imágenes Basada en Contenido y se propone una estrategia. De la experimentación realizada se observa que, usando nuestra estrategia, se logra una mayor precisión (15%) en la recuperación de imágenes digitales.

Palabras clave: Recuperación de imágenes basada en contenido, características globales, detección de texto, stroke widht transform.

Digital Images Text Detection as Strategy to Improve Content-Based Image Retrieval

Abstract. Recent research advances in text detection allow us for finding text regions in natural scenes rather accurately. Global features in Content-Based Image Retrieval,however, typically do not cover such a high level information. While characteristics of text regions may be reflected by texture or color properties, the respective pixels are not treated in a different way. In this work we investigate the impact of text detection on Content-Based Image Retrieval using global features. Detected text regions are preprocessed to allow for different treatment by feature extraction algorithms, and we show that our strategy, for certain domains, leads to a much higher precision (15%) in Content-Based Retrieval.

Keywords: Content-based image retrieval, global features, text detection, stroke width transform.

1. Introducción

La detección de texto en escenas naturales ha tenido avances significativos en los últimos 5 años. El principal objetivo de los algoritmos de detección de texto es localizar el texto que aparece inmerso en una imagen digital. Normalmente lo que sigue después de la localización del texto es el reconocimiento del texto, es decir, convertir la imagen de texto en caracteres de texto (OCR por sus siglas del inglés *Optical Character Recognition*).

Uno de los mejores algoritmos recientes para detección de texto en imágenes es el conocido como Transformada del Ancho del Trazo ó *Stroke Widht Transform* (SWT) [1], el cual tiene la habilidad de encontrar texto en fotografías digitales independientemente del idioma en que esté escrito el texto. Para nuestra investigación la parte de OCR no es relevante. Lo que nos interesa es la detección del texto en el contexto de la Recuperación de Imágenes Basada en Contenido (CBIR por sus siglas del inglés *Content-Based Image Retrieval*), en donde se requiere procesar-buscar, a gran velocidad y en una inmensa cantidad de imágenes digitales, la o las imágenes más parecidas a la imagen que el usuario presenta al sistema como muestra. De esta manera, más que saber qué dice el texto inmerso en la imagen digital, nuestro interés se centra en localizar la región de la imagen en donde aparece el texto: su tamaño, posición, color y textura, que nosotros asumimos (como hipótesis) que permitirían mejorar la precisión, a bajo nivel, para la recuperación basada en contenido.

Para lograr lo anterior, nosotros pre-procesamos la imagen para:

(i) Obtener una máscara de la región en donde hay texto, aplicando un color homogéneo,

(ii) Obtener una máscara de la región complementaria, es decir, en donde no hay texto.

Después de esto se hace la indexación, usando características globales comúnmente empleadas en los sistemas CBIR y que está probado que funcionan bien en imágenes digitales de escenas naturales. Para realizar la evaluación de nuestra estrategia propuesta, usamos la base de datos de imágenes *SIMPLIcity* [2]. También experimentamos con la base de imágenes *Street View Text* [3], para tener en total 11 diferentes categorías de imágenes. En el presente artículo actualizamos las referencias y ampliamos los experimentos previamente presentados por nosotros mismos en [4], validando y ratificando con más casos experimentales los beneficios que se obtienen empleando la estrategia que proponemos y que siguen vigentes desde entonces.

El resto de este trabajo está organizado de la siguiente manera: en la Sección 2 presentamos el trabajo relacionado al tema de la estrategia que proponemos; en la Sección 3 describimos la estrategia propuesta; la Sección 4 detalla los experimentos realizados y los resultados obtenidos y la Sección 5 concluye y discute el trabajo a realizar en el futuro inmediato.

2. Trabajo previo relacionado

Según [5] y [6] existen dos grupos principales de enfoques para la detección de texto en imágenes digitales fijas, los basados en:

(i) Textura y
(ii) Regiones.

Los algoritmos del primer grupo tratan la imagen a diferentes escalas y generalmente son costosas en términos de recursos computacionales de tiempo y memoria. Los algoritmos pertenecientes al segundo grupo se basan en las propiedades características a nivel *pixel*, como lo son un color constante, grupos de *pixeles*, etc. Nuestra investigación está más relacionada con el este segundo grupo.

En la literatura especializada podemos encontrar numerosos enfoques publicadas a lo largo de los años [5, 6, 7, 8, 9]. Por ejemplo, en [1] se propone el algoritmo SWT para la detección de texto. Ahí se emplea la idea de encontrar los trazos (*strokes*) que constituyen una letra. Primero se detectan los bordes y luego se trazan líneas a los bordes vecinos. Si se encuentra un borde y la dirección del gradiente coincide en este punto con la dirección gradiente del borde original y además la distancia permanece estable a lo largo del borde, se asume que los dos bordes son los límites de un trazo de un cierto grosor o anchura. Después que se tiene identificada una letra candidata, se procede a identificar grupos de letras que forman palabras y, sobre todo esto, finalmente se dibuja una caja que delimita el texto así localizado. Se ha reportado que SWT se desempeña bien en escenas naturales que contienen texto, con la interesante característica extra de ser independiente al lenguaje en que esté escrito el texto (lo cual por cierto no es el caso del método descrito en [10]). En [11] se propone un método para detectar texto con orientaciones más bien arbitrarias en escenas naturales consideradas complejas. El enfoque de los autores es similar al presentado en [1], con la diferencia de que cada *pixel* dentro de un trazo es asignado al haz de anchura del trazo. En este caso el proceso de identificación de una letra se basa en la consistencia de la anchura del trazo.

Algunos grupos de investigación han demostrado la utilidad de la detección de texto para mejorar los procesos de los sistemas CBIR. En [12] los autores proponen una nueva técnica para usar áreas de texto en imágenes que van a ser procesadas por un sistema CBIR. Básicamente emplean el algoritmo antes presentado en [13] y cuyos resultados tienen los mismos valores de las métricas de *presicion* y *recall* que los obtenidos usando SWT; ellos extraen el centro de la caja que confina a la o las letras detectadas, obteniendo además la escala y la orientación, para crear así un vector de características para que el sistema CBIR funcione mejor. En [14] otro sistema CBIR se propone y se usa la idea de identificar mediante la segmentación y búsqueda del componente (es decir, el caracter ó letra en cuestión), aplicando propiedades geométricas de los componentes detectados para crear así vectores de características.

En el presente trabajo nosotros empleamos SWT [1] para identificar regiones de texto. Con estas regiones ya localizadas, nos interesa investigar la influencia que se tendría al usar características globales, como las conocidas con el nombre de *Pyramid Histogram of Oriented Gradientes* (PHOG) [15], *Auto Color Correlogram* (ACC) [16], *Fuzzy Color and Texture Histogram* (FCTH) [17] y el *Joint Composite Descriptor* (JCD), que también es descrito en [17].

A continuación detallaremos la estrategia que nosotros proponemos y con la cual se desea mejorar el rendimiento de los sistemas tipo CBIR.

3. La estrategia propuesta

Nuestra estrategia usa cuatro características globales ampliamente conocidas en el área de los sistemas CBIR y que se sabe tienen buen desempeño de manera individual (en [18] el lector interesado puede consultar un estudio comparativo). Estas características globales son, como ya se había mencionado anteriormente: *Pyramid Histogram of Oriented Gradientes* (PHOG) [15], *Auto Color Correlogram* (ACC) [16], *Fuzzy Color and Texture Histogram* (FCTH) [17] y el *Joint Composite Descriptor* (JCD) [17].

Para extraer las características globales de las imágenes nosotros usamos LIRE [18], que es una librería *open-source* desarrollada explícitamente para evaluar sistemas CBIR. A su vez la librería LIRE está basada en una máquina de búsqueda robusta, veloz y bien conocida en el medio llamada *Lucene*. La principal aportación de LIRE consiste en contener una amplia gama de algoritmos de características globales, como PHOG, ACC, FCTH y JDC, entre otros muchos.

En particular nosotros implementamos SWT en lenguaje Java, usando métodos disponibles en LIRE, tales como *Canny Edge Detector* (CED) y el cálculo del gradiente de magnitud y dirección. Para la implementación seguimos lo mencionado por los autores iniciales del SWT [1]. El umbral para considerar válido un haz en un trazo lo fijamos en $\pi/2$; el tamaño mínimo de una letra lo situamos en los 10 *pixeles* y el máximo en 300 *pixeles*; la cantidad mínima para considerar que existe un texto son dos letras juntas que tengan el mismo color, con una distancia entre letras que no exceda tres veces el tamaño de la letra y cuyo grosor de trazo no sea el doble con respecto a otra(s) letra(s).

Para verificar que nuestra implementación de SWT estuviera arrojando resultados correctos, realizamos una inspección visual sobre las mismas imágenes usadas en este artículo. Nuestra implementación previamente validada de SWT ya forma parte de LIRE (https://code.google.com/p/lire/).

Para realizar la evaluación de nuestra estrategia usamos dos bases de imágenes. Lo anterior porque consideramos que es necesario probar SWT con imágenes que tienen bastantes regiones de texto y a la vez que han sido usadas para sistemas CBIR.

La base de imágenes *SIMPLIcity* [2] es bastante conocida y usada en el contexto de los sistemas CBIR. Por otro lado, la base de imágenes *Street View Text* [3] fue concebida pensando en experimentar y evaluar algoritmos especializados en la detección de textos. Con estas dos bases de imágenes creamos 11 categorías de imágenes. Luego seleccionamos las primeras 100 imágenes de *Street View Text* para mantener la consistencia con *SIMPLIcity*, cuya cada categoría está compuesta por 100 imágenes. Al final creamos y obtuvimos una base para nuestra experimentación con 1,100 imágenes. En la Figura 1 se muestran ejemplos de las imágenes con las regiones de texto enmascaradas o cubiertas por rectángulos en color negro. Por el contrario, en la Figura 2 se muestran imágenes de ejemplo en donde las regiones en donde no hay texto son cubiertas de color negro. Este pre-procesamiento de enmascaramiento se

realizó sólo sobre las imágenes en donde SWT detectó texto; aquellas imágenes en donde SWT no localizó texto alguno, se dejaron sin cambios.

Fig. 1. Imágenes de ejemplo donde el texto es enmascarado en negro por SWT

Para evaluar objetivamente la capacidad de recuperación de imágenes de nuestra estrategia usamos la métrica conocida como *P10* (*Precision-at-10*) cuya definición matemática es:

$$Pk = 1/k \sum r(Xn),$$

donde *X1, X2, X3 ...Xn* son los resultados según los ordena el método CBIR y *r(Xn) = 1 si Xn es relevante y 0 de otra manera*.

De esta manera cada imagen de cada categoría es considerada como una consulta o *query*; es decir, al final tendremos 1,100 consultas posibles. Para cada consulta verificamos si las primeras 10 imágenes (*k = 10*) recuperadas pertenecen a la categoría respectiva. La interpretación entonces de un *Pk = 1* equivale a un resultado (recuperación de imágenes) perfecto, mientras que *Pk = 0* indica que ninguna de las 10 primeras imágenes recuperadas pertenecen a la categoría que se mostró inicialmente como ejemplo.

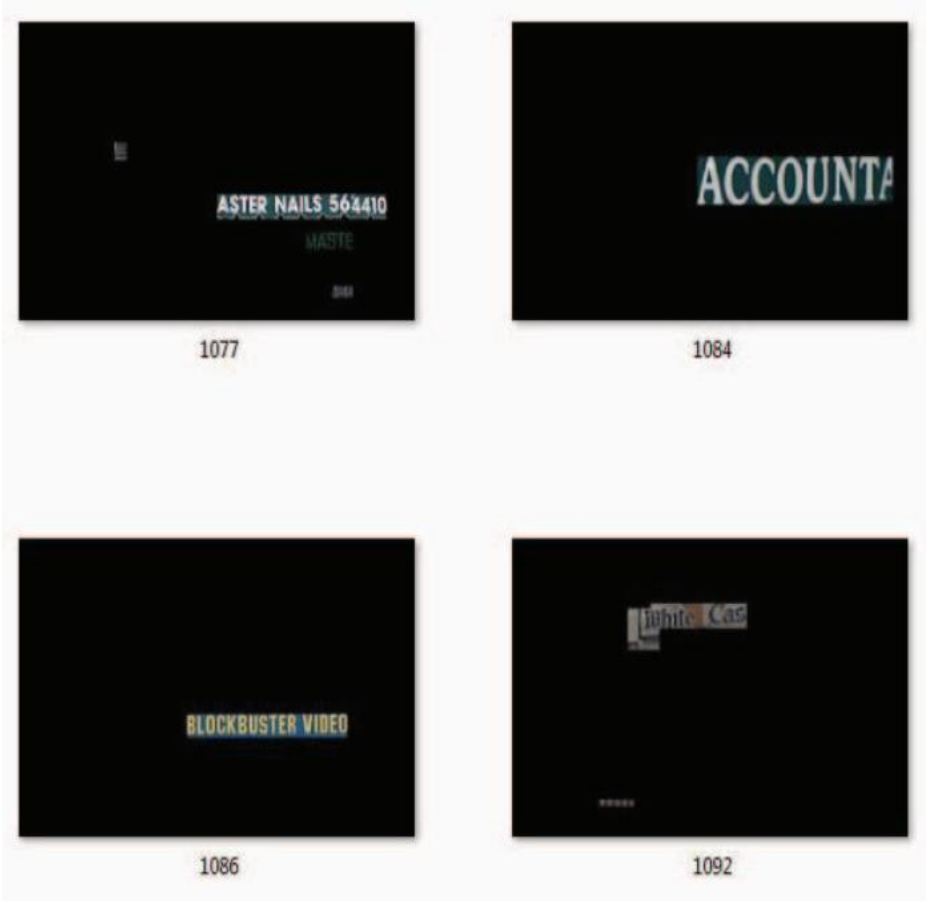

Fig. 2. Imágenes de ejemplo donde el texto no es enmascarado en negro por SWT, pero sí el resto de la imagen

4. Experimentación y resultados

En nuestra experimentación calculamos el valor *P10* para cada una de las 1,100 imágenes antes descritas y las promediamos por categoría. Puesto que nuestro objetivo inicial era investigar el impacto de la detección de texto cuando se usan las características globales (PHOG, ACC, FCTH y JDC) creamos dos conjuntos de imágenes de prueba: con el texto enmascarado en negro y con el resto de la imagen enmascarado en negro, dejando el texto visible (Figuras 1 y 2). Aún más, creamos tres versiones adicionales de prueba: las imágenes originales *D0*, las imágenes con el texto enmascarado *Dm* (Figura 1) y finalmente el complemento *Dt* (Figura 2). Las imágenes en donde SWT no detectó texto alguno, se dejaron sin aplicar ninguna máscara.

En las gráficas mostradas en las Figuras 3 a 5 las barras muestran el promedio de *P10* para cada una de las características globales de izquierda a derecha ACC, FCTH, JCD y PHOG respectivamente. La Figura 3 muestra los resultados sin detección de

texto *D0* y son los resultados de base para la comparación. La Figura 4 muestra el caso *Dm*, donde se observa una mejoría con respecto a *D0*; y la Figura 5 el caso *Dt*, donde se aprecian resultados no tan buenos como con *Dm*.

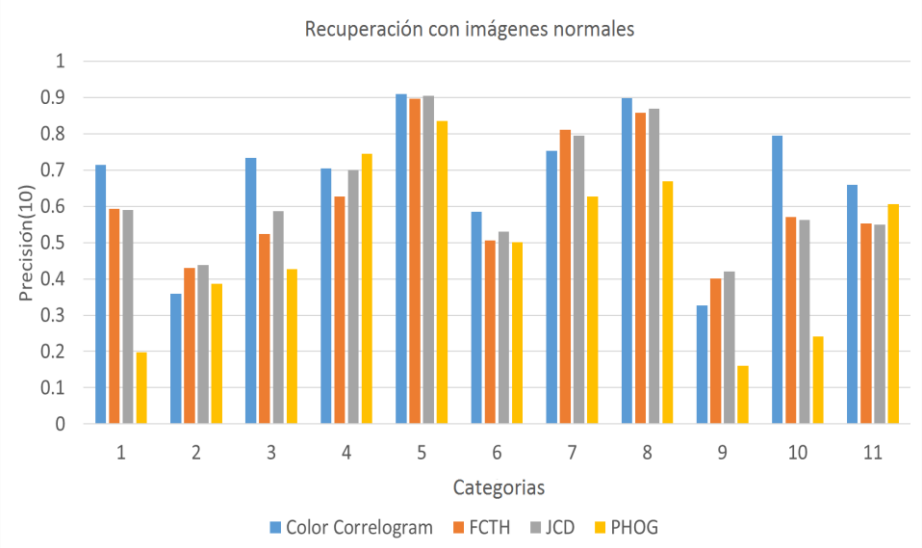

Fig. 3. Resultados cuando no se aplica pre-procesamiento alguno

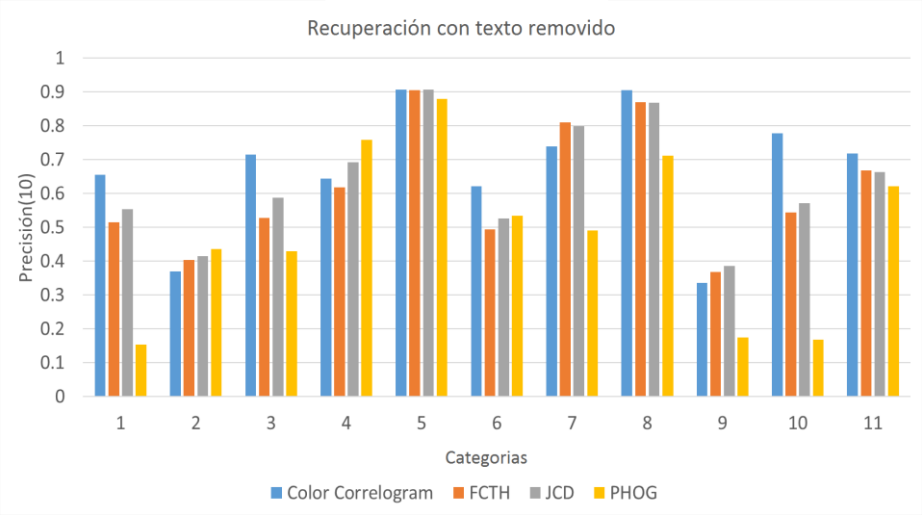

Fig. 4. Resultados cuando se aplica pre-procesamiento *Dm*

En la Tabla 1, a manera de resultados a detalle, se muestran los valores de *P10* de la categoría 11 usando las tres variantes de la base de imágenes. Se observa que cuando se incluye la detección de texto enmascarando el texto, se incrementa la precisión considerablemente para todos los cuatro extractores o descriptores de características globales investigados.

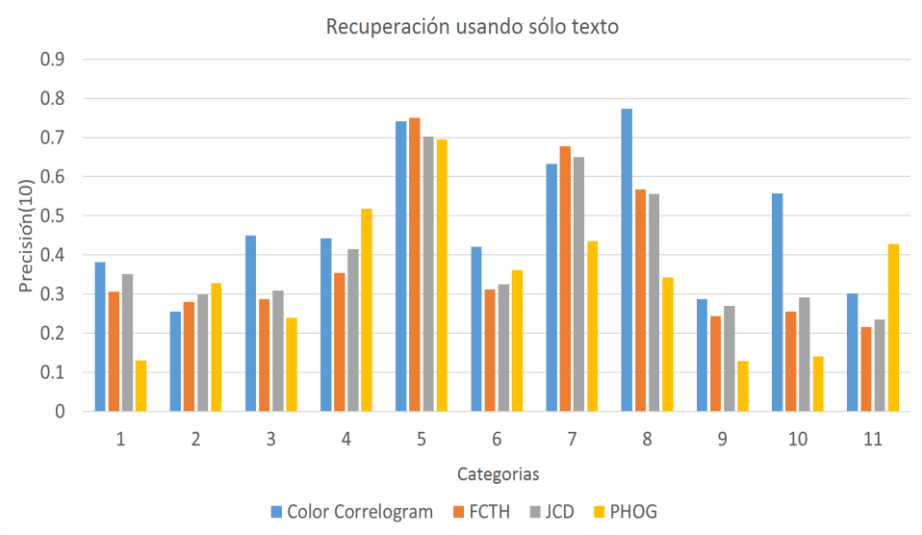

Fig. 5. Resultados cuando se aplica pre-procesamiento *Dt*

Tabla 1. Promedio de *P10* para la categoría 11, los mejores valores están en negritas

P10	**Variante**	**Descriptor**			
		ACC	JCD	PHOG	FCTH
Data Set	*D0*	0.658	0.550	0.605	0.552
	Dm	**0.717**	**0.664**	**0.621**	**0.668**
	Dt	0.302	0.235	0.428	0.215

En la Tabla 2 se muestra en porcentaje la mejoría promedio lograda por *Dm* comparada contra *D0*, obteniéndose al final una mejoría global de casi el 15%.

Tabla 2. Mejoría en porcentaje lograda por *Dm* en relación a *D0*

% mejor		**Descriptor**			
		ACC	JCD	PHOG	FCTH
Data Sets	*Dm* vs *D0*	+9	+21	+3	+21

5. Conclusiones y trabajo futuro

Presentamos una estrategia distinta para mejorar la recuperación de imágenes basada en contenido y que se beneficia de la detección de texto en las imágenes digitales por medio de SWT y cuatro descriptores globales comúnmente usados en los sistemas CBIR.

En nuestra experimentación calculamos el valor *P10* para cada una de las 1,100 imágenes de prueba y las promediamos por cada una de las 11 categorías que definimos.

Puesto que nuestro objetivo inicial era investigar el impacto de la detección de texto cuando se usan las características globales (PHOG, ACC, FCTH y JDC) creamos dos conjuntos de imágenes de prueba: con el texto enmascarado en negro y con el resto de la imagen enmascarado en negro, dejando el texto visible. Así creamos tres versiones adicionales de prueba: las imágenes originales *D0*, las imágenes con el texto enmascarado *Dm* y finalmente el complemento *Dt*.

Encontramos que la estrategia propuesta funciona bien (15% en promedio mejor) cuando el texto detectado es enmascarado en negro. En general, los trabajos del estado del arte revisados que buscan mejorar a los sistemas de recuperación de imágenes por contenido, logran mejorías por debajo de lo que nosotros logramos, observándose que aún mejorías inferiores al 10% ya son consideradas como más que aceptables por los autores de estos trabajos. Los resultados y mejoría alcanzada por nosotros creemos que significan una importante contribución al área, pues se pueden ver beneficiados dominios como la geo-localización, la búsqueda de instrumental especializado o la clasificación de escenas, por citar tan sólo tres casos en donde los sistemas CBIR son relevantes en el día a día.

Como trabajo futuro deberemos experimentar con más bases de imágenes y otros detectores globales. También se considera hacer pruebas con otros detectores de bordes, como se hizo en [19], que pudieran mejorar aún más los resultados reportados en el presente artículo. Algo interesante a experimentar es crear un algoritmo de fusión de los métodos basados en el procesamiento de la imagen y en combinación con la caracterización del texto detectado.

Referencias

1. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 2963–2970 (2010)
2. Wang, J.: *SIMPLIcity*: Semantics-sensitive Integrated Matching for Picture Libraries, 1 Introduction. Pattern Analysis and Machine Intelligence, Vol. 23, No. 9, pp. 947–963 (2001)
3. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: Computer Vision, IEEE International Conference on, pp. 1457–1464 (2011)
4. Perez, C., Lux, M., Mejia-Lavalle, M.: Toward Improving Content-Based Image Retrieval Systems by means of Text Detection. In: Mechatronics, Electronics and Automotive Engineering, IEEE Int. Conf on, pp. 50–53 (2014)
5. Gonzalez, A., Bergasa, L.M., Yebes, J.J., Bronte, S.: Text location in complex images. In: Pattern Recognition (ICPR), 21st International Conference on, IEEE (2012)
6. Li, Y., Lu H.: Scene text detection via stroke width. In: Pattern Recognition (ICPR), 21st International Conference on, IEEE (2012)
7. Lin, Z., Wu, Y., Zhao, Z., Fang, C.: A robust hybrid method for text detection in natural scenes by learning-based partial differential equations. Neurocomputing, Vol. 168, pp. 23–34 (2015)
8. Zhang, H., Zhao, K., Song, Y., Guo, J.: Text extraction from natural scene image: a survey. Neurocomputing, Vol. 122, pp. 310–323 (2013)
9. Ye, Q., Doermann, D.: Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. Vol. 37, No. 7, pp. 1480–1500 (2015)
10. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Computer Vision and Pattern Recognition (CVPR) IEEE Conference on, pp. 3538–3545 (2012)

11. Yi, C., Tian, Y.: Text string detection from natural scenes by structure-based partition and grouping. Image Processing, IEEE Transactions on, Vol. 20, No. 9, pp. 2594–2605 (2011)
12. Tsai, S., Chen, H., Chen, D., Parameswaran, V., Grzeszczuk, R., Girod, B.: Visual text features for image matching. In: Multimedia (ISM), IEEE International Symposium on, pp. 408–412 (2012)
13. Chen, H., Tsai, S.S., Schroth, G., Chen, D.M., Grzeszczuk, R., Girod, B.: Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In: Macqand, B., Schelkens, P. (eds), ICIP, pp. 2609–2612, IEEE (2011)
14. Nigam, A., Garg, A.K., Tripathi, R.: Content based trademark retrieval by integrating shape with colour and texture information. International Journal of Computer Applications, Vol. 22, No. 7, pp. 40–45 (2011)
15. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM, International Conference on Image and Video Retrieval, CIVR '07, pp. 401–408, New York, NY, USA ACM (2007)
16. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), Washington, DC, USA IEEE Computer Society (1997)
17. Chatzichristofis, S., Boutalis, Y.: Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In: Gasteratos, A., Vincze, M., Tsotsos, J. (eds), Computer Vision Systems, Vol. 5008 of Lecture Notes in Computer Science, pp. 312–322, Springer Berlin Heidelberg (2008)
18. Lux, M.: Lire: Open source image retrieval in Java. In: ACM International Conference on Multimedia (2013)
19. Mosleh, A., Bouguila, N., Hamza, A.B.: Image text detection using a bandlet-based edge detector and stroke width transform. In: Proceedings of the British Machine Vision Conference, pp. 63.1–63.12.BMVA Press (2012)