Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California

Doctorado en Ciencias en Ciencias de la Computación

Desarrollo de un sistema de detección y reconocimiento de texto multi-orientado en imágenes naturales bajo iluminación no uniforme

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de

Doctor en Ciencias

Presenta:

Julia Diaz Escobar

Ensenada, Baja California, México

2019

Tesis defendida por

Julia Diaz Escobar

y aprobada por el siguiente Comité

Dr. Vitaly Kober

Director de tesis

Dr. Hugo Homero Hidalgo Silva

Dr. Josué Álvarez Borrego

Dr. Félix Calderón Solorio

Dr. Jesús Favela Vara

Coordinador del Posgrado en Ciencias de la Computación

Dra. Rufina Hernández Martínez

Directora de Estudios de Posgrado

Resumen de la tesis que presenta Julia Diaz Escobar como requisito parcial para la obtención del grado de Doctor en Ciencias en Ciencias de la Computación.

Desarrollo de un sistema de detección y reconocimiento de texto multi-orientado en imágenes naturales bajo iluminación no uniforme

Resumen aprobado por:

Dr. Vitaly Kober
Director de tesis

La detección y reconocimiento de texto en imágenes naturales se refiere al proceso de localizar regiones de texto en una imagen capturada por un dispositivo móvil y convertir el texto en código único “entendible” para un ordenador o crear un archivo de texto editable. Los sistemas de detección y reconocimiento de texto contribuyen en múltiples aplicaciones relacionadas con clasificación de documentos, recuperación multimedia, interacción humana-computadora, navegación robótica y realidad aumentada. A diferencia de las imágenes digitalizadas por medio de un escáner, donde las condiciones de captura son controladas, las imágenes naturales suelen verse afectadas por aspectos ambientales, condiciones de captura y contenido del texto. Por esta razón, en este trabajo de tesis se proponen nuevos métodos de detección y reconocimiento de texto en imágenes naturales. En este trabajo se abordan los problemas de detección y reconocimiento de texto multi-orientado bajo condiciones de iluminación no uniforme, baja resolución y presencia de ruido aditivo. Se consideró el alfabeto latino sin acentos, así como los números del 0 al 9. Se consideró sólo texto escrito a máquina, con diferentes tipos de fuente y tamaños, así como fondos simples y complejos. El método propuesto de detección y segmentación de texto se basa en el modelo de energía local y el espacio-escala de la señal monogénica. Los resultados experimentales mostraron que el método propuesto es robusto a distorsiones geométricas, tipos de fuente, fondos complejos y cambios de iluminación. También, el método propuesto logró un alto desempeño en la tarea de segmentación de caracteres manteniendo un bajo número de componentes extraídos. El método superó los algoritmos del estado-del-arte en conjuntos de imágenes comunes en términos de segmentación de caracteres, localización de texto y número de regiones candidatas. Por otro lado, también se atacó el problema del reconocimiento de caracteres. Para este problema, se propuso un descriptor de características llamado LUIFT. Este descriptor y el modelo de bolsa de características fueron utilizados para la representación y clasificación de caracteres. El método sugerido mostró robustez a ligeras distorsiones geométricas, oclusión y degradaciones tales como iluminación no uniforme, ruido y baja resolución. Finalmente, se obtuvo un sistema completo de detección y reconocimiento de texto multi-orientado en imágenes naturales. El método propuesto fue evaluado utilizando conjuntos de imágenes comunes y comparado con técnicas del estado-del-arte.

Palabras clave: detección y segmentación de texto, iluminación no uniforme, señal monogénica, OCR.

Abstract of the thesis presented by Julia Diaz Escobar as a partial requirement to obtain the Doctor of Science degree in Computer Science.

Multi-oriented natural text detection and recognition system under non-uniform illumination.

Abstract approved by:

Dr. Vitaly Kober
Thesis Director

Text detection and recognition in natural images refer to the process of text regions localization in real images, captured by a mobile device (smartphone, digital camera, electronic tablet, etc.), in order to convert the detected text into Unicode characters "understandable" for computers and/or to create editable text files. Natural scene text detection and recognition systems, have gained much attention from the computer vision community due to their contribution to multiple applications, such as document classification, multimedia retrieval, language translator, human-computer interaction, robotic navigation, and augmented reality. Unlike the images obtained by scanning with controlled conditions, natural images may be affected by various aspects of the environment (non-uniform illumination, shadows, scene complexity), image acquisition problems (low resolution, blurring, perspective distortion, occlusion), and the text content (orientation, size, fontstyle, texture, color), owing to the nature of a capture device. In order to solve the before said problems, in this thesis robust natural text detection and recognition methods are proposed. In other words, the work addresses the problem of detection and recognition of multi-oriented text, under non-uniform illumination, low resolution and presence of additive noise. Typewritten text is considered with different types of fonts and sizes as well as with simple and complex backgrounds. A new method based on the local energy model and the scale-space monogenic signal for multi-oriented text detection and segmentation was proposed. Experimental results show that the suggested method is robust to geometric distortions, complex backgrounds, and illumination variations. The method outperforms the state-of-art algorithms on common datasets for character segmentation and text localization. Also, the character recognition problem was attacked. For this problem, a feature descriptor referred to as LUIFT was proposed. The LUIFT descriptor and the BOF approach were used to represent and classify characters. The suggested method shows a robust performance to slight geometric distortions, occlusion, and degradations such as non-uniform illumination, noise and low resolution. Finally, the end-to-end system was designed and implemented for detection and recognition of text in natural images. The performance of the system was evaluated using common datasets and compared to the state-of-the-art techniques.

Keywords: text detection and segmentation, non-uniform illumination, monogenic signal, OCR.

Dedicatoria

A Ernestina Escobar Daniel, mi madre.

Agradecimientos

A mi familia por todo su apoyo, en especial a mi hermana Eva y mis sobrinas Yahaira, Sharise y Korine, por todo su cariño y apoyo.

A mi asesor de tesis, el Dr. Vitaly Kober, por brindarme el apoyo y la confianza en estos últimos 6 años de formación académica.

A todos los miembros de mi comité de tesis, el Dr. Hugo Hidalgo Silva, el Dr. Josué Álvarez Borrego y el Dr. Félix Calderón Solorio, por sus críticas constructivas y consejos.

A todos mis compañeros y amigos en CICESE, por los cafecitos y las cervecitas.

A mis amigas Burny y Gaby, por su cariño, apoyo y porras.

Al Centro de Investigación Científica y de Educación Superior de Ensenada por permitirme realizar mis estudios de posgrado.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico para realizar mis estudios de doctorado. No. de becario: 275945.

Tabla de contenido

Página

Tabla de contenido (continuación)

Tabla de contenido (continuación)

Lista de figuras

Figura Página

Lista de figuras (continuación)

Lista de figuras (continuación)

Lista de figuras (continuación)

Figura Página

Lista de tablas

Tabla Página

Siglas

AdaBoost Adaptive Boosting. 74

BOF Bags of Features. 13, 92, 95

BRIEF Binary Robust Independent Elementary Features. 51

BRISK Binary Robust Scalable Keypoints. 51

CenSurE Center Surround Extremas. 51

CNN Convolutional Neural Network. 11

DaLI Deformation and Light Invariant. 51

DCT Discrete Cosine Transform. 8

DNN Deep Neural Network. 11

DOP Differences of Poisson. 46

ER Extremal Regions. 10, 11

FAST Features from Accelerated Segment Test. 50

FREAK Fast Retina Keypoint. 51

GB Geometric Blur. 12, 13

GT Ground Truth. 30

HOG Histogram of Oriented Gradients. 9–11, 13, 50, 57

HOPC Histogram of Phase Congruency. 57

LIFT Learned Invariant Feature Transform. 51

LIOP Local Intensity Order Pattern. 52

LOG Laplacian of Gaussian. 50, 51

LUIFT LUminance Invariant Feature Transform. 52, 92, 95, 108

MC-MR Multi-Channel Multi-Resolution. 11

MLBP Mean Local Binary Patterns. 10, 11

MR Maximum Response. 13

MSER Maximally Stable Extremal Regions. 9–11

NMS Non-Maximum Suppression. 56

OCR Optical Character Recognition. 6, 11, 12

PC Phase Congruency. 40

PCA Principal Components Analysis. 51

PD Patch Descriptor. 13

PDE Partial Differential Equations. 8

RF Random Forest. 9, 10

SC Shape Context. 12, 13

SFT Stroke Feature Transform. 10

SIFT Scale Invariant Feature Transform. 13, 50, 51

SURF Speed Up Robust Features. 50

SUSAN Smallest Univalue Segment Assimilating Nucleus. 50

SVM Support Vector Machine. 10, 93, 94, 96

SWT Stroke Width Transform. 9–11

TILDE Temporally Invariant Learned DEtector. 51

Variables

A_n Magnitud del n-ésimo componente de Fourier. 40

CC Componente conectado. 68, 69

CC_{SWV} Valor de ancho de trazo del componente. 69, 70

CC_{approx} Área aproximada del componente. 79

CC_{area} Número de pixeles del componente. 68, 70

$CC_{aspecto}$ Relación de aspecto del componente. 70

CC_{conext} Contorno externo del componente. 80

CC_{con} Contorno del componente. 69, 71

CC_{ejeMax} Eje máximo del componente. 69

CC_{ejeMin} Eje mínimo del componente. 69

CC_{envCon} Envolvente convexa del componente. 69

$CC_{minRect}$ Minimo rectángulo delimitador del componente. 69

CC_{numCon} Número de contornos del componente. 70, 71

CC_{pcr} Razón de congruencia de fase del componente. 78

CC_{pc} Congruencia de fase del componente. 69, 71, 77, 78

CC_{pf} Puntos finales del componente. 70, 71

CC_{rect} Rectángulo delimitador del componente. 69

CC_{skel} Esqueleto del componente. 69

CC_s Solidez del componente. 70

CC_{trazo} Ancho de trazo del componente. 70

Car Caracter. 81

D Rectángulo delimitador detectado. 33

D_{area} Diferencia de áreas entre dos componentes. 81

D_{gris} Diferencia de intensidades entre dos componentes. 81

D_h Razón de altura entre dos componentes. 82

$D_{minRect}$ Diferencia de áreas de los rectángulos delimitadores entre dos componentes. 81

D_{trazo} Diferencia de ancho de trazo entre dos componentes. 82

D_w Razón de anchura entre dos componentes. 82

$Dist_E$ Distancia de edición entre dos palabras. 101

$Dist_F$ Frecuencia de una palabra. 101

$Dist_H$ Distancia Hamming entre dos cadenas. 101

E Energía local en una dimensión. 40

F Señal en el dominio frecuencial. 44

F_M Señal monogénica en el dominio frecuencial. 44

F_{bp} Filtrado de la señal F por el filtro pasa-banda DOP. 46

GT Rectángulo delimitador verdadero. 33

I Imagen. 19, 21–23

I_s Imagen sintética. 19, 21–23

I_{area} Área de la imagen I. 70

L^2 Espacio de Hilbert. 40, 42

P Medida de precisión. 31

PC Congruencia de fase. 40, 55, 57

PC_{th} Umbral de la congruencia de fase. 77

R Medida de sensibilidad. 30

T Umbral estimado que controla la cantidad de ruido. 42, 56

VC Caracteres vecinos. 80, 81

$\mathbf{H}$ Homografía. 33

$\mathbf{M}_{\mathrm{H}}$ Matriz de Harris. 54

$\mathbf{M}_{\mathrm{c}}$ Matriz monogénica. 55

$\mathbf{M}_{\mathrm{m}}$ Matriz monogénica. 54

$\mathbf{M}_{\mathrm{pc}}$ Características con congruencia de fase. 55

$\mathbf{N}_{dwn}$ Máscara binaria descendente. xi, 67, 68

$\mathbf{N}_{up}$ Máscara binaria ascendente. xi, 67, 68

$\mathbf{R}$ Función de transferencia de la transformada de Riesz. 44

$\mathcal{C}$ Conjunto $\{0, 1, 2, .., 255\}^2$. 19

$\mathcal{D}$ Subconjunto de los números naturales. 19, 33, 34

$\mathcal{F}$ Transformada de Fourier. 43

err Error de traslape. 32

f Señal en el dominio espacial. 45

f_m Señal monogénica en el dominio espacial. 54

$minDist$ Mínima distancia ponderada entre dos componentes. 84

pt_i Punto en la imagen $pt_i = (x, y)$. 71

r Radio. 81

sim Similaridad entre dos componentes. 83

$\mathbf{T}$ Transformación afín. 20, 31

Capítulo 1. Introducción

Para el ser humano la detección y el reconocimiento de texto pueden llegar a ser tareas triviales. Todos los días realizamos actividades que involucran el reconocimiento de escenas con contenido textual: letreros de tiendas, señalamientos de tránsito, etiquetas de productos, anuncios de televisión, subtítulos de películas, noticias en periódicos o revistas, espectaculares, entre muchas otras. Estas escenas suelen contener texto con diferentes orientaciones, tipos de fuente, tamaños y colores, aun así, el ser humano es capaz de localizar y reconocer el texto en menos de un segundo, sin importar la presencia de otros objetos a su alrededor (personas, carros, arbustos, cercas, edificios, etc.), o que la iluminación no sea la adecuada (sombras, baja iluminación, iluminación no uniforme). Más aún, el ser humano es capaz de identificar zonas que contienen texto aunque no pueda entender su significado (texto en otro idioma).

Por otro lado, a diferencia de los seres humanos, los ordenadores no poseen dicha habilidad. Desde hace más de medio siglo, el hombre ha intentado “enseñar a leer” a las máquinas, pero aún no lo ha logrado del todo. Tal es el caso de Gustav Tauschek, quien en 1929 inventó la máquina mecánica de lectura, considerada como el primer dispositivo de reconocimiento óptico de caracteres (OCR, Optical Character Recognition), desarrollada mucho antes de que los ordenadores existieran tal y como los conocemos ahora. Desde entonces, múltiples esfuerzos se han realizado y, hoy en día, se puede decir que el reconocimiento de caracteres para documentos impresos (digitalizados por medio de un escáner) es un problema resuelto (Ye y Doermann, 2015). Sin embargo, debido al desarrollo tecnológico de los últimos años, el problema de reconocimiento de texto ha surgido nuevamente. De hecho, el problema se ha vuelto más complicado, ya que no sólo es necesario reconocer el texto, sino también localizar su posición dentro de la imagen (no necesariamente alineado horizontalmente).

Hoy en día, los sistemas de detección y reconocimiento de texto han atraído la atención de la comunidad de visión por computadora y análisis de documentos debido a su gran contribución en múltiples aplicaciones relacionadas con clasificación de documentos, recuperación multimedia, traducción de idiomas, interacción humana-computadora, navegación robótica y realidad aumentada, por mencionar algunas (Ye y Doermann, 2015; Zhu *et al.*, 2016b). Lamentablemente, a diferencia de las imágenes digitalizadas por medio de un escáner, donde las condiciones de captura suelen ser

controladas, las imágenes naturales (imágenes capturadas por un dispositivo móvil) no están sujetas a ningún tipo de restricción. Las escenas naturales de texto generalmente contienen diferentes tipos de fuente, símbolos, colores, escalas y orientaciones de caracteres, lo que hace que la detección de texto sea una tarea complicada. Además, las escenas naturales suelen capturarse en condiciones no controladas (cambios de iluminación, oclusión parcial, baja resolución, ruido del sensor, desenfoque, perspectiva), y pueden contener fondos complejos (personas, edificios, cercas, ladrillos, césped, árboles, autos) debido a la naturaleza del dispositivo (Ye y Doermann, 2015; Zhu *et al.*, 2016b; Kaur *et al.*, 2017).

En particular, para la detección y reconocimiento de texto se han desarrollado diversos trabajos que atacan algunos de estos desafíos, sin embargo, aún no existe un sistema robusto y eficiente que funcione sin ningún tipo de restricción. La mayoría de los métodos desarrollados hasta el momento suponen que el texto se encuentra alineado horizontalmente, que la iluminación de la escena es uniforme, o que el fondo de la imagen es simple. Desafortunadamente, no todas las escenas cumplen con estas suposiciones (Kaur *et al.*, 2017).

Es por ello que en esta tesis se proponen nuevos métodos robustos de detección y reconocimiento de texto en imágenes naturales. En este trabajo se aborda la problemática de detección y reconocimiento de texto multi-orientado, bajo iluminación no uniforme, baja resolución y presencia de ruido aditivo. Se considera texto escrito a máquina, con diferentes tipos de fuentes y tamaños. Se consideran fondos simples y complejos.

Los métodos propuestos en este trabajo de investigación tiene múltiples aplicaciones en diversas áreas de la ciencia, tales como: visión por computadora, navegación robótica, interacción humano computadora, sistemas de seguridad, recuperación de información, entretenimiento y sistemas industriales.

1.1. Antecedentes

El problema de detección y reconocimiento de texto en imágenes naturales está integrado por un conjunto de sub-tareas, las cuales a su vez, son problemas complejos y, por lo tanto, son atacados individualmente. Estas tareas son: (1) la localización

del texto dentro de la imagen y la segmentación de los caracteres del texto, (2) el reconocimiento de caracteres y (3) el reconocimiento de palabras (Ye y Doermann, 2015; Zhu *et al.*, 2016b; Kaur *et al.*, 2017). A continuación se describen cada una de ellas.

1.1.1. Detección y segmentación de texto

El objetivo de la detección de texto es obtener una estimación de la localización de las áreas de texto en la imagen. Esta estimación es comúnmente representada por rectángulos, mejor conocidos como rectángulos delimitadores, los cuales pueden encerrar palabras o lineas completas de texto (Figura 1(a)) .

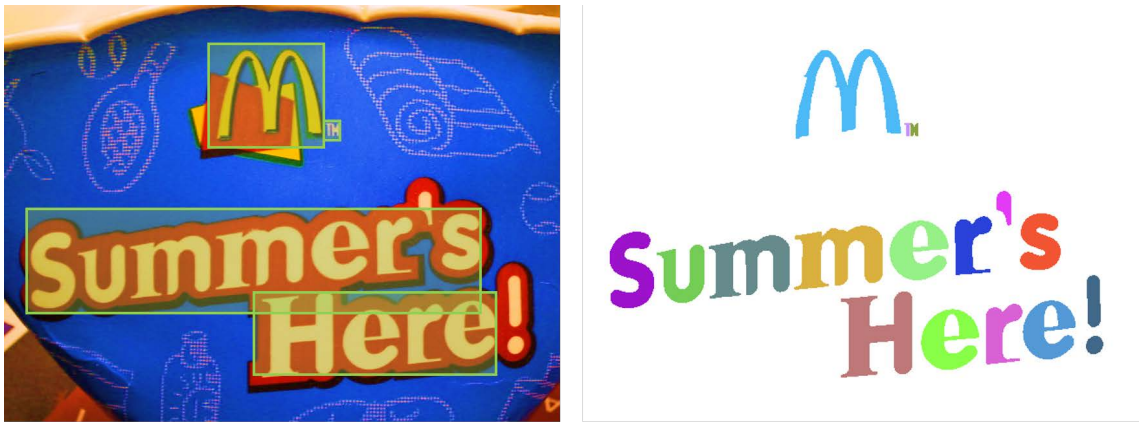


Figura 1. Ejemplo de las tareas de: a) localización y b) segmentación de texto en imagen natural.

Por otro lado, la tarea de segmentación de texto se refiere a la separación, a nivel de pixel, de los componentes del texto y el fondo de la imagen (Figura 1(b)).

En las últimas décadas, se han explorado diversas técnicas para resolver el problema de la detección y segmentación de texto. Estos métodos se pueden dividir en tres categorías: basados en ventanas deslizantes, basados en componentes conectados y métodos híbridos (Zhu *et al.*, 2016b). Los métodos basados en ventanas deslizantes, también llamados métodos basados en texturas, consideran una ventana deslizante en toda la imagen bajo diferentes escalas para identificar regiones de texto. Ecuaciones diferenciales parciales (PDE, Partial Differential Equations) (Zhao *et al.*, 2015), la transformada discreta de coseno (DCT, Discrete Cosine Transform) (Angadi y Koda-

bagi, 2009), filtros espaciales (Kim *et al.*, 2003), histogramas de gradiente orientado (HOG, Histogram of Oriented Gradients) (Pan *et al.*, 2008) y los coeficientes de ondícula (Saoi *et al.*, 2005), son comúnmente usados como propiedades texturales. Sin embargo, los métodos de ventanas deslizantes son sensibles a las variaciones de escala y rotación, además de ser costosos desde el punto de vista computacional ya que es necesario procesar múltiples ventanas de diferentes tamaños y rotaciones.

Los métodos basados en componentes conectados consideran propiedades como el color, el ancho de trazo, la relación de aspecto, el tamaño y otros, para distinguir entre regiones de texto y no-texto. Por lo general, los componentes conectados se obtienen mediante la agrupación de colores (Wu *et al.*, 2016; Tang *et al.*, 2015), binarización de imágenes (Liu y Sarkar, 2008; Karaoglu *et al.*, 2010), detección de bordes (Yu *et al.*, 2016), cálculo de la transformada de ancho de trazo (SWT, Stroke Width Transform) (Epshtein *et al.*, 2010) y la extracción de regiones extremas máximamente estables (MSER, Maximally Stable Extremal Regions) (Matas *et al.*, 2004; Matas y Zimmermann, 2005). Por último, los métodos híbridos combinan las técnicas de ventanas deslizantes y los métodos basados en componentes conectados (Yin *et al.*, 2014).

En los últimos años, las métodos MSER y SWT se han convertido en las técnicas más utilizadas para el proceso de detección de texto debido a su invariabilidad a las transformaciones de escala y rotación. El operador local SWT calcula el ancho de trazo del caracter para cada píxel del mapa de bordes, de tal forma que los trazos que tienen valores de anchura constantes pueden considerarse caracteres, y los componentes que tienen valores de anchura de trazo similares pueden agruparse en palabras. Dado que el SWT original es invariable a las variaciones de rotación y escala, se han desarrollado varios métodos basados en este enfoque. En los trabajos de Yao *et al.* (2012, 2014) se propone un método basado en SWT para la detección de texto multi-orientado. El detector de bordes Canny (Canny, 1986) se utiliza para calcular el mapa SWT a partir de la imagen. Los píxeles de la imagen se asocian considerando la relación SWT, y se agrupan en componentes conectados.Los componentes obtenidos se clasifican en elementos de texto y no-texto mediante un esquema de filtrado. Se considera un conjunto de reglas heurísticas y se aplica un clasificador del tipo bosque aleatorio (RF, Random Forest). Finalmente, los candidatos a caracter se agrupan en cadenas de texto satisfaciendo un cierto conjunto de reglas. Por otro lado, en el trabajo de Huang *et al.*

(2013) se propone una versión extendida del método SWT, llamada transformada de característica de trazo (SFT, Stroke Feature Transform). Además de las restricciones de ancho de trazo, la SFT considera la uniformidad de color y las relaciones locales de los píxeles de borde. También, se definen dos descriptores de covarianza de texto, los cuales se utilizan en conjunto con los descriptores HOG para el entrenamiento de clasificadores RF a nivel de componente y de línea de texto. En el trabajo de Huang *et al.* (2013) se propone el cálculo eficiente del valor del ancho de trazo. El valor del ancho de trazo obtenido se utiliza junto con información perceptual y un descriptor HOG de bordes para medir las propiedades de los caracteres bajo un marco bayesiano.

Por otra parte, el método MSER básicamente extrae regiones de la imagen que permanecen estables bajo un cierto número de umbrales, las cuales se consideran candidatos potenciales a caracteres. La técnica MSER fue introducida por primera vez por Matas y Zimmermann (2005) para la detección de caracteres, y recientemente ampliada para la detección y reconocimiento de texto (Neumann y Matas, 2016). En el trabajo de Yin *et al.* (2015) se propone un método de segmentación de texto basado en la técnica MSER. Las regiones se extraen utilizando el algoritmo MSER y se agrupan utilizando características como orientación y morfología. Posteriormente, los candidatos de texto se clasifican en componentes de texto y no-texto.

Recientemente, los candidatos a caracter no sólo se extraen de las regiones extremas máximamente estables, sino de todas las regiones extremas (ER, Extremal Regions) de la imagen. En el trabajo de Sung *et al.* (2015) se hace una división de sub-caminos del árbol de ER, creando múltiples sub-caminos de acuerdo con las similitudes de tamaño y posición o de regiones ER. Posteriormente, un clasificador AdaBoost (Schapire y Singer, 1999) es entrenado usando patrones binarios locales promedio (MLBP, Mean Local Binary Patterns) para la clasificación de texto y no-texto. Por último, algunas reglas heurísticas son utilizadas para el filtrado de caracteres mal clasificados. Por otro lado, en el trabajo de Saric (2017) también se utilizan regiones ER de baja variación y se clasifican utilizando características geométricas y una máquina de soporte vectorial (SVM, Support Vector Machine). Los caracteres obtenidos se agrupan en líneas de texto utilizando reglas heurísticas y se considera una etapa final de recuperación, si las regiones adyacentes satisfacen un conjunto de condiciones predefinidas. En (Zheng *et al.*, 2017) se propone un método similar basado en ER,

pero en lugar de características geométricas, se seleccionan las características HOG y MLBP para la clasificación y reconocimiento de caracteres. Posteriormente, los caracteres se agrupan en líneas de texto y se utiliza un modelo de (CNN, Convolutional Neural Network) para su verificación, eliminando los componentes que no son caracteres. En (Tian *et al.*, 2017) se propone una estrategia multi-canal y multi-resolución (MC-MR, Multi-Channel Multi-Resolution). Las regiones candidatas se extraen utilizando la técnica MSER bajo espacios de color RGB y YUV y bajo diferentes resoluciones. Posteriormente, los candidatos son filtrados y clasificados como componentes de texto y no-texto por un clasificador de CNN.

A pesar de que utilizar todas las regiones ER mejora el desempeño de los métodos descritos anteriormente, los métodos basados en ER necesitan procesar múltiples regiones repetidas para obtener una segmentación correcta de los caracteres, generando errores de clasificación y un alto costo computacional. Además, las técnicas basadas en SWT dependen de un detector de bordes preciso, lo que no es factible en muchos casos.

Recientemente, las técnicas basadas en redes neuronales profundas (DNN, Deep Neural Network) se han vuelto muy populares para el reconocimiento de patrones. En particular, para las tareas de detección y reconocimiento de texto se han propuesto diferentes enfoques y configuraciones de estas (Jaderberg *et al.*, 2014; Nguyen *et al.*, 2015; Tian *et al.*, 2016; Bušta *et al.*, 2017; He *et al.*, 2017). Sin embargo, las DNNs necesitan ser entrenadas utilizando miles, incluso millones de imágenes para lograr un buen desempeño y en muchos casos, se realiza un ajuste final con las imágenes de entrenamiento del conjunto de datos a ser evaluado (Bušta *et al.*, 2017; Liu *et al.*, 2018). Además, se ha demostrado que este tipo de enfoque puede ser fácilmente engañado modificando algunos valores de los píxeles de la imagen (Nguyen *et al.*, 2015).

1.1.2. Reconocimiento de caracteres

El reconocimiento óptico de caracteres (OCR, Optical Character Recognition) se refiere al proceso de reconocer de manera automática por medio de un ordenador caracteres y símbolos en imágenes digitales para posteriormente convertirlos en código único. La figura 2 muestra un ejemplo.

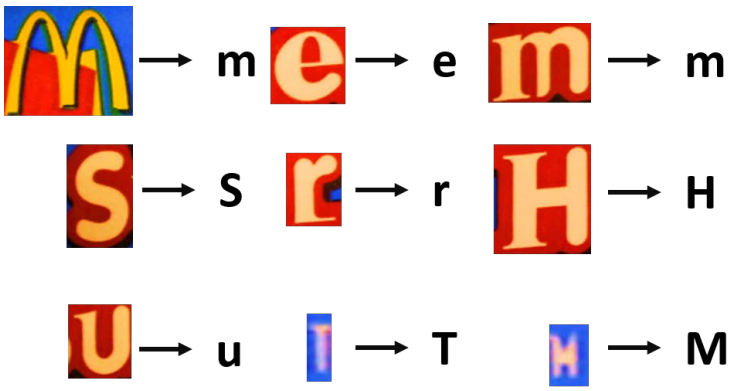


Figura 2. Ejemplo del reconocimiento de caracteres.

A pesar de que durante más de cuatro décadas el problema de reconocimiento de caracteres ha sido ampliamente estudiado y recientemente considerado como un problema resuelto (en imágenes de documentos impresos), en imágenes naturales sigue siendo un problema desafiante. Los caracteres son más difíciles de reconocer en imágenes naturales que en el análisis tradicional de documentos impresos, ya que las condiciones de captura no son controladas y los fondos de la escena se vuelven complejos. Por lo tanto, muchas de las técnicas conocidas de reconocimiento y clasificación de caracteres se han vuelto inadecuadas.

La correspondencia de plantillas fue uno de los primeros enfoques utilizados para el problema del reconocimiento de caracteres. La técnica consiste en hacer coincidir directamente la imagen de entrada con un conjunto de caracteres prototipo (Kumar *et al.*, 2006). Sin embargo, este enfoque es sensible a las deformaciones y variaciones de estilo (Diaz-Escobar *et al.*, 2015). Por otra parte, el enfoque por características también ha sido utilizado para la tarea de reconocimiento de caracteres. La distribución de puntos (momentos, cruces, distancias), transformaciones (Fourier, Haar, Hough) y análisis estructural (trazos, puntos finales, intersecciones, ángulos), son algunas características comunes utilizadas para la tarea de OCR (Trier *et al.*, 1996).

En los últimos años, una de las representaciones más populares para el reconocimiento de caracteres es el enfoque de contexto de forma (SC, Shape Context) (Belongie *et al.*, 2002). El descriptor SC es un histograma log-polar que considera la orientación y distancia de los puntos vecinos del contorno. Un método similar es el descriptor de desenfoque geométrico (GB, Geometric Blur) (Berg *et al.*, 2005), que a diferencia del descriptor SC, la región alrededor de cada punto de la característica se difumina

de acuerdo a la distancia con él, relajando la cuantificación. Además, también han sido utilizados el descriptor HOG, la transformada de característica invariante a escala (SIFT, Scale Invariant Feature Transform) y descriptores binarios (Yi *et al.*, 2013). De Campos *et al.* (2009) evaluó el rendimiento de seis tipos diferentes de descriptores locales: GB, SC, SIFT (Lowe, 1999), descriptor de vecindad (PD, Patch Descriptor) (Varma y Zisserman, 2003), descriptor de imagen giratoria (Lazebnik *et al.*, 2005) y respuesta máxima de filtros (MR, Maximum Response) (Varma y Zisserman, 2002), utilizando el enfoque de bolsa de características (BOF, Bags of Features). Por otro lado, en el trabajo de Bai *et al.* (2016) se propuso una representación multi-escala de los caracteres en imágenes naturales. Las características de trazo propuestas capturan las propiedades estructurales de los caracteres en diferentes granularidades. Además, estas características de trazo proporcionan una forma alternativa de identificar con precisión los caracteres individuales y componer un histograma para describirlos.

1.1.3. Reconocimiento de palabras

Recientemente, el reconocimiento de texto no se limita sólo al reconocimiento de caracteres individuales, sino que algunos métodos restringen aún más el problema cuando un léxico de palabras relativamente pequeño es dado con cada imagen y el objetivo es localizar sólo las palabras que se encuentran presentes en el léxico (Wang y Belongie, 2010; Jaderberg *et al.*, 2016). Sin embargo, cuando se aumenta este léxico, el desempeño disminuye considerablemente (Wang *et al.*, 2012).

Neumann y Matas (2016) proponen un descriptor de código de cadena y utilizar el algoritmo de k-vecinos más cercanos para su clasificación. En (Bissacco *et al.*, 2013) se propone un clasificador de red neuronal profunda entrenada utilizando descriptores HOG. Cada imagen de palabra es primero sobre-segmentada en fragmentos, cada segmento es clasificado por una red neuronal y finalmente es aplicada una búsqueda de haces para encontrar la secuencia óptima de caracteres.

Hasta ahora, la mayoría de los métodos propuestos relacionados con la detección de texto en escenas naturales se basan en los valores de intensidad de los píxeles. Como consecuencia, el rendimiento de los métodos se ve afectado por la presencia de iluminación no uniforme, bajo contraste, desenfoque o degradaciones provocadas por la presencia de ruido proveniente del sensor del sistema de captura. Además, la ma-

yoría de los métodos existentes en el estado del arte se restringen a texto horizontal y no consideran el reconocimiento de caracteres aislados.

1.2. Planteamiento del problema

El objetivo de la detección y reconocimiento de texto en imágenes naturales consiste en localizar zonas de texto dentro de una imagen, obtener su posición exacta y convertir dicho texto en una secuencia de caracteres en código único, "entendible" para los ordenadores (ver Figura 3), de tal forma que el texto pueda ser procesado posteriormente por el ordenador dependiendo de la aplicación a desarrollar.

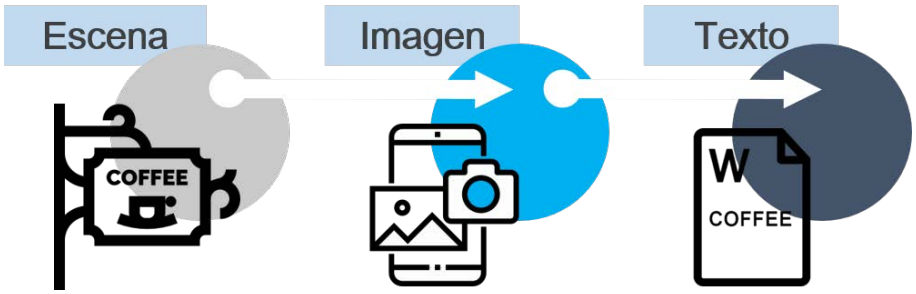


Figura 3. Detección y reconocimiento de texto en imágenes naturales.

Las imágenes digitales se pueden clasificar en tres tipos. Las imágenes de documentos impresos, las cuales son digitalizadas por medio de un escáner o sistemas especializados; imágenes naturales, las cuales son capturadas utilizando dispositivos móviles, como por ejemplo: cámaras digitales o vídeos, teléfonos móviles, tabletas electrónicas, cámaras webs, etc; y las imágenes de origen digital, las cuales fueron creadas digitalmente (ver Figura 4).

A diferencia del reconocimiento de imágenes de documentos impresos, donde los sistemas existentes en el estado del arte tienen una eficiencia del 99 %, la detección y el reconocimiento de texto en imágenes naturales sigue siendo un problema abierto para las comunidades de análisis de documentos y visión por computadora[1]. Esto debido a que las imágenes naturales suelen ser capturadas bajo condiciones no controladas (cambios de iluminación, oclusión parcial, baja resolución, ruido del sensor, borrosidad, perspectiva); y pueden contener fondos complejos (personas, edificios,

[1] http://rrc.cvc.uab.es/?com=introduction

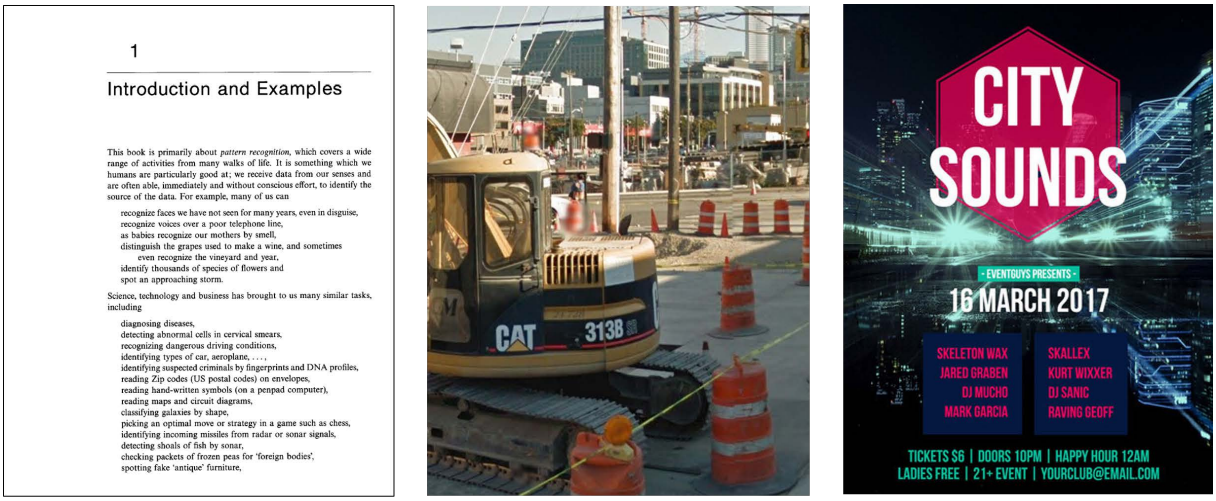


Figura 4. Ejemplo de imagen: (a) documento impreso, (b) natural y (c) origen digital.

cercas, ladrillos, césped, árboles, autos) (Zhang *et al.*, 2013; Ye y Doermann, 2015; Zhu *et al.*, 2016b).

En la literatura, los sistemas que realizan ambas tareas de detección y reconocimiento de texto en imágenes naturales se les conoce como sistemas de “principio-a-fin”. En este trabajo de investigación, el objetivo final es desarrollar un sistema de principio-a-fin.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un sistema de detección y reconocimiento de texto multi-orientado en imágenes naturales capturadas por dispositivos móviles en las cuales se presenten condiciones de iluminación no uniforme, baja resolución, con ligeras distorsiones geométricas y presencia de ruido, garantizando un grado de confiabilidad respecto a distintas métricas de desempeño.

1.3.2. Objetivos específicos

- Desarrollar una técnica de detección de texto multi-orientado en imágenes naturales bajo iluminación no uniforme, baja resolución y ruido.

- Desarrollar una técnica de selección y extracción de características para el reconocimiento de caracteres bajo iluminación no uniforme, baja resolución, ligeras distorsiones geométricas y ruido.
- Investigar y aplicar la técnicas para el reconocimiento de palabras haciendo uso de diccionarios.
- Evaluar el sistema propuesto y realizar un estudio comparativo con estado del arte en imágenes naturales.

1.4. Limitaciones y suposiciones

Como se mencionó anteriormente, la mayoría de los métodos desarrollados hasta el momento suponen que el texto se encuentra alineado horizontalmente, que la iluminación de la escena es uniforme, o que el fondo de la imagen es simple. Desafortunadamente, no todas las escenas cumplen con estas suposiciones.

En este trabajo se considera el alfabeto latino sin acentos, así como los números del 0 al 9. Se consideró sólo texto escrito a máquina, con diferentes tipos de fuente y tamaños, así como fondos simples y complejos. Se consideran degradaciones tales como iluminación no homogénea, sombras y ruido aditivo Gaussiano. Se consideran distorsiones geométricas tales como rotaciones en el plano dentro del rango de [-45,45] y cizallamiento en el rango de [-0.5,0.5].

1.5. Contribuciones y publicaciones

Esta tesis se basa en el resultado de las siguientes publicaciones obtenidas:

- **Character recognition in degraded document images using morphological and phase-only filtering**. Revista Communications Technology and Electronics (F.I. 0.48), 2015 Diaz-Escobar *et al.* (2015). Se propone un método para el reconocimiento de caracteres en imágenes de documentos degradados utilizando un banco de filtros morfológicos adaptativos y filtros de sólo fase.

- **Optical character recognition based on phase features**. Congreso ICCSAT, 2015 (Diaz-Escobar y Kober, 2015a). Se introduce una primera versión del reconomiento de caracteres utilizando información de fase local.

- **Optical character recognition of camera-captured images based on phase features**. Congreso SPIE, 2015 (Diaz-Escobar y Kober, 2015b). Se introduce una primera versión del descriptor propuesto basado en HOG para reconocimiento de caracteres.

- **Text detection in digital images captured with low resolution under nonuniform illumination conditions**. Congreso MCPR, 2016 (Diaz-Escobar y Kober, 2016b). Se explora y propone la detección de texto utilizando filtros SDF y la transformada sintética de acierto-fallo.

- **A robust HOG-based descriptor for pattern recognition**, Congreso SPIE, 2016 (Diaz-Escobar y Kober, 2016a). Se define el histogra de congruencia de fase orientado para reconocimiento de caracteres.

- **Text detection in natural scenes with phase congruency approach**. Congreso SPIE, 2017 (Diaz-Escobar y Kober, 2017). Se introduce una primera versión del detector de texto propuesto utilizando máscaras binarias basadas en fase local de la imagen.

- **Scene Text Segmentation Based on Local Image Phase Information and MSER Method**. Congreso MCPR, 2018 (Diaz-Escobar y Kober, 2018b). Se mejora el detector de texto propuesto utilizando regiones MSER basadas en fase local de la imagen.

- **Natural scene text detection and recognition with a three-stage local phase-based algorithm**. SPIE, 2018 (Diaz-Escobar y Kober, 2018a). Se introduce una primera versión del sistema propuesto de detección y reconocimiento de texto en imágenes naturales.

- **A new invariant to illumination feature Descriptor for pattern recognition**. Revista Communications Technology and Electronics, 2018, (Diaz-Escobar *et al.*, 2018b). Se introduce una primera versión del detector y descriptor LUIFT propuesto.

- **LUIFT: LUminance Invariant Feature Transform**. Revista Mathematical Problems in Engineering (F.I. 1.15), 2018 (Diaz-Escobar *et al.*, 2018a). Se extiende el artículo Diaz-Escobar *et al.* (2018b) y se realiza un mayor número de experimentos con diversas escenas reales.

- **Natural scene text detection and segmentation using phase-based regions and character retrieval**. Revista Neurocomputing (F.I. 3.2, Revisión), 2018 (Diaz-Escobar y Kober, 2018c). Se presenta y describe el detector de texto propuesto basado en la información de fase local de la imagen.

- **Algoritmos de binarizacion robusta de imágenes con iluminación no uniforme**. Revista Revista Iberoamericana de Automática e Informática industrial, 2017 (Molina *et al.*, 2017). Binarización de imágenes de texto con iluminación no uniforme (F.I. 0.49, colaboración).

1.6. Organización de la tesis

A continuación se describe la organización de esta tesis. En el Capítulo 2, Se presentan y describen las imágenes y métricas utilizadas para la evaluación del sistema propuesto y sus diferentes etapas. En el Capítulo 3, se presentan los fundamentos teóricos en los que se basa el presente trabajo de investigación. En el Capítulo 4, se presenta y describe el detector y descriptor de características propuesto LUIFT, así como algunos resultados experimentales. En el Capítulo 5, se presentan y describen los métodos desarrollados para la detección y segmentación de texto, así como algunos resultados experimentales. En el Capítulo 6, se presenta y describe el método de reconocimiento de caracteres propuesto, así como algunos resultados experimentales. En el Capítulo 7, se presenta el sistema de principio-a-fin propuesto, la evaluación experimental y la comparación del sistema propuesto con los sistemas existentes en el estado del arte. Finalmente, en el Capítulo 8, se presentan las conclusiones de este trabajo de investigación.

Capítulo 2. Protocolo de evaluación

En este Capítulo, se presentan los diferentes conjuntos de imágenes y métricas de desempeño utilizadas para evaluar el trabajo de tesis desarrollado. Primero, para analizar la tolerancia del método propuesto a escalamiento y rotación, así como a degradaciones de bajo contraste, alto brillo, sombras e iluminación no uniforme, se realizaron experimentos utilizando imágenes sintéticas. Estas imágenes fueron generadas sintéticamente utilizando transformaciones afines y modelos de degradación. Segundo, para comparar el desempeño del método propuesto y los métodos del estado del arte, se consideraron diferentes conjuntos de imágenes comunes en la comunidad de análisis de documentos, los cuales se presentan y se describen brevemente. Finalmente, se presentan y describen las métricas de desempeño que fueron utilizadas a lo largo de este trabajo de investigación para la evaluación y comparación del método propuesto y los existentes en el estado del arte.

2.1. Imágenes sintéticas

Sea $I : \mathcal{D} \subset \mathbb{N}^2 \rightarrow \mathcal{C}$, con $\mathcal{C}$ el conjunto $\{0, \ldots, 255\}^2$ (imagen en escala de grises). Las imágenes sintéticas I_s, están constituidas por una imagen digital en escala de grises I, modificada sintéticamente por una transformación afín o un modelo de degradación. A continuación se describen las trasformaciones y modelos utilizados.

2.1.1. Transformaciones afines

En términos de procesamiento digital de imágenes, una transformación geométrica consiste en dos operaciones básicas: (1) una transformación espacial de las coordenadas y (2) una interpolación de los valores de intensidad asignados a los píxeles transformados espacialmente (González y Woods, 2006).

Una de las transformaciones de coordenadas espaciales más utilizadas es la transformación, puede ser expresada como:

$$(x'_0, y'_0) = \mathbf{T}\{(x_0, y_0)\}, \tag{1}$$

con (x_0, y_0) las coordenadas del pixel en la imagen original, (x'_0, y'_0) las coordenadas del pixel en la imagen transformada y afín **T** representa a tranformación geométrica.

La transformaciones afines tienen la forma general:

$$[x'_0, y'_0, 1]^t = \begin{bmatrix} s_x \cdot cos(\theta) & c_x \cdot sin(\theta) & t_x \\ -c_y \cdot sin(\theta) & s_y \cdot cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} [x_0, y_0, 1]^t, \quad (2)$$

con t la transpuesta del vector. Esta transformación puede escalar (s_x, s_y), rotar (θ), trasladar (t_x, t_y) o cizallar (c_x, c_y) un conjunto de puntos, dependiendo del valor elegido para los elementos de la matriz **T**. La Figura 5 ilustra las distintas transformaciones.

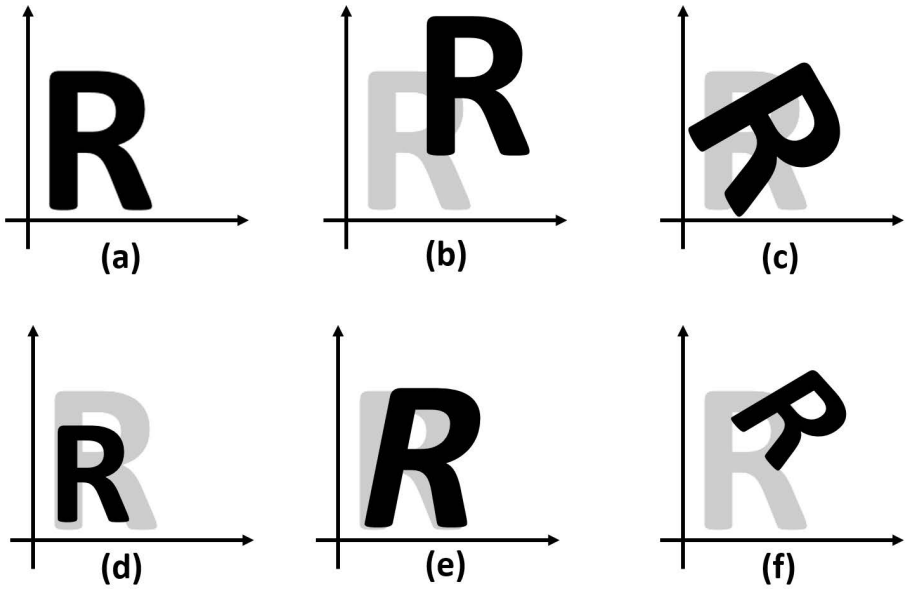


Figura 5. Transformaciones afines: (a) imagen original, (b) traslación, (c) rotación, (d) escalamiento, (e) cizallamiento, (f) combinación de todas las transformaciones.

2.1.2. Ruido

Se le llama ruido a cualquier información indeseable que contamina una imagen, principalmente esta información es obtenida en el proceso digital de adquisición de las imágenes. El ruido Gaussiano, también llamado ruido blanco aditivo, comúnmente presente en imágenes naturales (ruido electrónico en un sistema de captura de imagen). La función de densidad de probabilidad Gaussiana de una variable aleatoria se

define como:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right], \quad (3)$$

donde z representa el nivel de gris, μ es la media y σ es la desviación estándar de los valores de z (González y Woods, 2006).

Una imagen I contaminada por el ruido η se modela como:

$$I_s(x, y) = I(x, y) + \eta(x, y), \quad (4)$$

con I_s la imagen sintética resultante. La Figura 6(a) muestra un ejemplo de imágenes sintéticas degradadas por ruido Gaussiano.

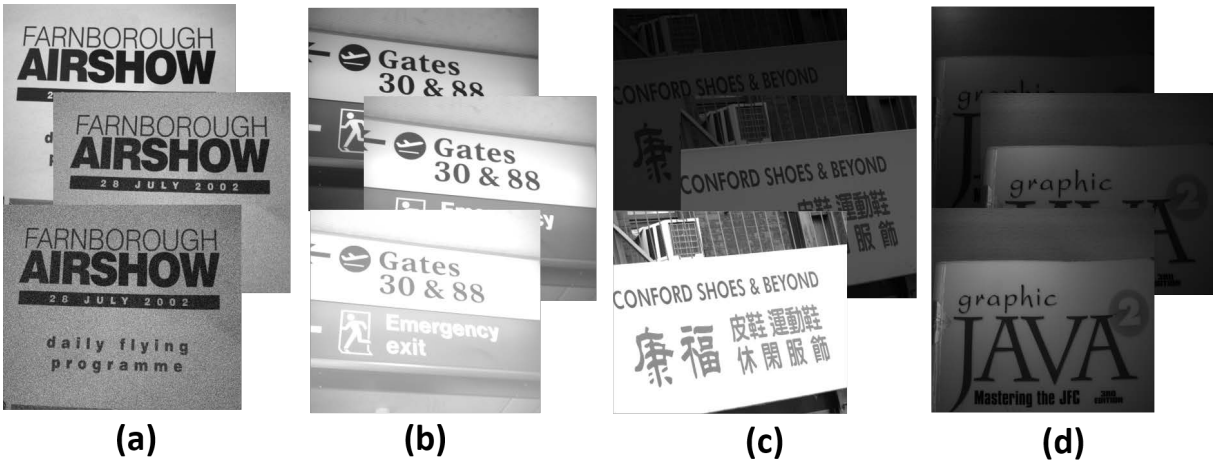


Figura 6. Ejemplo de degradaciones sintéticas: (a) ruido Gaussiano, (b) brillo, (c) contraste y (d) iluminación no uniforme.

2.1.3. Brillo y contraste

El contraste se puede definir como la diferencia de intensidad entre un pixel y sus pixeles vecinos. El bajo contraste puede ser causado por una baja iluminación, el tamaño del sensor del dispositivo de captura, la exposición (cantidad de luz recibida en un tiempo determinado) o la apertura de la lente durante la adquisición de la imagen.

El brillo (b) y el contraste (c) de la imagen I es modificado utilizando la siguiente función (González y Woods, 2006):

$$I_s(x, y) = c \cdot I(x, y) + b, \quad (5)$$

con I_s la imagen sintética resultante. Las Figuras 6(b) y (c) muestran un ejemplo de imágenes sintéticas degradadas por brillo y contraste, respectivamente.

2.1.4. Modelo de iluminación Lambertiano

La iluminación de una escena real depende de la fuente de iluminación utilizada y la forma de la superficie, un modelo de iluminación describe la relación entre estas dos. En general se pueden describir tres modelos (Horn, 1990):

- Especular: en este modelo el ángulo de incidencia es el mismo que el ángulo reflejado.
- Lambertiano: en este modelo se supone que la superficie refleja la luz en todas direcciones.
- Mixto: este modelo es una combinación de los modelos anteriores.

El modelo que se utilizó en este trabajo fue el modelo Lambertiano, el cual se describe a continuación (Diaz-Ramirez y Kober, 2009).

Considérese la escena que se muestra en la Figura 7 donde la superficie es iluminada por una fuente de luz puntual con los siguientes parámetros $I = [\rho, \tau, \alpha]$, donde ρ representa la distancia entre un punto en la superficie y la fuente de luz y τ, α representan los ángulos de inclinación entre la normal de la superficie y el punto de observación. Sea θ el ángulo de incidencia de la luz, es decir el ángulo entre el vector de la superficie normal N y el vector de la dirección de la luz, $\mathcal{I}$, la luz reflejada por una superficie Lambertiana esta dada por $R_L = cos(\theta)$. De acuerdo con la Figura 7, la luz reflejada por la superficie para una posición conocida de la fuente de iluminación y suponiendo que el punto de observación se encuentra sobre el eje z (Diaz-Ramirez y Kober, 2009):

$$d(x,y) = cos\left\{\frac{\pi}{2} - arctan\left[\frac{\rho}{cos(\tau)}\left[(\rho tan(\tau)cos(\alpha) - x)^2 + (\rho tan(\tau)sin(\alpha) - y)^2\right]^{-1/2}\right]\right\}. \tag{6}$$

Note que la función $d(x, y)$ en la Ecuación 6 es una función multiplicativa que depende de los parámetros ρ, τ y α, de tal forma que una imagen I degradada por iluminación

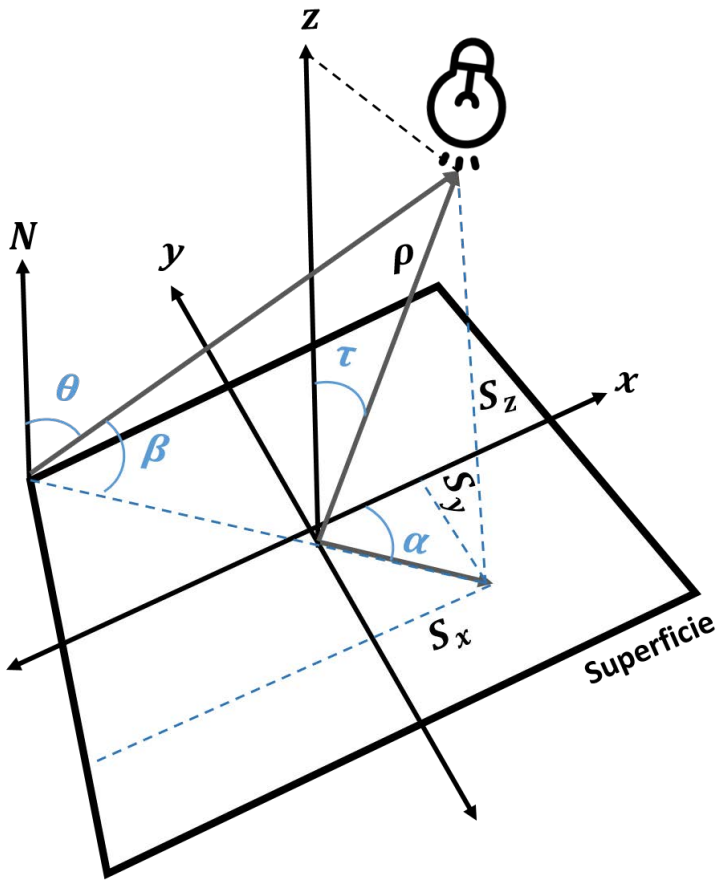


Figura 7. Modelo de iluminación Lambertiano.

no uniforme se modela como:

$$I_s(x, y) = d(x, y) \cdot I(x, y), \tag{7}$$

con I_s la imagen sintética resultante. La Figura 6(d) muestra un ejemplo de imágenes sintéticas degradadas por iluminación no uniforme.

2.1.5. Plantillas sintéticas

Para evaluar el desempeño del método propuesto bajo degradaciones de sombras e iluminación no uniforme, se crearon 25 plantillas sintéticas (ver Figura 8), las cuales simulan sombras e iluminación no uniforme. Las plantillas fueron generadas utilizando un fondo blanco, una cámara digital de 8MP y una lámpara. Cada una de las plantillas es normalizada (minMax, rango [0,1]). Las imágenes sintéticas, se obtienen de multiplicar la imagen original I con cada unas de las plantillas sintéticas, como muestra la

Figura 9.

Figura 8. Plantillas sintéticas.

Figura 9. Ejemplo de iluminación no uniforme generada utilizando plantillas sintéticas.

2.2. Conjuntos de imágenes del estado del arte

Los conjuntos de imágenes que se describen a continuación fueron los utilizados en este trabajo de investigación para evaluar distintas etapas del sistema propuesto. Los conjuntos de imágenes PHOS[1] y OFFICE[2] fueron utilizados para evaluar el desempeño del descriptor propuesto bajo condiciones de ruido, iluminación no uniforme, sombras y ligeras distorsiones geométricas. El conjunto de imágenes Chars74K[3] fue utilizado para el entrenamiento y evaluación del clasificador utilizado para el reconocimiento de

[1] http://www.computervisiononline.com/dataset/1105138614
[2] http://www.zhwang.me/datasets.html
[3] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

caracteres, mientras que los conjuntos ICDAR2013[4], MSRA-TD500[5] y USTB-SV1K_V1[6] fueron utilizados para evaluar el desempeño de los métodos de segmentación y clasificación de texto. Además, el conjunto ICDAR2013 fue utilizado también para el entrenamiento de los clasificadores en la etapa de detección de texto. Finalmente los conjuntos OSTD y ICDAR2013 fueron utilizados para evaluar el sistema final completo.

2.2.1. PHOS

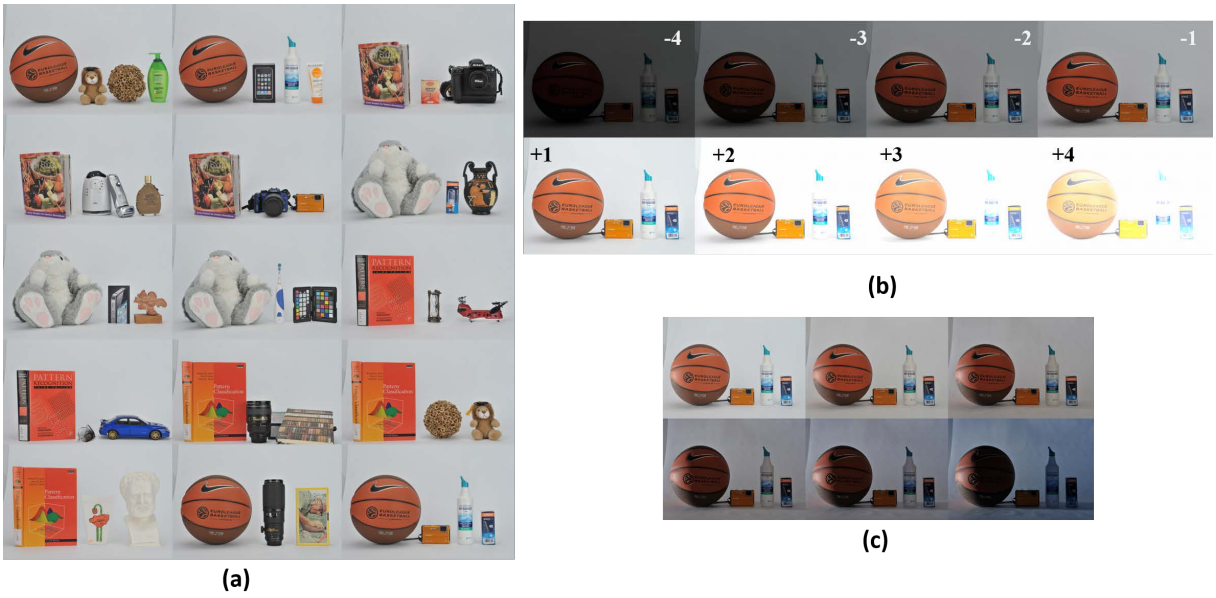


Figura 10. Ejemplo conjunto de imágenes PHOS: (a) escenas, (b) variación de exposición y (c) variación de iluminación.

El conjunto de imágenes PHOS (Vonikakis *et al.*, 2013) contiene 15 escenas diferentes (ver Figura 10(a)) capturadas bajo distintas condiciones de iluminación. Cada una de las escenas contiene 15 imágenes diferentes: 9 imágenes capturadas bajo diferentes grados de iluminación uniforme, variando la exposición de la cámara entre -4 y +4 de la imagen original expuesta correctamente (ver Figura 10(b)); y 6 imágenes bajo diferentes grados de iluminación no uniforme,obtenida añadiendo una fuente de luz dirigida a las luces uniformes ubicadas alrededor de los objetos (ver Figura 10(c)).

[4]http://rrc.cvc.uab.es/
[5]http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA-TD500)
[6]http://prir.ustb.edu.cn/TexStar/MOMV-text-detection/

2.2.2. OFFICE

El conjunto de imágenes OFFICE (Wang *et al.*, 2011b) contiene dos diferentes escenas llamadas “corredor” y “escritorio”. Cada una de las escenas contiene un conjunto de 5 imágenes con variaciones de iluminación. La Figura 11 muestra las imágenes del conjunto.

Figura 11. Conjunto de imágenes OFFICE: Escenas corredor y escritorio.

2.2.3. OSTD

Figura 12. Conjunto de imágenes OSTD.

El conjunto de imagenes OSTD (Yi y Tian, 2011) contiene 89 imágenes con texto multi-orientado. Estas imágenes fueron capturadas por los autores o seleccionadas de internet. Las imágenes contienen texto con diferentes tipos de fuente, tamaños,

colores y fondos complicados. La resolución de la mayoría de las imágenes oscila entre 600 × 450 y 1280 × 960. La Figura 12 muestra algunas escenas del conjunto.

2.2.4. ICDAR2013

El conjunto de escenas ICDAR2013 (Karatzas *et al.*, 2013) consta de 462 escenas complejas divididas en imágenes de entrenamiento (299) y de prueba (233). Cada imagen contiene diferentes fondos complejos, tipos de fuente, tamaños, desenfoque, iluminación, contraste, etc. El tamaño de las imágenes varía de 480 × 640 a 3888 × 2592. La Figura 13 muestra algunas escenas del conjunto.

Figura 13. Conjunto de imágenes ICDAR2013.

2.2.5. CHARS74K

El conjunto de imágenes Chars74K (De Campos *et al.*, 2009) está formado por 62 clases de caracteres (números y letras en mayúsculas y minúsculas). El conjunto de imágenes contiene fuentes sintéticas (Fnt) e imágenes reales (Img) tomadas de escenas naturales.

El subconjunto Img contiene 12,503 imágenes de caracteres RGB, recortados de imágenes naturales de señales, productos, letreros, etc. La Figura 14(a) muestra algunos ejemplos de los caracteres. El tamaño de cada imagen varía de 5×9 a 160×161.

El subconjunto Fnt está compuesto por 254 fuentes diferentes en 4 estilos (normal, negrita, cursiva y negrita + itálica) generando un total de 62,992 caracteres en escala de grises. Todas las imágenes sintéticas tienen un tamaño de 121×121. La Figura 14(b) muestra algunas escenas del conjunto.

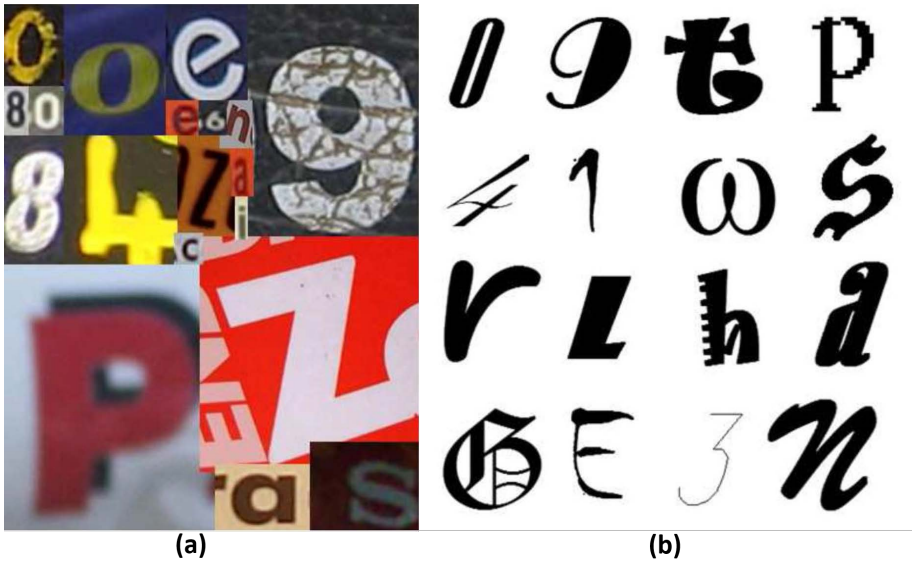


Figura 14. Conjunto de imágenes chars74k: (a) conjunto Img, (b) conjunto Fnt.

2.2.6. USTB-SV1K_V1

El conjunto de imágenes USTB-SV1K_V1 (Yin *et al.*, 2015) consta de 1000 imágenes de vistas de la calle de Google divididas en imágenes de entrenamiento (500) y de prueba (500). Las imágenes contienen texto multi-orientado y distorsión de perspectiva. El tamaño de las imágenes es de 512×512. La Figura 15 muestra algunas escenas

del conjunto.

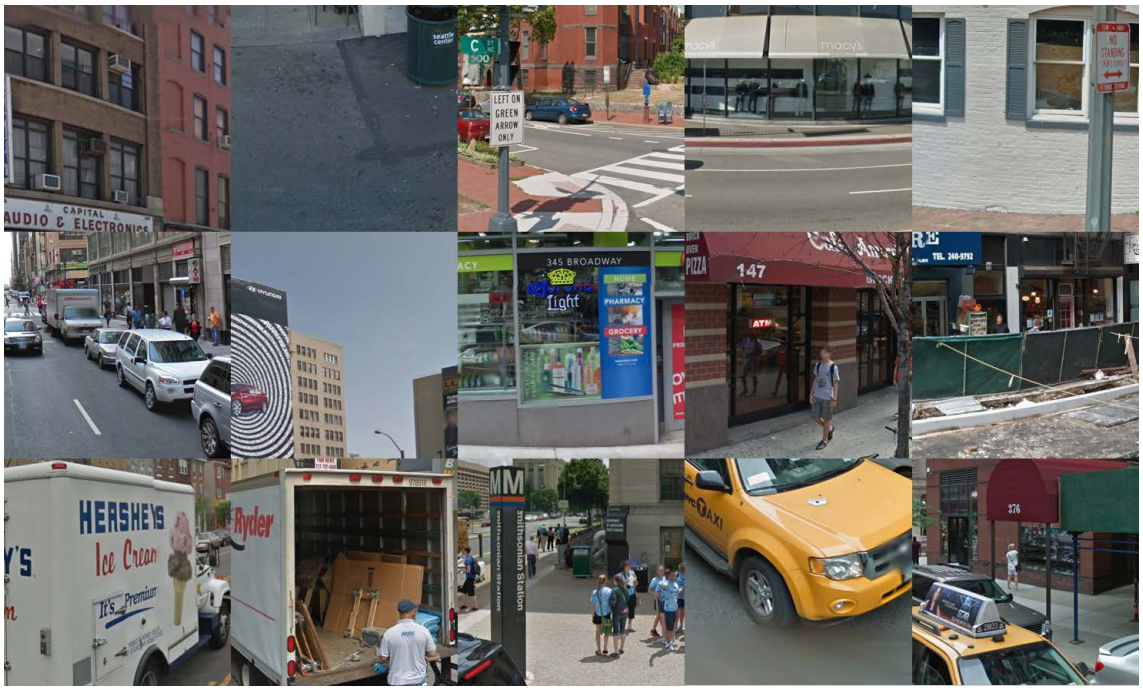


Figura 15. Conjunto de imágenes USTB-SV1K_V1.

2.2.7. MSRA-TD500

El conjunto de imágenes MSRA-TD500 (Yao *et al.*, 2012) contiene 500 imágenes naturales divididas en 300 imágenes de entrenamiento y 200 imágenes de prueba. El conjunto de imágenes esta conformado por escenas de interiores (oficina y centro comercial) y exteriores (calle) utilizando una cámara de paquete. Las imágenes de interiores son principalmente letreros de tiendas, placas de puertas y señalamiento de precaución, mientras que las imágenes de exteriores son en su mayoría son letreros de guía y publicitarios con fondos complejos. Las imágenes del conjunto contienen texto con diferentes fuentes, tamaños, colores, orientaciones e idiomas (chino, inglés o mezcla de ambos). Los fondos pueden contener vegetación (árboles y hierbas) y patrones repetidos (ventanas y ladrillos). Finalmente, la resolución de las imágenes varía de 1296×864 a 1920×1280. La Figura 16 muestra algunas escenas del conjunto.

Figura 16. Conjunto de imágenes MSRA-TD500.

2.3. Métricas de desempeño

Para evaluar el desempeño del sistema propuesto, distintas métricas fueron utilizadas en cada etapa del sistema. A continuación se describen brevemente cada una de ellas.

2.3.1. Matriz de confusión

Para evaluar la predicción del clasificador, se comparan los resultados del predictor con las observaciones reales (GT, Ground Truth), tal como lo muestra la Tabla 1 (Han *et al.*, 2011).

Tabla 1. Matriz de Confusión

		OBSERVACIÓN REAL	
PREDICCIÓN		**Clase A**	**No Clase A**
	Clase A	Verdadero Positivo (VP)	Falso Positivo (FP)
	No Clase A	Falso Negativo (FN)	Verdadero Negativo (VN)

2.3.2. Sensibilidad y Precisión

Estas métricas son muy empleadas en el ámbito de búsqueda y recuperación de información, así como en el reconocimiento de patrones. La medida de sensibilidad (R)

indica la fracción de los elementos relevantes que fueron reconocidos, mientras que la precisión (P) indica la fracción de los elementos reconocidos que fueron relevantes. La sensibilidad y precisión se definen formalmente de la siguiente manera (Han *et al.*, 2011):

$$R = \frac{VP}{VP + FP}, \tag{8}$$

$$P = \frac{VP}{VP + FN}. \tag{9}$$

Finalmente, la medida F (F) representa un promedio ponderado de las métricas de sensibilidad y precisión. La medida F se definen de la siguiente manera (Han *et al.*, 2011):

$$F = 2 \cdot \frac{P \cdot R}{P + R}. \tag{10}$$

Estas tres métricas son utilizadas para la evaluación del desempeño del método desarrollado en este trabajo. Sin embargo, dependiendo del tipo de tarea a evaluar, las métricas descritas anteriormente son definidas adecuándolas al problema en particular.

Para evaluar el desempeño de los detectores y descriptores, comúnmente son empleadas las métricas de medida de repetibilidad, medida de correspondencia y error de traslape. A continuación se describen brevemente cada una de ellas.

2.3.3. Repetibilidad y sensibilidad

Sea $P = \{p_i |\ i = 1, 2, \cdots, N,\ N \in \mathbb{N}\}$ el conjunto de características detectadas en la imagen original $I(x, y)$ con $x, y \in \mathcal{D}$, $Q = \{q_j |\ j = 1, 2, \cdots, M,\ M \in \mathbb{N}\}$ el conjunto de características detectadas en la imagen de prueba $I'(x, y)$ y $\mathbf{T}$ una trasnformación geométrica tal que $\mathbf{T}\{p_i\} = q_j$. Se considera una coincidencia si $\|\mathbf{T}\{p_i\} - q_j\| \leq \varepsilon$, donde $\|\cdot\|$ denota la distancia Euclidiana, y $\varepsilon = 2$ pixeles (Carneiro y Jepson, 2002). El desempeño del detector de características se evalúa utilizando la medida de repetibilidad (Mikolajczyk y Schmid, 2004) definida como la relación entre el número de coincidencias punto a punto y el mínimo número de puntos detectados en ambas imágenes. Un valor de repetibilidad cercano a uno nos indica que las características detectadas por el método evaluado se mantienen estables (en número y posición) bajo diferentes imá-

genes de prueba, en caso contrario, el detector falla en la detección de características. Finalmente, la sensibilidad se define como la relación entre el número de coincidencias punto a punto y el número de puntos detectados en la imagen original.

2.3.4. Error de traslape y correspondencia

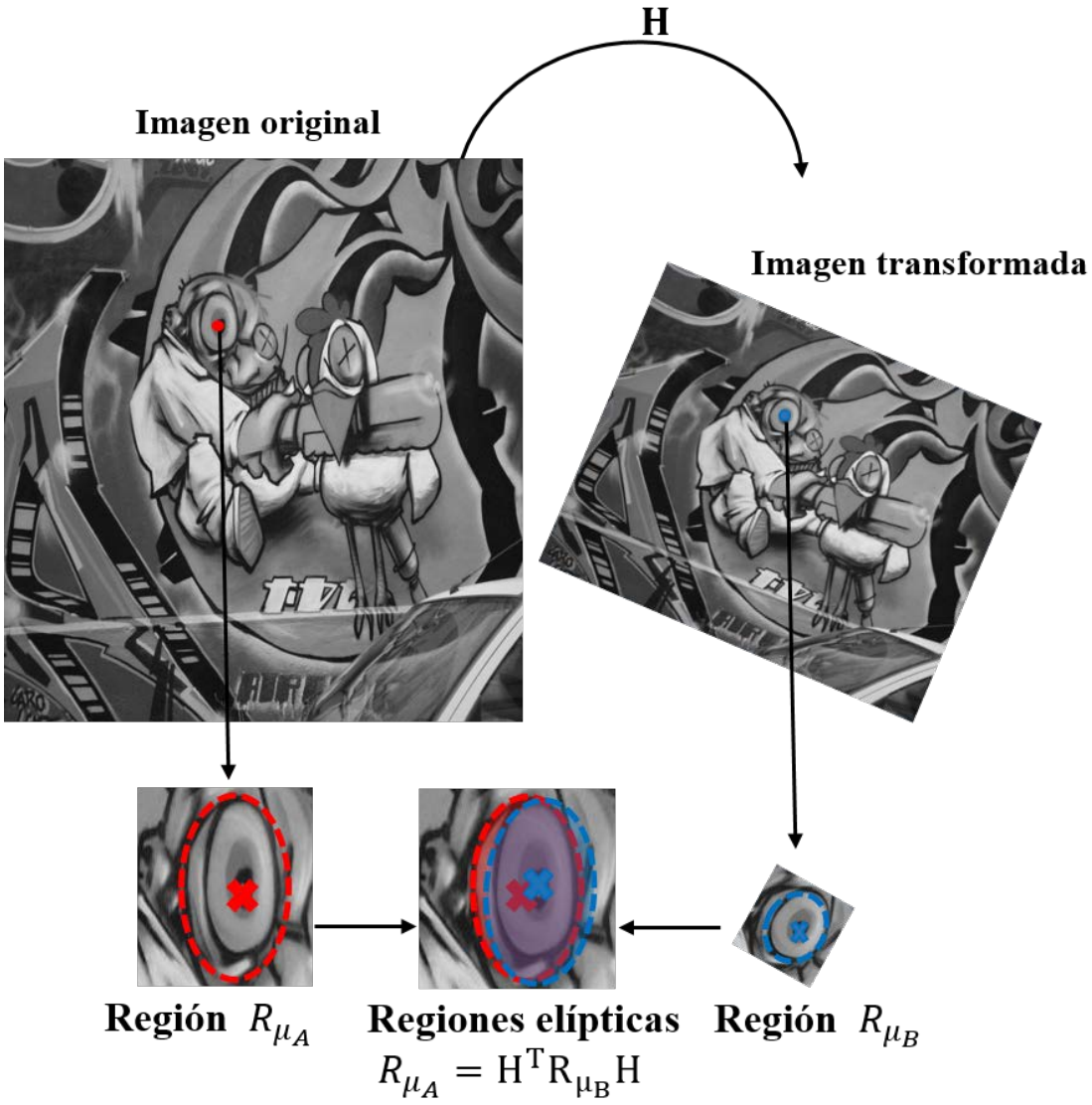


Figura 17. Error de traslape: intersección de regiones con correspondencia correcta.

La medida de correspondencia se determina con el error de traslape (*err*) (Mikolajczyk y Schmid, 2004). Básicamente, el error de traslape (también conocido como error de superficie) nos indica qué tan bien se intersectan dos regiones de caracte-

rísticas detectadas (ver Figura 17). El error de traslape se define como el cociente de la intersección de las regiones $R_{\mu_A} \cap R_{(H^T \mu_B H)}$ y su unión $R_{\mu_A} \cup R_{(H^T \mu_B H)}$ de la siguiente manera:

$$\varepsilon = 1 - \frac{R_{\mu_A} \cap R_{(H^T \mu_B H)}}{R_{\mu_A} \cup R_{(H^T \mu_B H)}}, \tag{11}$$

donde R_{μ_A} and R_{μ_B} son las regiones elípticas definidas por la matriz de segundo momento que satisfacen $x^T \mu x = 1$, y $\mathbf{H}$ es la homografía que relaciona a las dos imágenes. El error de traslape utilizado es $\varepsilon < 0.5$ (Mikolajczyk y Schmid, 2004).

Finalmente, la medida de correspondencia se calcula como:

$$correspondencia = \frac{correspondencias\ correctas}{correspondencias\ totales}. \tag{12}$$

2.3.5. Medida de sensibilidad-similitud

Para la evaluación de la generación de regiones candidatas a caracter, se utiliza la medida de sensibilidad-similitud. La medida de sensibilidad-similitud se define como la razón entre el total de las regiones candidatas correctamente detectadas y las regiones de caracteres verdaderas. Se considera que una región candidata es correctamente detectada si el valor de similitud es mayor del 50 %. El valor de similitud se define como (Sung *et al.*, 2015):

$$similitud(D, GT) = \frac{area(D) \cap area(GT)}{area(D) \cup area(GT)}, \tag{13}$$

donde D y GT representan el rectángulo delimitador de la región detectada y la región verdadera, respectivamente. Un rectángulo delimitador se refiere al mínimo rectángulo que encierra una palabra (ver Figura 18).

2.3.6. Métricas de segmentación de caracteres

Para la evaluación de la segmentación, se utilizan las medidas basadas en píxeles y estructurs. Sea $SG(x, y)$ con $x, y \in \mathcal{D}$, la segmentación verdadera de la imagen definida

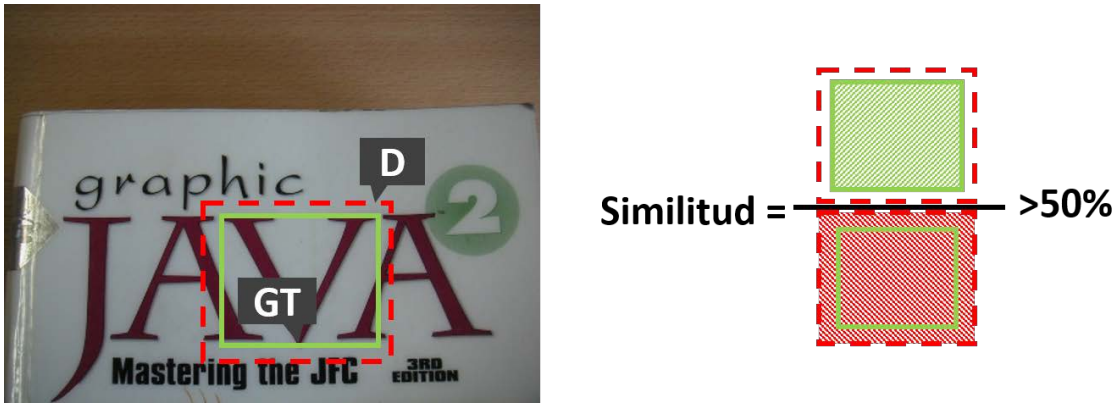


Figura 18. Medida de similitud. GT: rectángulo delimitador de la región del caracter verdadero (línea continua verde), D: rectángulo delimitador de la región detectada (línea punteada roja).

como:

$$SG(x,y) = \begin{cases} 0, & fondo, \\ 1, & texto. \end{cases} \tag{14}$$

y $B(x,y)$ $x,y \in \mathcal{D}$, la segmentación obtenida por el detector. La medida a nivel-pixel (Ntirogiannis *et al.*, 2008) define las medidas de sensibilidad y precisión de la siguiente manera:

$$R = \frac{\sum_{i=1,j=1}^{i=I_x,j=I_y} SG(i,j) \cdot B(i,j)}{\sum_{i=1,j=1}^{i=I_x,j=I_y} SG(i,j)}, \tag{15}$$

$$P = \frac{\sum_{i=1,j=1}^{i=I_x,j=I_y} EG(i,j) \cdot B(i,j)}{\sum_{i=1,j=1}^{i=I_x,j=I_y} B(i,j)}. \tag{16}$$

Por otro lado, a diferencia de la medida a nivel-pixel, la medida a nivel-estructura no sólo considera la precisión a nivel de los píxeles, sino también las propiedades morfológicas de los caracteres. En el trabajo de Clavelli *et al.* (2010) se introdujeron los criterios de cobertura mínima y máxima, los cuales miden el grado de superposición entre el área del componente verdadero (estructura) y el componente segmentado obtenido. El criterio de cobertura mínima se cumple si el umbral predefinido $T_{min} = 90\,\%$ de los píxeles del esqueleto del componente verdadero es cubierto por el componente segmentado. Del mismo modo, para el criterio máximo, la distancia de los pixeles del componente detectado al borde del componente verdadero no debe exceder un umbral máximo de $T_{max} = min(5, 0.5 \cdot SW)$, donde SW es el valor máximo de ancho de trazo del componente verdadero. La Figura 19(a) muestra un ejemplo donde se

cumplen los dos criterios de cobertura máxima y mínima. Como se puede observar, el esqueleto (negro) del componente verdadero (línea continua verde) es cubierto perfectamente por el componente segmentado (línea punteada roja), además, el ancho de trazo (sw) del componente segmentado cubre perfectamente al componente verdadero. Por lo tanto, se le considera al componente segmentado como correctamente detectado. En la Figura 19(b) el componente detectado no cubre por completo al esqueleto del componente verdadero. Por lo tanto, no cumple el criterio de cobertura mínima. Finalmente, en la Figura 19(c) el componente segmentado cubre perfectamente el esqueleto del componente verdadero pero su ancho de trazo no cubre el ancho de trazo del componente verdadero. Por lo tanto, no se cumple el criterio de cobertura máxima.

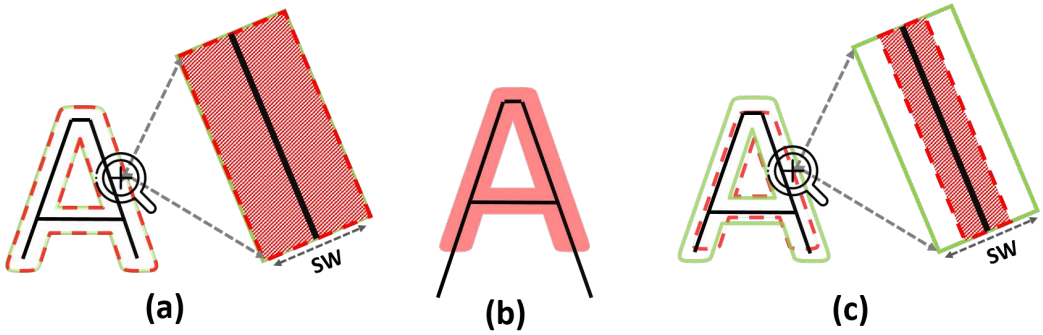


Figura 19. Ejemplos de medida nivel-estructura: componente verdadero (línea continua verde), esqueleto del componente verdadero (negro) y componente segmentado (línea punteada roja). (a) Se cumplen los criterios de cobertura máxima y mínima; (b) no se cumple el criterio de cobertura mínima; (c) no se cumple el criterio de cobertura máxima.

Suponiendo que N_{GT} es el número de elementos estructurales verdaderos, N_{RET} es el número de elementos estructurales detectados por el método de segmentación de texto y N_C es el número de elementos estructurales detectados correctamente, se definen las métricas de sensibilidad y precisión para el método de segmentación de texto, de la siguiente manera:

$$P = \frac{N_C}{N_{RET}} \quad \text{y} \quad R = \frac{N_C}{N_{GT}}. \tag{17}$$

2.3.7. Métricas de detección de texto

Para evaluar la detección de texto en una imagen se utilizan las áreas de los rectángulos delimitadores de las palabras (o líneas) de texto en una imagen.

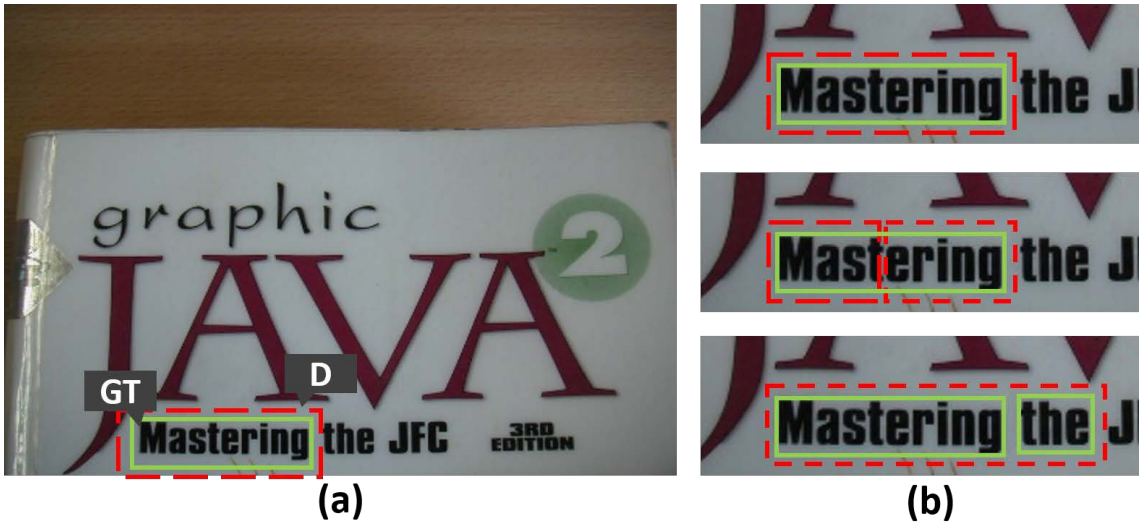


Figura 20. Ejemplo de rectángulo delimitador: (a) D: rectángulo delimitador detectado (línea punteada roja), (b) GT: rectángulo delimitador verdadero (línea continua verde).

Sean $D = \{D_i = (x_i, y_i, w_i, h_i), i = 1, 2, \ldots, N\}$ el conjunto de rectángulos delimitadores detectados y $GT = \{GT_j = (x_j, y_j, w_j, h_j), j = 1, 2, \ldots, M\}$ el conjunto de rectángulos delimitadores verdaderos (manualmente definidos). La coordenada (x_s, y_s) representa la esquina superior izquierda del rectángulo, w_s el ancho y h_s la altura.

Las métricas de sensibilidad (R) y precisión (P) se definen de la siguiente manera: (Wolf y Jolion, 2006):

$$P(GT, D, t_r, t_p) = \frac{\sum_i^N m_D(D_i, GT, t_r, t_p)}{|D|}, \tag{18}$$

y

$$R(GT, D, t_r, t_p) = \frac{\sum_j^M m_D(GT_j, D, t_r, t_p)}{|G|}, \tag{19}$$

donde $|\cdot|$ representa la cardinalidad del conjunto, m_D es una función de correspondencia tal que sus valores dependen de los parámetros $t_r \in [0, 1]$ y $t_p \in [0, 1]$ de sensibilidad y precisión, respectivamente, y las correspondencias entre los rectángu-

los (ver Figura 20(b)) se definen de la siguiente manera (Wolf y Jolion, 2006):

$$m_D(A_l, B, t_r, t_p) = \begin{cases} 1, & si\ \exists!\ B_i \in B \mid r(A_l, B_i) \geq t_r\ \wedge\ p(A_l, B_i) \geq t_p \\ 0.8, & si\ \exists!\ S \subset B \mid \sum_{s_i \in S}\ r(A_l, s_i) \geq t_r\ \wedge\ \forall\ s_i \in S,\ p(A_l, s_i) \leq t_p \\ 0.8, & si\ \exists!\ S \subset B \mid \forall\ s_i \in S,\ r(A_l, s_i) \geq t_r\ \wedge\ \sum_{s_i \in S}\ p(A_l, s_i) \leq t_p \\ 0 & otro\ caso. \end{cases}, \tag{20}$$

con

$$r(a, b) = \frac{area(a \cap b)}{area(b)} \quad y \quad p(a, b) = \frac{area(a \cap b)}{area(a)}. \tag{21}$$

2.3.8. Métricas de reconocimiento de texto

Existen dos métricas comúnmente utilizadas para evaluar el desempeño del método de reconocimiento de texto (Karatzas *et al.*, 2013). La primera métrica es la tasa de reconocimiento a nivel-palabra, donde cada palabra es evaluada como correctamente reconocida si todos sus caracteres son reconocidos correctamente, de caso contrario se considera errónea. La segunda métrica corresponde a la distancia de Levenshtein (Levenshtein, 1966), también conocida como distancia de edición, con pesos iguales para cada una de las operaciones de inserción, eliminación y sustitución. A diferencia de la métrica nivel-palabra, la métrica de distancia de edición puede tolerar errores parciales para cada palabra.

En resumen, en este Capítulo se presentaron los diferentes conjuntos de imágenes y métricas de desempeño utilizadas para la evaluación y comparación de los métodos desarrollados a lo largo de esta tesis, así como para su comparación con los métodos existentes en el estado del arte. Primero, se presento y describió el conjunto de imágenes sintéticas utilizadas para la evaluación de los métodos propuestos bajo distintas degradaciones y distorsiones geométricas. Posteriormente, se presentaron y describieron los conjuntos de imágenes naturales utilizados para la comparación de los métodos propuestos con el estado del arte. Finalmente, se presentaron y definieron las métricas de desempeño que fueron utilizadas a lo largo de este trabajo de investigación.

Capítulo 3. Fundamentos

Desde el trabajo de Attneave (1954), se sabe que la información más importante dentro una imagen se concentra en los puntos del contorno de los objetos que presentan mayor curvatura, como esquinas e intersecciones. Desde entonces, decenas de trabajos se han desarrollado tratando de representar la información de una imagen de manera codificada y única, de tal forma que se pueda despreciar toda aquella información irrelevante, pero que, a su vez, continúe preservando su esencia.

La mayoría de los métodos existentes en el estado del arte, para el procesamiento, representación y análisis de imágenes 2D, utilizan un enfoque basado en la intensidad de los pixeles de la imagen. Este enfoque, analiza la imagen en búsqueda de puntos con gradientes de intensidad altos (Sección 4.1). Lamentablemente, el desempeño de estos métodos se ve afectado por variaciones en los valores de intensidad, tales como iluminación no uniforme, presencia de sombras, ruido, o bajo contraste.

Otro enfoque distinto al de las intensidades de los pixeles es el de la energía local de la imagen. Es decir, podemos localizar características importantes de una imagen analizando el comportamiento de la señal en el dominio frecuencial (Granlund y Knutsson, 2013). En este enfoque, el análisis se realiza utilizando la información de fase y energía local de la señal.

Existen algunas ventajas de los métodos que se basan en el enfoque de la energía local sobre aquellos basados en gradientes de la imagen. (1) se sabe que la información estructural de la imagen se encuentra en la fase de la imagen; (2) los métodos de energía local obtienen una mejor localización de las características de la imagen; (3) los métodos de gradientes sólo obtienen características que representen un cambio de intensidad tipo borde, mientras que los basados en energía puede obtener diferentes tipos de características (borde, línea). Más adelante se explican con mayor detalle (Sección 3.2).

3.1. Importancia de la fase local de una señal

Desde los trabajos de Hubel y Wiesel (1962) se sabe que diferentes grupos de neuronas en la corteza visual biológica, llamadas células simples, responden selectivamente a las barras y bordes en una orientación y ubicación en particular. Además,

existe evidencia psico-física que sugiere la existencia de neuronas V1 selectivas en frecuencia y que el cálculo de las energías en las células complejas se obtiene como una suma de las respuestas de las células simples al cuadrado (Gladilin y Eils, 2015). Estas y otras evidencias nos sugieren que las neuronas del sistema visual humano funcionan como una especie de “filtros” (pasa-banda) capaces de descomponer las señales visuales en frecuencias y seleccionar (activarse) según corresponda, ya sea para reconocer lineas o bordes, o para reconocer orientaciones y/o ubicación.

Por otro lado, se ha demostrado que la fase local contiene la mayor parte de la información estructural importante de una imagen (Oppenheim y Lim, 1981). Además se sabe que la información de fase local es independiente de la amplitud (energía) local de la estructura y sólo cambia cuando es alterada la información estructural de la imagen. Por lo tanto, la información de fase local es invariante a variaciones en las intensidades de los píxeles (Felsberg, 2002). Morrone y Burr (1988) desarrollaron un

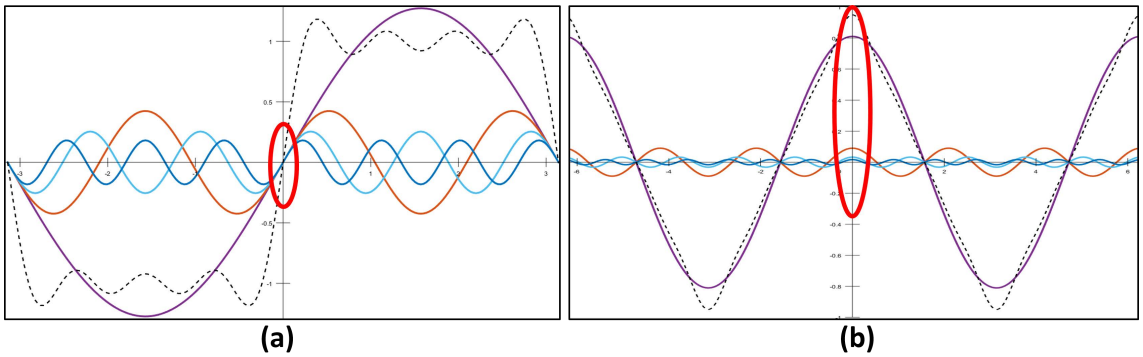


Figura 21. Puntos característica de una señal (círculo rojo) que contienen una mayor congruencia de fase.

modelo bio-inspirado para localizar características de objetos tales como bordes, líneas y sombras llamado modelo de energía local. Este modelo establece que mediante el análisis de la fase local de una señal podemos conocer qué tipo de estructura está contenida en ella.

La mejor manera de entender este modelo es considerando la representación local de los bordes y las líneas. En el espacio de Fourier, la simetría de los bordes y las líneas se refleja en el espectro de fase. Un borde aislado (función impar) se expande como una serie de componentes sinusoidales (eligiendo como origen el punto de cruce de luminancia media), con todos los componentes del coseno con amplitud cero, tal como

se muestra en la Figura 21(a). Una línea aislada (función simétrica) se expande como una serie de componentes de coseno (eligiendo el centro de la línea como origen), con todos los componentes del seno con amplitud cero (ver Figura 21(b)). Entonces, las líneas y los bordes se pueden localizar en los puntos de la señal donde las fases de los componentes de Fourier son muy similares o tienen una alta congruencia de fase (PC, Phase Congruency). Además, el promedio de las fases de las componentes en el punto correspondiente determina la naturaleza de la característica: valores cercanos a cero corresponden a una línea, y valores cercanos a $\pi/2$ corresponden a un borde (Morrone y Owens, 1987; Morrone y Burr, 1988). A continuación se describe formalmente el modelo de energía local y la función de congruencia de fase.

3.2. Modelo de energía local y congruencia de fase

Del análisis de Fourier sabemos que una función periódica $f(x) \in L^2$ puede ser aproximada por su expansión en series de Fourier de la siguiente manera:

$$f(x) = \sum_n A_n \cos(\varphi_n(x)), \tag{22}$$

donde A_n representa la magnitud del n-ésimo componente de Fourier, y $\varphi_n(x) = n\omega x + \phi(x)$ representa la fase local. La congruencia de fase (PC) se define en términos de la expanción en series de Fourier de la señal en un punto x como (Morrone y Owens, 1987):

$$PC(x) = \max_{\overline{\varphi}(x)\in[0,2\pi]} \frac{\sum_n A_n \cos(\varphi_n(x) - \overline{\varphi}(x))}{\sum_n A_n}, \tag{23}$$

con $\overline{\varphi}(x)$ el promedio de la fase local de los componentes de Fourier en x.

Por otro lado, Venkatesh y Owens (1989) demostraron que la suma de los componentes de Fourier en dicho punto es igual a la energía local E de la señal:

$$E(x) = \sum_n A_n \cos(\varphi_n(x) - \overline{\varphi}(x)). \tag{24}$$

Esta relación (Ecuaciones 23 y 24) puede visualizarse geométricamente. En la Figura 22(a) se muestra un diagrama polar de los componentes de Fourier de una señal en un punto x donde todos sus componentes comparten una congruencia de fase má-

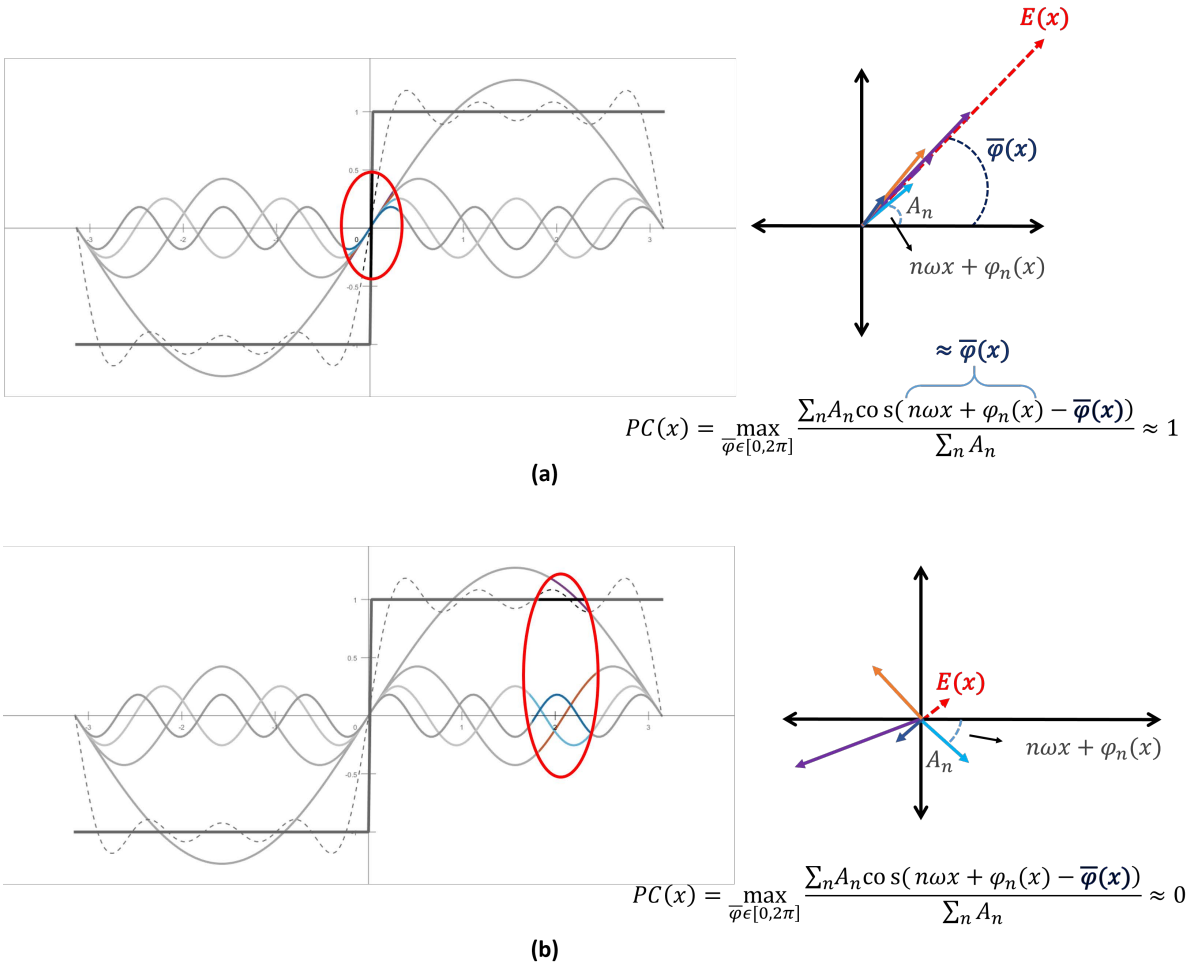


Figura 22. Expanción en series de Fourier de $f(x)$: (a) máxima congruencia de fase ($PC(x) \approx 1$), (b) mínima congruencia de fase ($PC(x) \approx 0$).

xima ($PC(x) \approx 1$), mientras que en 22(b) se puede apreciar el comportamiento de los componentes en cualquier otro punto donde las fases son distintas y la congruencia de fase es mínima o nula ($PC(x) \approx 0$).

En general, las señales contienen diferentes componentes espectrales con diferentes amplitudes y fases. Debido a la superposición de todos los componentes de frecuencia, la información contenida en cada componente se reduce a un valor promedio dominado por los términos de frecuencia de mayor amplitud. Para “separar” la señal, se deben separar las diferentes componentes de frecuencia de la señal. Esto se puede realizar aplicando diferentes filtros que seleccionen sólo una pequeña parte de la información espectral, permitiendo identificar los distintos componentes de la señal. Para señales en una dimensión, la combinación de un filtro con su transformada de Hilbert produce un par de filtros en cuadratura, los cuales pueden utilizarse para estimar

la amplitud local, la fase local y la energía local de una señal (Granlund y Knutsson, 2013).

Sea $\mathrm{H_e} \in L^2$ y $\mathrm{H_o} \in L^2$, dos filtros de igual amplitud espectral pero con fases ortogonales ($\mathrm{H_o}$ denota la Transformada de Hilbert de $\mathrm{H_e}$). La función de energía local se define como (Morrone y Owens, 1987):

$$E(x) = \sqrt{(\mathrm{H_e}(x) * f(x))^2 + (\mathrm{H_o}(x) * f(x))^2}, \tag{25}$$

donde $f(x) \in L^2$ es una señal periódica, y $(*)$ es el operador de convolución.

La función de energía local localiza la posición de las características de la imagen, pero no tiene información sobre el tipo de característica. Para determinar el tipo de característica, es necesario considerar el argumento definido como:

$$\phi(x) = \tan^{-1}(\mathrm{H_e}(x) * f(x), \mathrm{H_o}(x) * f(x)). \tag{26}$$

La congruencia de fase se puede expresar como (Venkatesh y Owens, 1989):

$$PC(x) = \frac{E(x)}{\sum_n A_n}. \tag{27}$$

Desafortunadamente, la función definida $PC(x)$ es altamente sensible al ruido y a la dispersión de frecuencias. Para superar este problema, Kovesi (1999) propuso la siguiente definición de la función de congruencia de fase:

$$PC(x) = \frac{\sum_n W(x) \lfloor E(x) - T \rfloor}{\sum_n A_n(x) + \varepsilon}, \tag{28}$$

donde $W(x)$ es un peso para controlar la dispersión de frecuencias, T es un umbral estimado que controla la cantidad de ruido, y ε es una pequeña constante para evitar la división por cero. Para más detalles ver (Kovesi, 1999, 2000; Kovesi *et al.*, 2002).

En la práctica, para señales en 2D la información de frecuencia local se obtiene a través de bancos de filtros de Gabor orientados (Kovesi *et al.*, 2002), pero este procedimiento es costoso desde el punto de vista computacional. Por otro lado, Felsberg y Sommer (2004) propusieron un esquema llamado espacio-escala de la señal mo-

nogénica, el cual permite obtener la información de fase local, la energía local y la orientación local de una imagen. A continuación se describe brevemente la señal monogénica y el espacio-escala de la señal monogénica.

3.3. La señal monogénica

Como se mencionó anteriormente, para señales en 1D, la señal analítica puede brindar una medida del comportamiento de una señal localmente y comúnmente se utiliza para obtener la energía, fase y orientación local de la señal. Para entender mejor la señal monogénica es necesario describir brevemente la señal analítica.

3.3.1. La transformada de Hilbert

La transformada de Hilbert es, al igual que la transformada de Fourier ($\mathcal{F}$), un mapeo entre dos conjuntos de funciones $\mathcal{H} : S \rightarrow S$, tal que (Granlund y Knutsson, 2013):

$$f_{\mathcal{H}}(x) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{f(\tau)}{\tau - x} d\tau = f * \left(-\frac{1}{\pi x} \right). \tag{29}$$

Es decir, la transformada de Hilbert se puede ver como la convolución de la función f con la función $-\frac{1}{\pi x}$. Sea $F_{\mathcal{H}}(u) = \mathcal{F}\{f_{\mathcal{H}}(x)\}$, en el dominio frecuencial tenemos que:

$$F_{\mathcal{H}}(u) = F(u) \cdot i\, signo(u), \tag{30}$$

con $F(u) = \mathcal{F}\{f(x)\}$. Entonces, la transformada de Fourier de $f_{\mathcal{H}}$ se obtiene de multiplicar F con la unidad imaginaria i y cambiando el signo ($signo(u)$) del resultado para frecuencias negativas. En otras palabras, la Transformada de Hilbert es básicamente un desfase de $\pi/2$.

3.3.2. La Señal analítica

Una vez definida la transformada de Hilbert se define la función analítica f_A que corresponde a la función f como (Granlund y Knutsson, 2013):

$$f_A = f - if_{\mathcal{H}}. \tag{31}$$

En el dominio frecuencial

$$F_A = F \cdot [1 + signo(u)] = 2F \cdot step(u), \tag{32}$$

con $step(u)$ la función escalón. En otras palabras, la señal analítica de f tiene dos veces la energía de la señal original ya que f y $f_{\mathcal{H}}$ son ortogonales y la respuesta de amplitud de la transformada de Hilbert es igual a uno para todas las frecuencias distintas de cero ($|\mathcal{H}(u)| = 1,\ \forall\, u \neq 0$).

Para señales reales, la señal analítica se encuentra bien definida y realiza una "división de identidad". La división de la identidad se cumple si un conjunto de características cumple con la propiedad de invarianza-equivarianza y al mismo tiempo es una descripción única de la señal. La invarianza significa que el valor de una característica no es modificado por una acción que actúa sobre una señal. Por otro lado, la equivarianza significa que hay una dependencia monótona del valor de la característica del parámetro de la acción. Por lo tanto, la señal analítica, considerada en términos de amplitud local $|f_A(x)|$ y fase local $arg[f_A(x)]$, es como una descomposición ortogonal de la información contenida en la señal. La amplitud local representa la energía de una estructura en la vecindad de una determinada posición, por lo tanto, varía con la intensidad local de una estructura, pero es invariable a los cambios en la estructura misma. La fase local es independiente de la intensidad local, pero cambia con la estructura. Es decir, los componentes son independientes (Granlund y Knutsson, 2013).

Para señales en dos dimensiones, la señal analítica no está definida. Afortunadamente Felsberg y Sommer (2001) propusieron la señal monogénica, la cual es una generalización de la señal analítica.

La señal monogénica (F_M), se define como la combinación de la señal 2D (F) y la señal transformada por la transformada de Riesz, la cual a su vez es una generalización 2-D de la transformada de Hilbert, en el dominio frecuencial de la siguiente manera:

$$F_M(u, v) = F(u, v) - i\mathbf{R} \cdot F(u, v), \tag{33}$$

donde $\mathbf{R} = (R_x, R_y)$ es la función de transferencia de la transformada de Riesz de primer

orden en el dominio frecuencial:

$$\mathrm{R_x}(u,v) = i\frac{u}{\sqrt{u^2+v^2}} = \mathcal{F}\left\{\frac{x}{2\pi(x^2+y^2)^{\frac{3}{2}}}\right\}, \tag{34}$$

$$\mathrm{R_y}(u,v) = i\frac{v}{\sqrt{u^2+v^2}} = \mathcal{F}\left\{\frac{y}{2\pi(x^2+y^2)^{\frac{3}{2}}}\right\}. \tag{35}$$

La señal monogénica, al igual que la señal analítica cumple con la propiedad de descomposición ortogonal de la información contenida en la señal, a menos que la señal sea una composición de diferentes señales con diferentes escalas. En este caso, para descomponer la estructura en todas sus señales parciales, sería necesario calcular la señal analítica para anchos de banda infinitamente estrechos. Sin embargo, una buena aproximación de la descomposición de escala se puede realizar utilizando filtros pasa-banda apropiados, con el fin de obtener la localización en ambos dominios, espacial y frecuencial.

Por lo tanto, la señal analítica se combina con un filtro pasa-banda y el conjunto de filtros resultante se denominan filtros en cuadratura. Los filtros en cuadratura con anchos de banda apropiados también permiten descomponer la señal en sus señales parciales y cumplir con la propiedad de invarianza-equivarianza para cada escala. Además, los filtros en cuadratura permiten localizar las señales en los dominios espacial y frecuencial al mismo tiempo.

3.3.3. Espacio-escala de la señal monogénica

El método de espacio-escala manipula estructuras de la imagen en diferentes escalas representando una señal como una familia de parámetros de escala sin cambiar los tamaños de las estructuras. Asumiendo un parámetro de escala no negativo s, Felsberg y Sommer (2004) definen el espacio-escala de Poisson (f_p) como la convolución de la señal (f) con el núcleo de Poisson de la siguiente manera:

$$f_p(x,y,s) = \frac{s}{2\pi(s^2+x^2+y^2)^{2/3}} * f(x,y), \tag{36}$$

y en el dominio frecuencial como:

$$F_p = \mathcal{F}\{f_p(x, y, s)\} = e^{-2\pi s\sqrt{u^2+v^2}} \cdot F(u, v). \tag{37}$$

El parámetro de escala s controla el grado de resolución aplicado a la señal. Es decir, si s tiende a cero, el espacio de escala representa a la misma señal f. Entre mayor sea el valor de s menor serán los detalles de las estructuras. Esto significa que aumentar el valor de s conduce a una disminución en la resolución del espacio-escala $f_p(x, y, s)$.

Para mejorar las características de frecuencias bajas y altas, Felsberg y Sommer (2004) proponen un conjunto de filtros pasa-banda, el cual utiliza una combinación de los parámetros de escala de Poisson s_c y s_f con $s_c > s_f > 0$. Esto puede diseñarse como la diferencia entre dos núcleos de Poisson ($B_{s_0,\lambda,k}$) utilizando el parámetro $s_0 = s_c$ de la siguiente manera:

$$B_{s_0,\lambda,k}(u, v) = \left(e^{-2\pi s_o \lambda^k \sqrt{u^2+v^2}} - e^{-2\pi s_o \lambda^{k-1} \sqrt{u^2+v^2}}\right), \tag{38}$$

donde $\lambda \in (0, 1)$ indica el ancho de banda relativo, s_0 indica la escala más grande, y $k \in \mathbb{N}$ indica el número de filtro. Al conjunto de estos filtros pasa-banda se les conoce como el espacio de diferencias de Poisson (DOP, Differences of Poisson), ya que se construye utilizando las diferencias de dos núcleos de Poisson. Combinando dos filtros pasa-baja con una relación fija de parámetros de escala se obtiene una familia de filtros DOP con un ancho de banda constante. La representación espacio-escala de Poison de la señal (F_{bp}) en el dominio frecuencial es de la forma:

$$F_{bp} = B_{s_0,\lambda,k}(u, v) \cdot F. \tag{39}$$

La representación final del espacio-escala de la señal monogénica en el dominio espacial (f_m) es:

$$f_\mathrm{m} = f_\mathrm{bp}(x, y) + f_\mathrm{x}(x, y) + f_\mathrm{y}(x, y), \tag{40}$$

donde

$$f_\mathrm{bp}(x, y) = \mathcal{F}^{-1}\{F_\mathrm{bp}(u, v)\}, \tag{41}$$

$$f_\mathrm{x}(x, y) = \mathcal{F}^{-1}\{R_\mathrm{x}(u, v) \cdot F_\mathrm{bp}(u, v)\}, \tag{42}$$

$$f_{\mathrm{y}}(x,y) = \mathcal{F}^{-1}\{R_{\mathrm{y}}(u,v) \cdot F_{\mathrm{bp}}(u,v)\}. \tag{43}$$

Finalmente, la energía local (E), la orientación local (θ), la dirección local (θ_d) y la fase local[1] (φ) puede calcularse como:

$$E(x,y) = \sqrt{f_{\mathrm{p}}^2(x,y) + f_{\mathrm{x}}^2(x,y) + f_{\mathrm{y}}^2(x,y)}, \tag{44}$$

$$\theta(x,y) = tan^{-1}\left(\frac{f_{\mathrm{y}}(x,y)}{f_{\mathrm{x}}(x,y)}\right), \tag{45}$$

$$\theta_d(x,y) = atan2\left(\frac{f_{\mathrm{y}}(x,y)}{f_{\mathrm{x}}(x,y)}\right), \tag{46}$$

$$\varphi(x,y) = atan2\left(\frac{f_{\mathrm{x}}(x,y) + f_{\mathrm{y}}(x,y)}{f_{\mathrm{p}}(x,y)}\right). \tag{47}$$

La Figura 23 muestra el diagrama de bloques para calcular el espacio-escala de la señal monogénica.

En resumen, en este Capítulo se presentan brevemente los fundamentos que son base de este trabajo de investigación. Se argumenta la importancia del enfoque basado en la información de fase local y se describe un marco teórico, conocido como espacio-escala de la señal monogénica, para la extracción de dicha información. Finalmente, se definieron los conceptos de energía local, orientación local, fase local y congruencia de fase local, los cuales son utilizados para el desarrollo de esta tesis.

[1] Note que la función $atan2(y/x) = sign(y) = sign(y) \cdot tan^{-1}(|y|/x)$, en la que el signo del factor y indica la dirección de rotación.

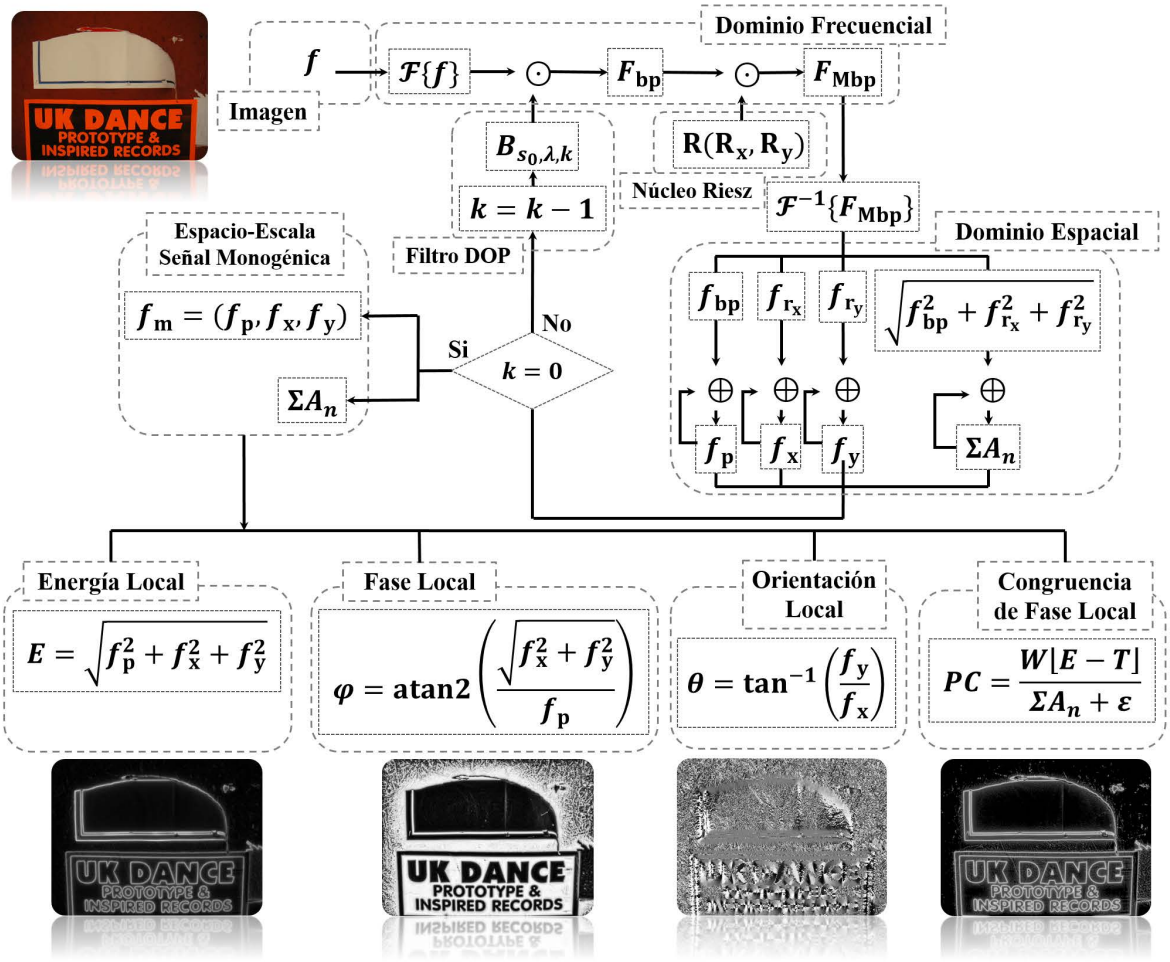


Figura 23. Diagrama espacio-escala de la señal monogénica.

Capítulo 4. Detector y descriptor LUIFT

La detección y descripción de características son tareas de bajo nivel utilizadas en muchas aplicaciones de reconocimiento de patrones y de visión por computadora, como por ejemplo, la clasificación y recuperación de imágenes (Deselaers *et al.*, 2008; Liu y Bai, 2012), la estimación del flujo óptico (Fortun *et al.*, 2015), el seguimiento de objetos (Tang *et al.*, 2014), sistemas biométricos (Jain *et al.*, 2011), el registro de imágenes y la reconstrucción en 3D (Moreels y Perona, 2007), por mencionar algunos.

La tarea de detección de características locales consiste en encontrar los “puntos característicos” (puntos, líneas, regiones, etc.) en la imagen. Estos puntos deben satisfacer ciertas propiedades tales como: distintividad, cantidad, localidad, precisión y, la más importante, repetibilidad (estables en número y posición bajo degradaciones y transformaciones) (Tuytelaars y Mikolajczyk, 2008). Una vez detectadas las características de la imagen, estas son representadas de una manera distintiva considerando un vecindario alrededor de cada punto, el cual es codificado en un vector, conocido como “descriptor” de la característica. Finalmente, los descriptores de diferentes imágenes son correspondidos utilizando distancias comúnes tales como la distancia Euclidiana o de Mahalanobis.

Es deseable que el comportamiento de los descriptores sea invariable a perspectiva, desenfoque y transformaciones afínes (Lowe, 1999; Tola *et al.*, 2008; Bay *et al.*, 2006; Alcantarilla *et al.*, 2012; Verdie *et al.*, 2015); pero también es necesario que sean robustos a otro tipo de degradaciones tales como ruido e iluminación no uniforme. Sin embargo, estas dos últimas condiciones no han sido completamente resueltas, incluso aún cuando son problemas comunes en aplicaciones reales. Por lo que, la iluminación no uniforme y el ruido siguen siendo desafíos que disminuyen el desempeño de los métodos existentes en el estado del arte.

4.1. Detectores y descriptores de características

Los primeros trabajos sobre puntos de característica de la imagen comenzaron con el trabajo de Attneave (1954), sobre percepción y puntos de curvatura alta como puntos de interés (esquinas e intersecciones). Desde entonces, se han desarrollado varias técnicas para la detección de características basándose en este principio, tales co-

mo los métodos basados en curvatura de contorno (Tuytelaars y Mikolajczyk, 2008; Papari y Petkov, 2011), técnicas de detección de regiones (Mikolajczyk *et al.*, 2005), enfoques diferenciales (Tuytelaars y Mikolajczyk, 2008; Mikolajczyk y Schmid, 2005), técnicas basadas en variaciones de intensidad (Smith y Brady, 1997; Rosten y Drummond, 2006), y recientemente, métodos basados en aprendizaje (Simo-Serra *et al.*, 2015; Yi *et al.*, 2016; Levi y Hassner, 2016).

El detector de puntos esquina Harris (Harris y Stephens, 1988), que es una mejora del enfoque Moravec (1979), es uno de los primeros y más utilizados detectores de puntos esquina, el cual describe la distribución de gradientes en un vecindario local de un punto basado en la matriz de segundo momento. Los puntos de característica se obtienen en los puntos donde el gradiente local varía significativamente en dos direcciones. Al igual que la matriz de Harris, la matriz Hessiana (Beaudet, 1987) se basa en el mismo principio y se construye utilizando la expansión de Taylor pero de segundo orden, codificando la información de la estructura de la imagen. Recientemente, se propuso un detector de esquinas basado en el detector Harris, llamado HarrisZ (Bellavia *et al.*, 2011). En este trabajo se utiliza la normalización $Z(x) = \frac{x-\mu}{\sigma}$ para adaptar la función de respuesta del detector, mejorando su desempeño.

Los detectores SUSAN (SUSAN, Smallest Univalue Segment Assimilating Nucleus) (Smith y Brady, 1997) y recientemente, FAST (FAST, Features from Accelerated Segment Test) (Rosten y Drummond, 2006) son también técnicas basadas en la intensidad de la imagen. Estos métodos, obtienen puntos de características rápidos asociando puntos de la imagen con brillo similar en un área local. El detector FAST se basa en el detector SUSAN, pero a diferencia de este, el detector FAST utiliza árboles de decisión eficientes para evaluar los valores de intensidad de los píxeles.

El descriptor SIFT (SIFT, Scale Invariant Feature Transform) (Lowe, 1999, 2004) utiliza una aproximación del Laplaciano de la Gaussiana (LOG, Laplacian of Gaussian) e histogramas de gradiente orientado (HOG, Histogram of Oriented Gradients) (Dalal y Triggs, 2005) para lograr la invarianza de escala y rotación, respectivamente. Hasta ahora, el descriptor SIFT es el descriptor más popular del estado del arte debido a su eficiencia en la detección de características y en la adaptación a los cambios de escala y rotación de la imagen. Por esta razón, diferentes variaciones del descriptor SIFT han sido propuestas. Los descriptores SURF (SURF, SURF) (Bay *et al.*, 2006, 2008) y KAZE

(Alcantarilla *et al.*, 2012) son un par de ejemplos. A diferencia del método SIFT, el descriptor SURF utiliza filtros de tipo Haar e imágenes integrales para mejorar el tiempo de procesamiento a expensas del desempeño del método; mientras que el descriptor KAZE se basa en un espacio de escala no lineal mejorando el desenfoque local adaptativo en la construcción de espacio de escala. El CenSurE (CenSurE, Center Surround Extremas) (Agrawal *et al.*, 2008) se basa en la estimación del LOG utilizando filtros simples de centro-alrededor e imágenes integrales para tareas en tiempo real. El descriptor Daisy (Tola *et al.*, 2008) está inspirado en los descriptores SIFT y GLOH (Mikolajczyk y Schmid, 2005) pero diseñado eficientemente reemplazando sumas ponderadas por sumas de convoluciones.

También se han propuesto descriptores binarios. FREAK (FREAK, Fast Retina Keypoint) (Alahi *et al.*, 2012), BRIEF (BRIEF, Binary Robust Independent Elementary Features) (Calonder *et al.*, 2010), y BRISK (BRISK, Binary Robust Scalable Keypoints) (Leutenegger *et al.*, 2011) son algunos de ellos. Básicamente, realizan comparaciones de intensidad por pares dentro de una vecindad de la imagen y utilizan la distancia de Hamming para realizar una correspondencia rápida de características.

Aunque todos los métodos mencionados anteriormente proporcionan resultados satisfactorios para algunas transformaciones geométricas (rotación y escalamiento), Estos descriptores se basan en las diferencias de intensidades de los píxeles de la imagen, por lo que son sensibles a la variación de la iluminación y a la degradación del ruido.

Para obtener descriptores robustos a cambios de intensidad, en los últimos años se han propuesto nuevos descriptores. El descriptor DaLI (DaLI, Deformation and Light Invariant) (Simo-Serra *et al.*, 2015) fue desarrollado para transformaciones no rígidas y cambios de iluminación. Parches de la imagen 2D se consideran como superficies 3D y se describen en términos de una especie de “firma” utilizando la función de difusión de calor. Posteriormente, para la reducción dimensional del descriptor, se aplica un análisis de componentes principales (PCA, Principal Components Analysis). Sin embargo, el descriptor DaLI no es invariante a transformaciones de escala y rotación, además de ser complejo debido al cálculo de valores propios para la ecuación de difusión de calor. Por otro lado, los descriptores TILDE (TILDE, Temporally Invariant Learned DEtector) (Verdie *et al.*, 2015) y LIFT (LIFT, Learned Invariant Feature Transform) (Yi *et al.*, 2016)

consideran un método de aprendizaje para la detección y descripción de características. Básicamente, el detector utiliza entrenamiento para obtener aquellas características que permanecen estables bajo diferentes condiciones. Sin embargo, es necesaria una etapa previa de entrenamiento y una colección de parches de la imagen. Finalmente, el descriptor LIOP (LIOP, Local Intensity Order Pattern) (Wang *et al.*, 2011b) se basa en el orden de los valores de intensidad, asumiendo el principio de que el orden relativo de las intensidades de píxeles permanece inalterado con los cambios monótonos de intensidad. Sin embargo, no se consideran las variaciones de iluminación no uniforme.

Los detectores y descriptores mencionados anteriormente comparten una característica en común, todos se basan en los cambios de las intensidades de los píxeles de la imagen. Por lo que, si el cambio en las intensidades de los píxeles de la imagen no es monótono, como es el caso de la iluminación no uniforme, sombras o ruido, el desempeño de los detectores y descriptores antes mencionados decrece o fallan por completo. Por ello, en este trabajo de investigación, proponemos un detector y descriptor de características basado en la fase local de la imagen.

4.2. Detector y descriptor LUIFT

En esta sección se describe el detector y descriptor LUIFT (LUIFT, LUminance Invariant Feature Transform) propuesto. Como se mencionó anteriormente, los detectores y descriptores existentes en el estado del arte se encuentran basados en diferencias entre las intensidades de los pixeles de la imagen, por lo que son sensibles a iluminación no uniforme. Para evitar esto, en este trabajo de investigación proponemos un método basado en la información de fase local de la imagen, la cual es invariante a los cambios en las intensidades de los pixeles. Básicamente, el detector de características propuesto se construye utilizando una modificación del detector Harris y el espacio-escala de la señal monogénica; mientras que el descriptor propuesto se construye utilizando una modificación de los histogramas de gradiente orientado y la congruencia de fase. A continuación se describen a detalle cada uno de ellos.

4.2.1. Detector de características

Primero, utilizando el espacio-escala de la señal monogénica (Sección 3.3.3) con el conjunto de filtros DOP (con $s_0 = 3,\ \lambda = 0.5$ y $k = 3$), se calculan los componentes de la señal monogénica $f_{\mathrm{m}} = (f_{\mathrm{bp}}, f_{\mathrm{x}}, f_{\mathrm{y}})$ y la sumatoria de las amplitudes $\sum_n A_n(x, y)$ (ver Figura 23). Recuerde que al aumentar el número de filtros en el conjunto DOP (k), serán reveladas un mayor número de características de la imagen.

Posteriormente, con el fin de obtener puntos característicos de la imagen, se utiliza una modificación del detector de puntos esquina Harris (Harris y Stephens, 1988), el cual será descrito brevemente a continuación.

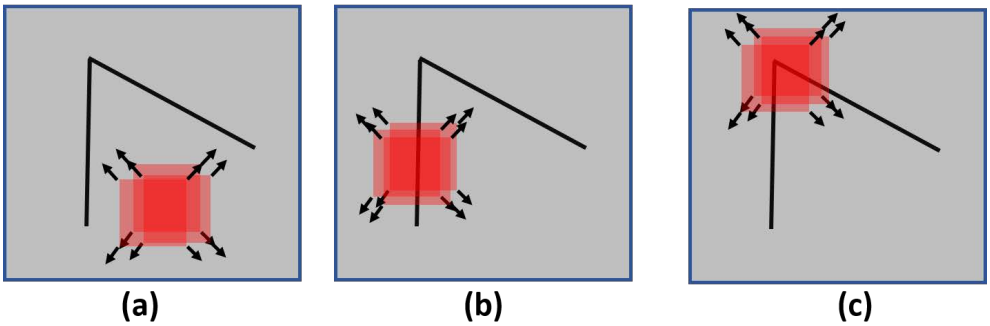


Figura 24. Análisis de intensidades dentro de una ventana según la región contenida. (a) Región plana, sin cambios en todas las direcciones; (b) borde, sin cambio en la dirección del borde; (c) esquina, cambio significativo en todas las direcciones.

Sea I una imagen en escala de grises, tal como muestra la Figura 24. Considérese una ventana local w en la imagen I, la cual determina el cambio de intensidad promedio que resulta de pequeños desplazamientos en varias direcciones de la ventana en la imagen (ver Figura 24). Si la zona evaluada es homogénea, es decir, aproximadamente constante en intensidad, entonces todos los pequeños desplazamientos resultan en un cambio pequeño (Figura 24(a)). Si la ventana contiene un borde, entonces los desplazamientos sobre el borde resultarán en un pequeño cambio, pero los desplazamientos perpendiculares al borde resultarán en un cambio grande (Figura 24(b)). Si la ventana contiene una esquina, entonces todos los desplazamientos resultarán en un cambio grande (Figura 24(c)). Por lo tanto, una esquina puede ser detectada cuando el cambio mínimo producido por cualquier desplazamiento que sea mayor. Este método

fue formulado por Moravec (1979) como:

$$D(x,y) = \sum_{m,n} w(m,n)|I(m+x,n+y) - I(m,n)|^2. \tag{48}$$

El operador de Harris y Stephens (1988) mejora el método de Moravec (1979) aproximando la ventana desplazada utilizando una expansión en series de Taylor

$$I(m+x,n+y) = I(m,n) + I_x(m,n)x + I_y(m,n)y, \tag{49}$$

obteniendo la siguiente expresión:

$$\begin{aligned} D(x,y) &= \textstyle\sum_{m,n} w(m,n)|I(m+x,n+y) - I(m,n)|^2 \\ &\approx \textstyle\sum_{m,n} w(m,n)|[I(m,n) + I_x(m,n)x + I_y(m,n)y] - I(m,n)|^2 \\ &= \textstyle\sum_{m,n} w(m,n)[I_x^2(m,n)x^2 + 2I_x(m,n)I_y(m,n)xy + I_y^2(m,n)y^2]. \end{aligned} \tag{50}$$

En su forma matricial tenemos:

$$D(x,y) \approx \begin{bmatrix} x & y \end{bmatrix} \left(\sum_{m,n} w(m,n) \begin{bmatrix} I_x^2(m,n) & I_x(m,n)I_y(m,n) \\ I_x(m,n)I_y(m,n) & I_y^2(m,n) \end{bmatrix} \right) \begin{bmatrix} x \\ y \end{bmatrix}. \tag{51}$$

Finalmente, la matriz de Harris, $\mathbf{M}_{\mathrm{H}}$, se define como:

$$\mathbf{M}_{\mathrm{H}} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}, \tag{52}$$

donde I_x y I_y son las derivadas parciales de la imagen I y $\langle \cdot \rangle$ denota el promedio dentro de la ventana w.

Considerando el espacio-escala de la señal monogénica $f_m = (f_{bp}, f_x, f_y)$, proponemos sustituir las derivadas parciales, I_x y I_y, de la matriz de Harris ($\mathbf{M}_{\mathrm{H}}$) por los componentes de la señal monogénica, f_x y f_y, definiendo una nueva matriz $\mathbf{M}_{\mathrm{m}}$ de la siguiente manera:

$$\mathbf{M}_{\mathrm{m}} = \begin{bmatrix} \langle f_x^2 \rangle & \langle f_x f_y \rangle \\ \langle f_x f_y \rangle & \langle f_y^2 \rangle \end{bmatrix}. \tag{53}$$

Dada la naturaleza de la transformada de Riesz, esta se puede interpretar como una versión suavizada del gradiente de la imagen (Unser *et al.*, 2009). La Figura 25(a)

muestra el gradiente de la imagen en la dirección x y y, mientras que la Figura 25(b) muestra los componentes de la señal monogénica f_x y f_y.

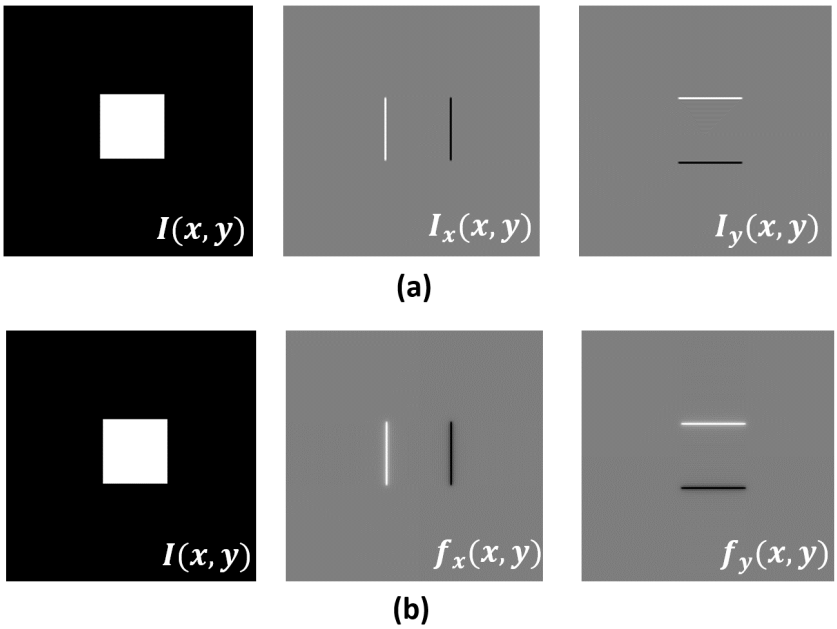


Figura 25. Gradiente vs componentes de la señal monogénica.

Posteriormente, la función de detección de esquinas definida en (Harris y Stephens, 1988) es utilizada para obtener las características de la imagen,

$$\mathbf{M}_{\mathrm{c}}(x,y) = \det(\mathbf{M}_{\mathrm{m}}(x,y)) - \beta \cdot \mathrm{traza}^2(\mathbf{M}_{\mathrm{m}}(x,y)), \tag{54}$$

donde β es un parámetro de sensibilidad, comúnmente utilizado $\beta = 0.04$.

Las características obtenidas $\mathbf{M}_{\mathrm{c}}$ son ponderadas por su correspondiente valor de congruencia de fase PC, con el fin de extraer las características con congruencia de fase alta ($\mathbf{M}_{\mathrm{pc}}$); es decir,

$$\mathbf{M}_{\mathrm{pc}}(x,y) = \mathbf{M}_{\mathrm{c}}(x,y) \cdot PC(x,y). \tag{55}$$

La función de congruencia de fase definida en la Ecuación 28, puede calcularse para

una señal en dos dimensiones de la siguiente manera:

$$PC(x,y) = \frac{W(x,y)\lfloor E(x,y) - T \rfloor}{\sum_n A_n(x,y) + \varepsilon}, \tag{56}$$

donde la energía $E(x,y) = \sqrt{f_{\text{bp}}^2 + f_{\text{x}}^2 + f_{\text{y}}^2}$ y la sumatoria de las amplitudes $\sum_n A_n(x,y)$, se obtienen a partir del espacio-escala de la señal monogénica. El peso de la dispersión de frecuencias $W(x,y)$ y el umbral de ruido T se calculan como en (Kovesi, 1999).

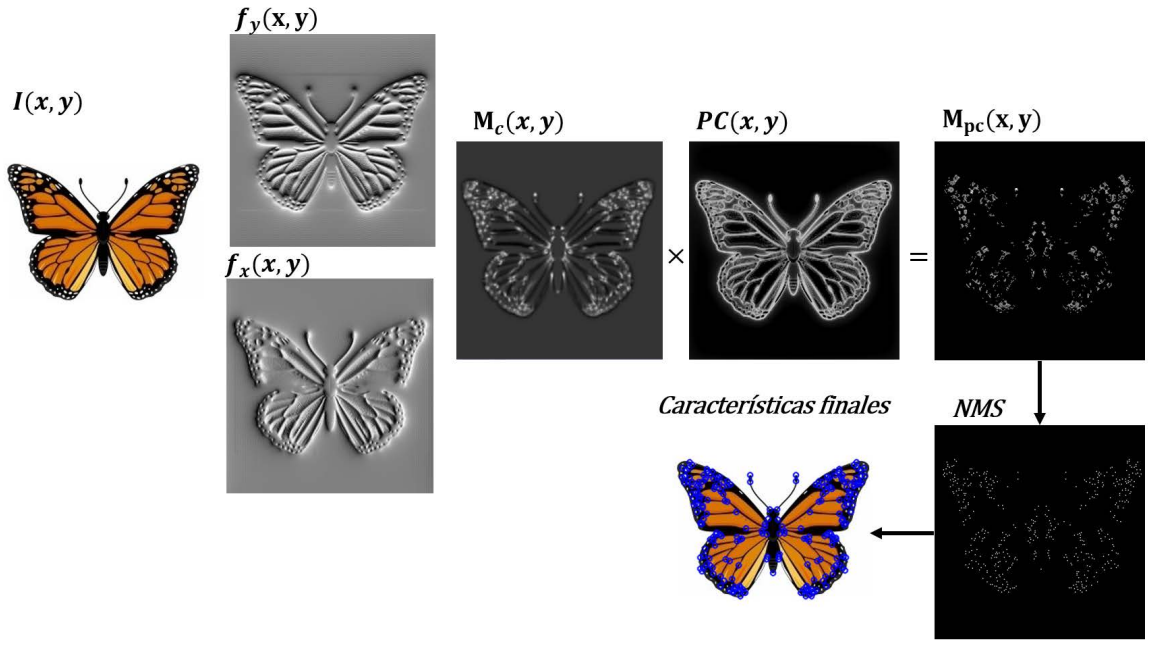


Figura 26. Diagrama del detector de características propuesto.

Finalmente, se aplica una umbralización seguida de un algoritmo de supresión no máxima (NMS, Non-Maximum Suppression) (Lowe, 2004) de tamaño 3×3 para obtener las características finales. Dado que el valor de la congruencia de fase PC(x,y) indica la importancia de las características detectadas (ver sección. 3.2), se define un valor de umbral para controlar el número de características a ser preservadas o eliminadas. Un umbral cercano a uno mantiene sólo aquellas características que pertenecen a líneas o bordes nítidos en la imagen. Al cambiar el valor umbral, se pueden preservar características importantes que pertenecen a bordes y líneas con bajo contraste, alto brillo o degradaciones de borrosidad. Para nuestros experimentos, se definió experimentalmente un umbral de 0.3. La Figura 26 ilustra un diagrama del detector de

características propuesto.

4.2.2. Descriptor de características

Debido a que los histogramas de gradiente orientado (Dalal y Triggs, 2005), muestran robustez a pequeñas deformaciones geométricas tales como escalamientos y rotaciones, se propone una modificación del descriptor HOG de la siguiente manera. Para cada punto característica detectado, se selecciona un vecindario de 16×16 alrededor de cada característica, el cual es ponderado por un núcleo gaussiano ($\sigma = 1.5$). Posteriormente, dicho vecindario se divide en celdas de 4×4. Al igual que para el descriptor HOG, en cada celda se calcula un histograma de congruencia de fase (HOPC, Histogram of Phase Congruency) considerando *nbins*. A diferencia del HOG, el HOPC propuesto utiliza la dirección local θ_d (Ecuación 46) y el valor de la la congruencia de fase local (Ecuación 56) para crear los histogramas que describen las vecindades, de tal forma que la cantidad que se agrega a cada *bin* depende del valor de *PC* de cada punto de la vecindad, de la siguiente manera:

$$HOPC(bin_{\theta_{\mathrm{d}}(\mathrm{x,y})}) = HOPC(bin_{\theta_{\mathrm{d}}(\mathrm{x,y})}) + PC(x, y), \tag{57}$$

donde

$$bin_\theta = \lfloor (nbins/360) \cdot \theta_{\mathrm{d}}(x, y) \rfloor. \tag{58}$$

La Figura 27 ilustra la construcción del descriptor propuesto.

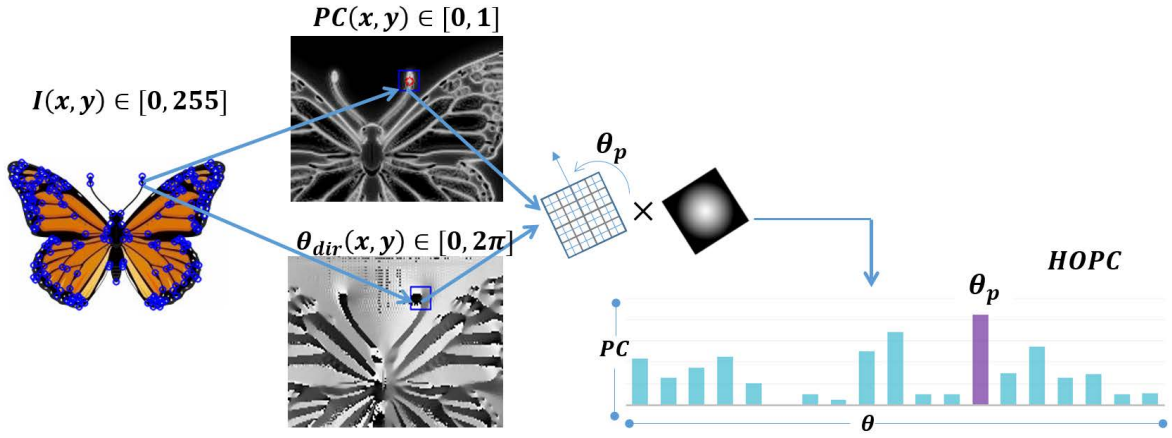


Figura 27. Diagrama del descriptor propuesto.

Por otro lado, para considerar aquellas características que caen cerca de la frontera entre dos *bins* adyacentes, se propone lo siguiente. Sea r_θ el residuo del módulo (mod),

$$r_\theta = \theta_{\mathrm{d}}(x, y) \mathrm{mod} \left(\frac{360}{nbins} \right). \tag{59}$$

Si $|r_\theta| < \epsilon$ o $|\frac{360}{nbins} - r_\theta| < \epsilon$ están cerca de cero, significa que $\theta_{\mathrm{d}}(x, y)$ está cerca de la frontera entre dos bins adyacentes. Por lo tanto, $\theta_{\mathrm{d}}(x, y)$ podría asignarse a una de las ubicaciones o dividirse entre las ubicaciones. Por lo que se asigna la mitad del valor de $PC(x, y)$ a cada una de las ubicaciones adyacentes.

Además, para proporcionar invariabilidad a la rotación, cada histograma se normaliza usando la orientación prominente (θ_{p}) (Lowe, 2004). Finalmente, los histogramas son concatenados y normalizados (usando la norma Euclidiana) para crear el descriptor final.

4.3. Resultados experimentales

En esta sección se presenta y analiza experimentalmente el desempeño del descriptor propuesto LUIFT. Tres versiones del descriptor LUIFT son evaluadas: LUIFT_8, LUIFT_36 y LUIFT_64, las cuales utilizan 8, 36 y 64 bins, respectivamente. El desempeño del método LUIFT se compara con los detectores y descriptores: FAST (Rosten y Drummond, 2006), STAR (Agrawal *et al.*, 2008), SIFT (Lowe, 1999), SURF (Bay *et al.*, 2006), KAZE (Alcantarilla *et al.*, 2012), HARRISZ (Bellavia *et al.*, 2011), DAISY (Tola *et al.*, 2008), y LIOP (Wang *et al.*, 2011b). Todas las simulaciones se realizaron utilizando la biblioteca C^{++} y OpenCV[1], con la excepción del descriptor LIOP, para el cual se utilizó Matlab haciendo uso de la biblioteca VLFeat[2]. Los resultados se presentan mediante la curva sensibilidad vs 1-precisión (ver Sección 2.3.5), el error de traslape y la medida de correspondencia (Sección 2.3.4).

4.3.1. Evaluación imágenes sintéticas

Con el fin de evaluar el desempeño del detector y descriptor LUIFT propuesto, se creó un conjunto de imágenes sintéticas en escala de grises (rango de 0 a 255), utili-

[1] http://opencv.org/
[2] http://www.vlfeat.org/

zando las transformaciones y degradaciones descritas en la Sección 2.1. El conjunto

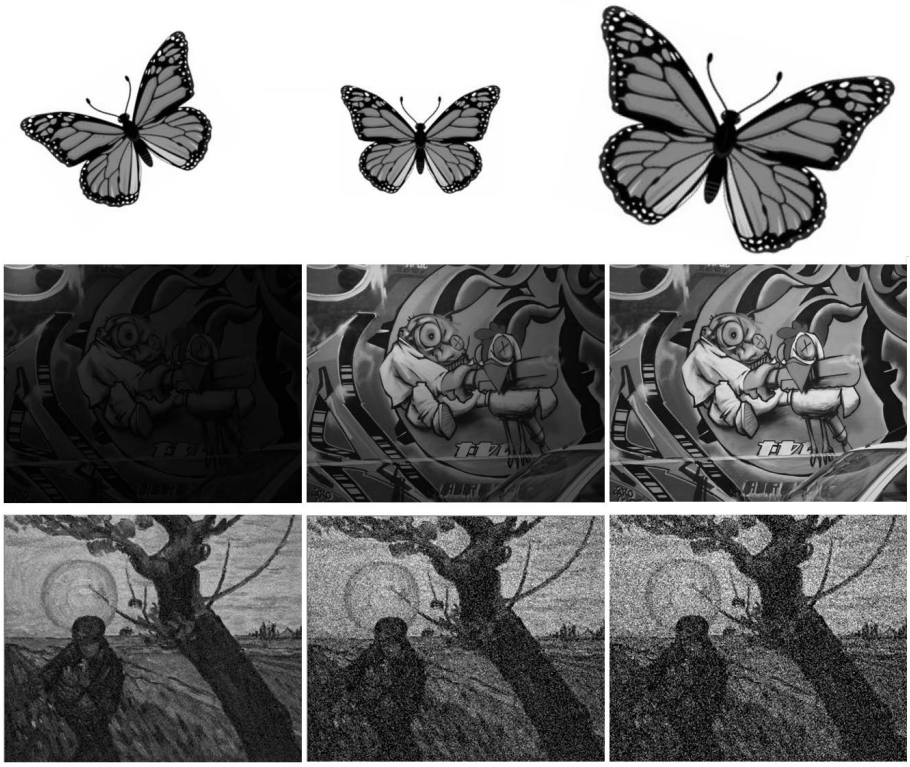


Figura 28. Ejemplo de imágenes de conjuntos de imágenes sintéticas. De arriba a abajo: escena de mariposa bajo rotación y distorsiones de escala; escena de graffiti bajo variaciones de iluminación no uniforme; y escena de Gogh bajo degradaciones de ruido aditivo.

de imágenes sintéticas contiene 7,254 imágenes, de las cuales 2,106 corresponden a tres escenas diferentes (mariposa, Gogh y graffiti) escaladas (6 escalamientos) y rotadas (13 rotaciones) bajo iluminación no uniforme (9 variaciones); 2,106 corresponden a tres escenas diferentes escaladas y rotadas bajo ruido Gaussiano aditivo (9 variaciones); y 3,042 imágenes corresponden a tres escenas diferentes escaladas y rotadas bajo cambios de brillo y contraste (13 variaciones). La tabla 2 muestra los parámetros utilizados para generar las imágenes sintéticas. La Figura 28 muestra ejemplos de imágenes del conjunto.

Utilizando el conjunto de imágenes sintéticas, se realizaron cuatro experimentos para evaluar el desempeño del descriptor propuesto LUIFT bajo variaciones de iluminación no uniforme, ruido, brillo y contraste. El desempeño del método propuesto se compara con el de los métodos comunes SIFT (Lowe, 2004) y SURF (Bay *et al.*, 2006),

Tabla 2. Parámetros utilizados para generar el conjunto de imágenes sintéticas.

Degradación	Tamaño de paso	Rango
Iluminación (ρ)	10	$[10, 50]$
Ruido (σ)	5	$[0, 40]$
Brillo (b)	30	$[-90, 90]$
Contraste (c)	0.3	$[0.5, 2]$
Distorsión		
Rotación	5	$[-30, 30]$
Escalamiento	0.1	$[0.8, 1.3]$

en términos derepetibilidad y sensibilidad (Sección 2.3.3) de las características, así como la medida de correspondencia (Sección 2.3.4) de los descriptores. Los resultados que se presentan a continuación son el resultado promedio de las evaluaciones realizadas.

Nuestro primer experimento para condiciones de iluminación no uniforme se llevó a cabo variando el parámetro de distancia ρ en las imágenes sintéticas de prueba (escenas giradas y escaladas). La Figura 29 muestra los resultados obtenidos para la degradación por iluminación no uniforme en términos de repetibilidad, sensibilidad y medida de correspondencia. Como puede observarse, todos los métodos evaluados son capaces de detectar correctamente cierto número de características de las imágenes sintéticas. Sin embargo, el desempeño de la detección de características, así como el desempeño de la correspondencia de características de los métodos SIFT y SURF, disminuye considerablemente cuando la iluminación se vuelve no uniforme. Note que el método propuesto supera significativamente a los métodos evaluados en escenas con baja iluminación, alcanzando hasta un 50 % de mejora.

El siguiente experimento consiste en evaluar el desempeño de los métodos bajo la degradación de ruido Gaussiano mediante la variación del valor de desviación estándar, σ, en las imágenes sintéticas de prueba (escenas giradas y escaladas). La Figura 30 muestra los resultados obtenidos para la degradación producida por ruido aditivo en términos de la repetibilidad, sensibilidad y medida de correspondencia.

Como es de esperarse, el desempeño del método SIFT disminuye a medida que aumenta la varianza del ruido, mientras que el desempeño del detector SURF permanece

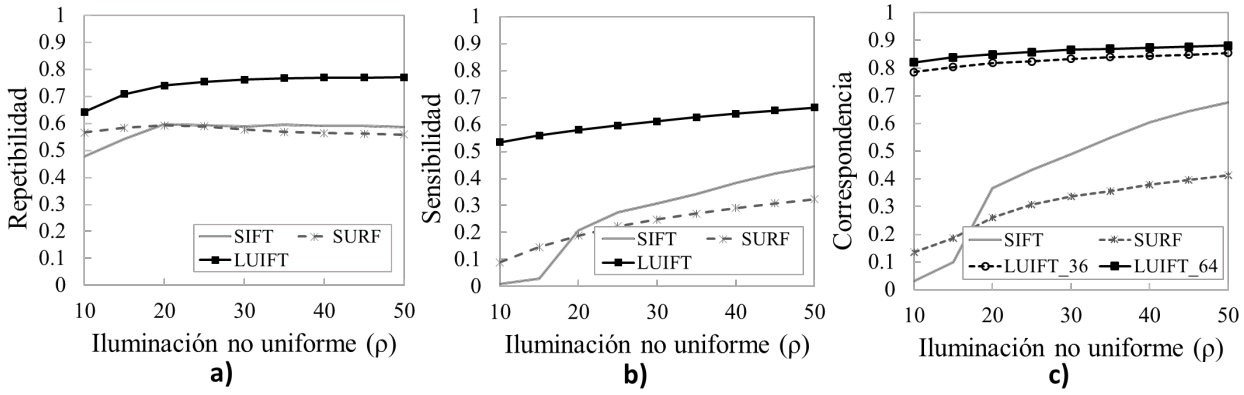


Figura 29. Desempeño de los métodos evaluados en imágenes sintéticas rotadas y escaladas bajo variaciones de iluminación no uniforme. a) Porcentaje de características que permanece estable bajo variaciones de iluminación; b) porcentaje de características correctamente detectadas con respecto a la imagen original; c) desempeño de los descriptores de características.

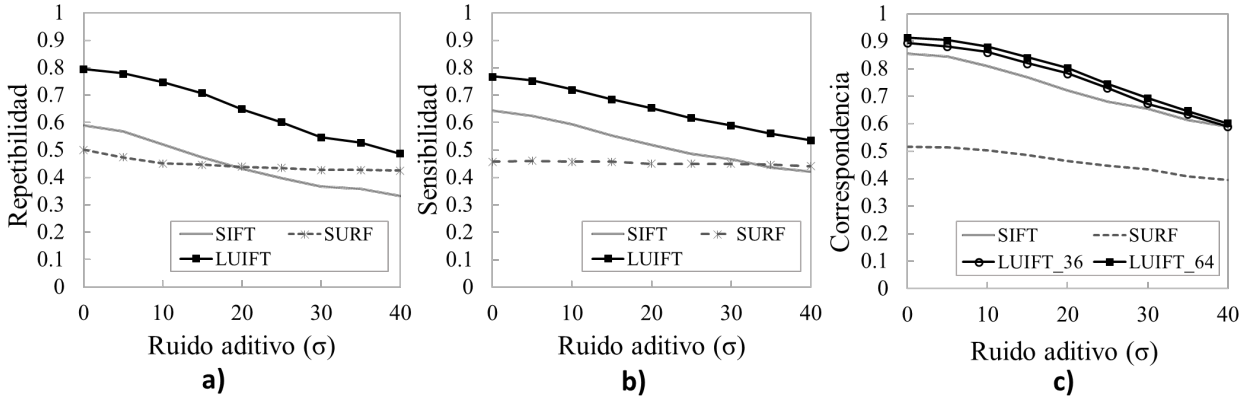


Figura 30. Desempeño de los métodos evaluados en imágenes sintéticas rotadas y escaladas bajo degradaciones de ruido aditivo. a) Porcentaje de características que permanecen estables bajo variaciones de ruido aditivo; b) porcentaje de características correctamente detectadas con respecto a la imagen original; c) desempeño de los descriptores de características.

estable. En términos de la medida de repetibilidad, el desempeño de los detectores SIFT y SURF es casi un 20 % peor que el del método LUIFT propuesto; mientras que el método SURF muestra el peor desempeño con respecto a la medida de correspondencia de los tres descriptores evaluados.

Los experimentos finales de variaciones de brillo y contraste se llevaron a cabo variando los parámetros *c* y *b* en las imágenes sintéticas de prueba (escenas giradas y escaladas). Las Figura 31 y 32 muestran los resultados obtenidos de las variaciones de contraste y brillo en términos de la repetibilidad, sensibilidad y medida de correspon-

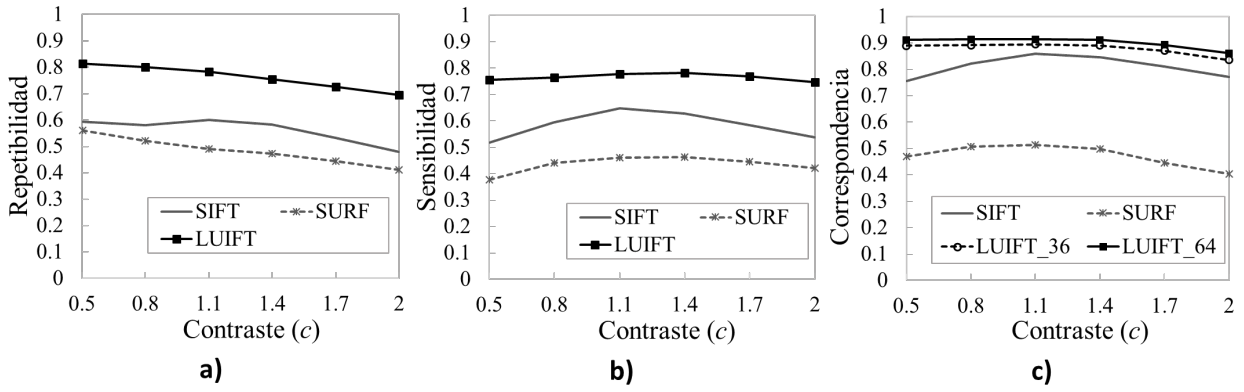


Figura 31. Desempeño de los métodos evaluados en imágenes sintéticas rotadas y escaladas bajo variaciones de contraste. a) Porcentaje de características que permanecen estables bajo variaciones de contraste; b) porcentaje de características correctamente detectadas con respecto a la imagen original; c) desempeño de los descriptores de características.

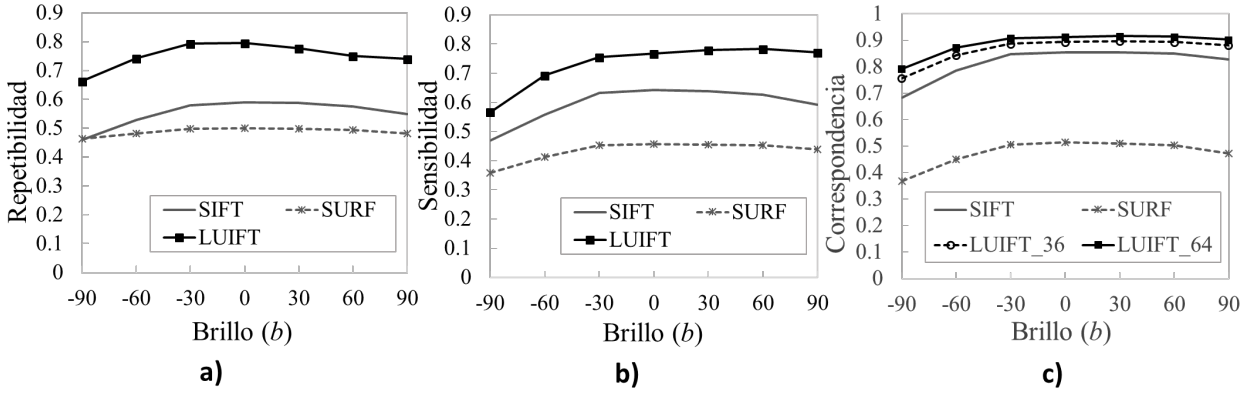


Figura 32. Desempeño de los métodos evaluados en imágenes sintéticas rotadas y escaladas bajo variaciones de brillo. a) Porcentaje de características que permanecen estables bajo variaciones de brillo; b) porcentaje de características correctamente detectadas con respecto a la imagen original; c) desempeño de los descriptores de características.

dencia, respectivamente. Los resultados obtenidos muestran que el método SIFT es menos sensible a los cambios de iluminación monótona. Sin embargo, el método propuesto muestra el mejor desempeño en términos de repetibilidad de características y correspondencia de descriptores.

Con el fin de comparar el desempeño del detector y descriptor propuesto con los métodos del estado del arte, se realizaron evaluaciones en los conjuntos de imágenes reales PHOS (Sección 2.2.1) y OFFICE (Sección 2.2.2).

4.3.2. Evaluación en los conjuntos OFFICE y PHOS

El conjunto de imágenes OFFICE contiene dos escenas diferentes llamadas “pasillo” y “escritorio”. Para cada conjunto de imágenes, se evaluó el desempeño del detector y descriptor propuesto y los métodos del estado del arte. La Figura 33 muestra el

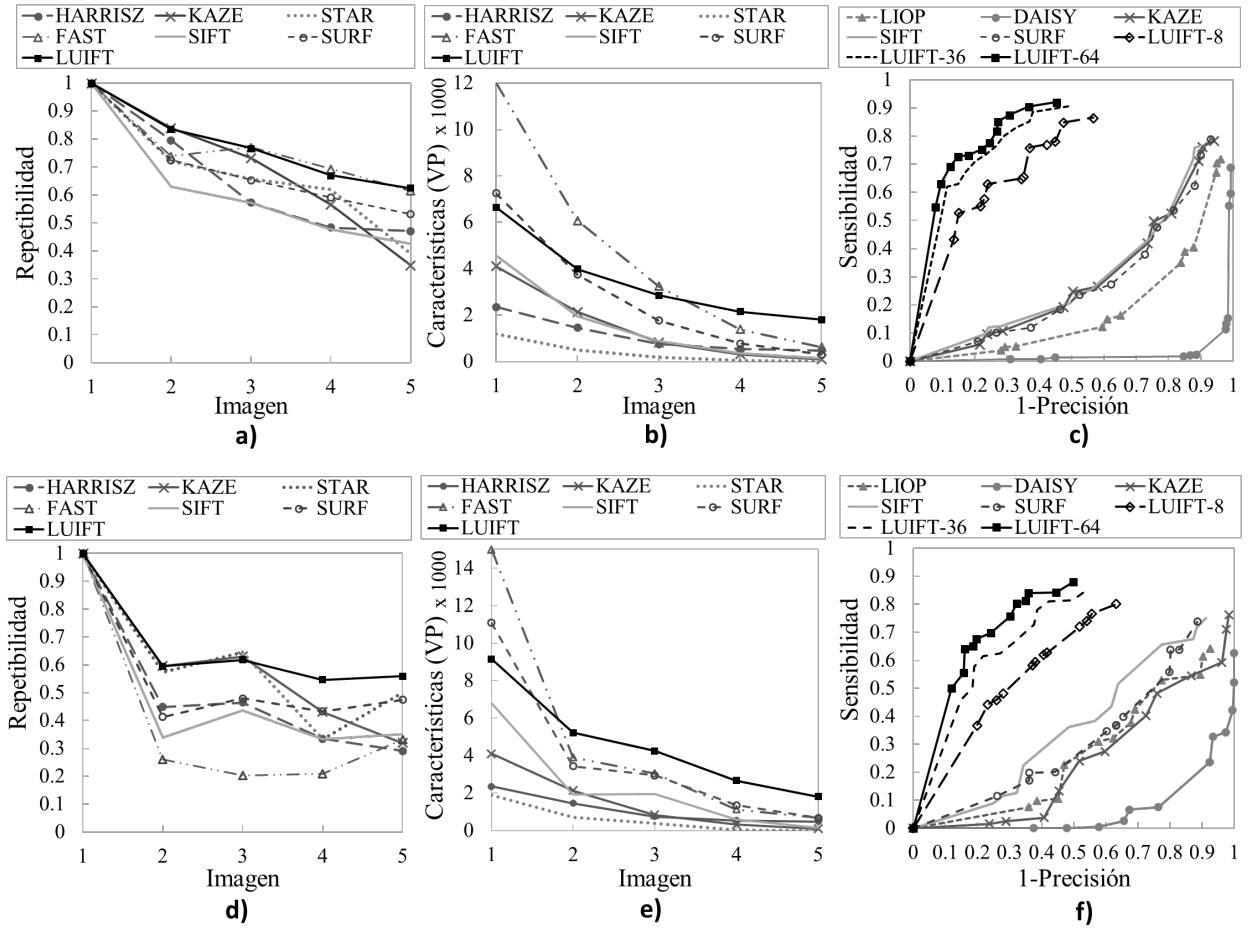


Figura 33. Resultados del conjunto de imágenes OFFICE. Para el conjunto de escenas del pasillo: (a) repetibilidad, (b) puntos característica correctamente detectados, y (c) curva de sensibilidad vs 1-precisión. Para el conjunto de escenas de escritorio: (d) detector de características repetibilidad; (d) repetibilidad, (e) puntos característica correctamente detectados, y (f) curva de sensibilidad vs 1-precisión.

desempeño de los métodos evaluados en términos de repetibilidad para el detector de características, y la curva de sensibilidad vs 1-precisión para el descriptor de características. Se puede observar que el descriptor propuesto obtiene un desempeño superior al de los métodos evaluados del estado del arte. A pesar de que el desempeño del detector de características FAST se asemeja al del detector propuesto LUIFT para la escena del pasillo en términos de repetibilidad (Figura 33(a)), el número de

puntos característica correctamente detectados en todas las imágenes por el detector propuesto es mayor que el del detector FAST (Figura 33(b)). Además, el número de características detectadas en la imagen original, utilizando el detector FAST, disminuye en más de un 50 % a medida que la escenas de pasillo se degradan (Figura 33(b)), y casi un 75 % para la escena de escritorio (Figura 33(e)). El principal inconveniente del detector FAST es que el número deseado de características detectadas por el método necesita ser ajustado para cada tipo de escena. Hay que tener en cuenta que es importante que los métodos de detección no sólo tengan una alta repetibilidad, sino que también obtengan un alto número de puntos característica correctos.

También se evaluó y comparó el detector y descriptor propuesto LUIFT en el conjunto de imágenes PHOS. La Figura 34 muestra el desempeño del descriptor LUIFT y los métodos del estado del arte en el conjunto de imágenes PHOS en términos de repetibilidad de texto y la curva sensibilidad vs 1-precisión.

Para el caso de las variaciones de exposición, la Figura 34(a) muestra el desempeño promedio del detector de características en términos de repetibilidad, mientras que la Figura 34(b) muestra el desempeño promedio del descriptor de características en términos de la curva de sensibilidad vs 1-precisión. Para el caso de variaciones de iluminación no uniforme, la Figura 34(c) muestra el desempeño promedio del detector de características en términos de repetibilidad, mientras que la Figura 34(d) muestra el desempeño promedio del descriptor de características en términos de la curva de sensibilidad vs 1-precisión. El desempeño del detector y del descriptor LUIFT propuesto es superior al de todos los métodos evaluados.

El desempeño de los métodos evaluados para cada conjunto de escenas (incluyendo la exposición y la iluminación no uniforme) se muestra en la Figura 35 en términos de la curva de sensibilidad vs 1-precisión. El método propuesto supera en todos los casos a los demás descriptores.

En este Capítulo se presentó un descriptor robusto basado en la información de fase local para el reconocimiento de patrones en imágenes degradadas utilizando el enfoque de congruencia de fase y espacio-escala de la señal monogénica. El método propuesto muestra un desempeño superior bajo variaciones de iluminación y degra-

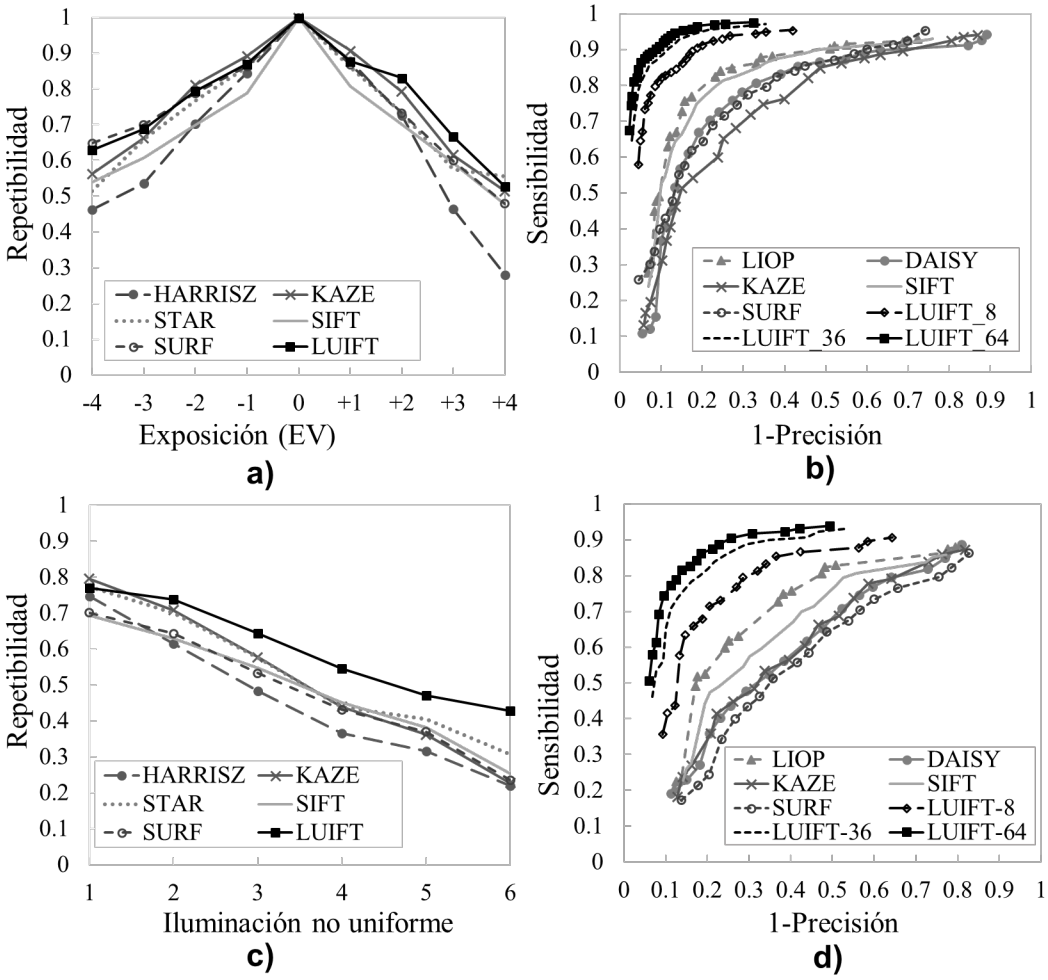


Figura 34. Desempeño de los métodos evaluados en el conjunto de datos PHOS en términos de repetibilidad y la curva de sensibilidad vs 1-precisión. (a) y (b) Resultados de la variación de la exposición; (c) y (d) resultados de la variación de la iluminación no uniforme.

daciones de ruido. Además, los resultados obtenidos en conjuntos de imágenes reales son competitivos con los obtenidos con descriptores del estado del arte. El desempeño del método propuesto puede mejorarse aún más incluyendo en el diseño la descomposición a escala piramidal y, dado que el método propuesto es intrínsecamente local, su implementación en GPU es relativamente sencilla, reduciendo su tiempo de ejecución.

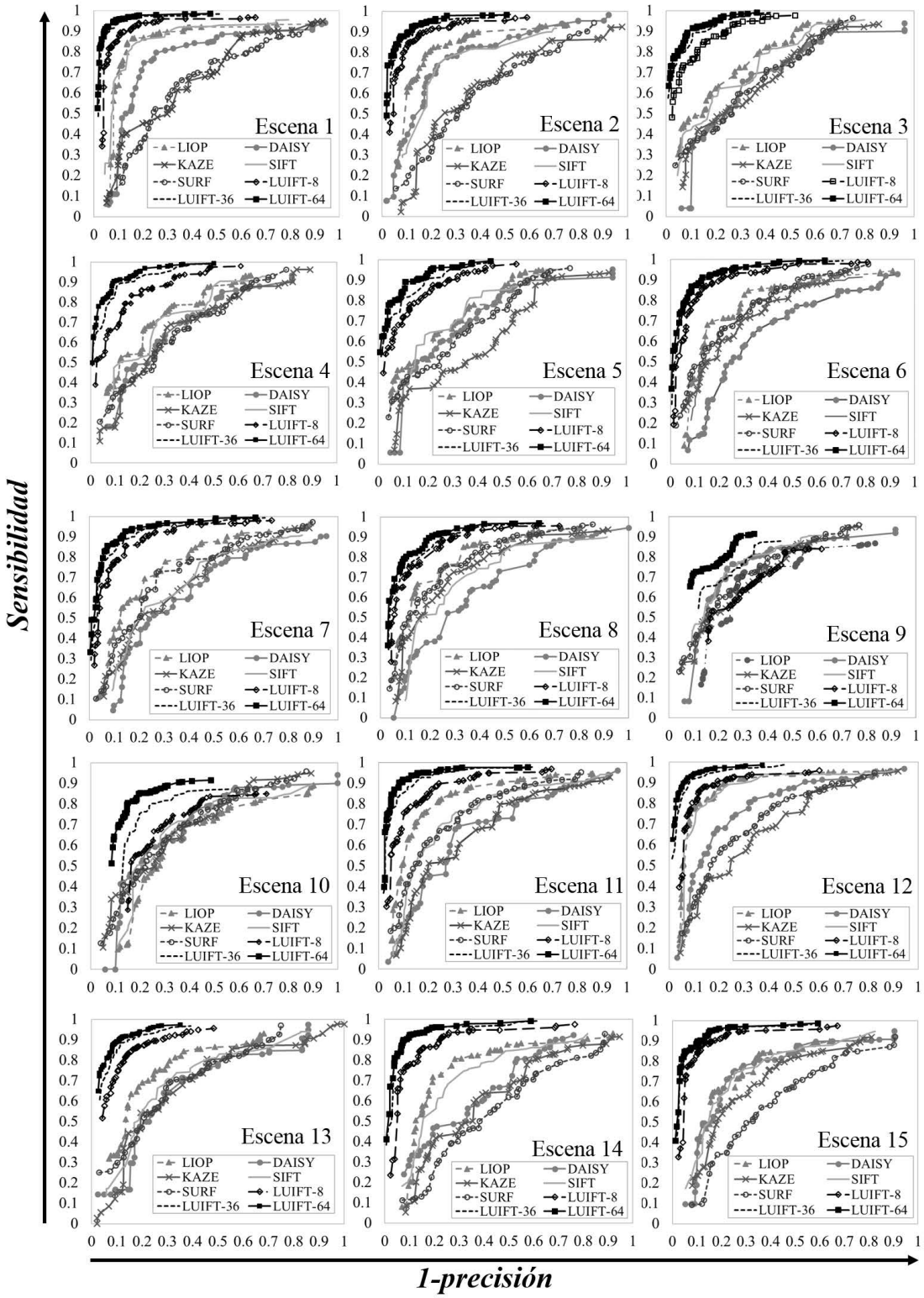


Figura 35. Desempeño de los métodos evaluados en el conjunto de imágenes PHOS en términos de la curva de sensibilidad vs 1-precisión.

Capítulo 5. Detección de Texto

En este Capítulo se describe el método propuesto para la detección y segmentación de caracteres en imágenes naturales. Básicamente, la imagen es segmentada en regiones llamadas *regiones candidatas*, las cuales son procesadas individualmente en busca de componentes de texto (caracteres). Las regiones se generan a partir de la información de fase local de la imagen, la cual se obtiene utilizando el espacio-escala de la señal monogénica (3.3.3). Para generar las regiones candidatas basadas en fase se utilizan dos enfoques: (1) máscaras binarias de fase local y (2) MSER de fase local. A continuación se describe cada una de ellas.

5.1. Máscaras binarias de fase local

Como se mencionó en el Capítulo 3, la fase local describe la información estructural de una imagen y nos permite distinguir sus diferentes características (borde o línea). Valores de fase cercanos a cero y π indican características de tipo borde (ascendente o descendente), mientras que valores de fase cercanos a $\pi/2$ y $3\pi/2$ indican características de tipo línea (oscura o brillante) (Kovesi *et al.*, 2002). Sin embargo, en este trabajo no estamos interesados en hacer una distinción entre líneas oscuras o brillantes, sino en encontrar características de tipo borde ascendentes y descendentes para la generación de regiones de componentes conectados. Por esta razón, consideramos el rango de 0 a π, mapeando los ángulos mayores que π de vuelta al rango.

Para generar las regiones, una primera propuesta fue definir dos máscaras binarias considerando la fase local de la imagen de la siguiente manera (Diaz-Escobar y Kober, 2017):

$$\mathbf{N}_{up} = \begin{cases} 1 & \text{si } \ 0 \leq \varphi(x,y) < \pi/3 \\ 0 & \text{si } \textit{otro caso,} \end{cases} \tag{60}$$

$$\mathbf{N}_{dwn} = \begin{cases} 1 & \text{si } \ 2\pi/3 < \varphi(x,y) \leq \pi \\ 0 & \textit{otro caso.} \end{cases} \tag{61}$$

donde $\mathbf{N}_{up}$ es la máscara binaria resultante de considerar bordes de tipo ascendente y $\mathbf{N}_{dwn}$ es la máscara binaria resultante de considerar bordes de tipo descendente (ver Figura 36).

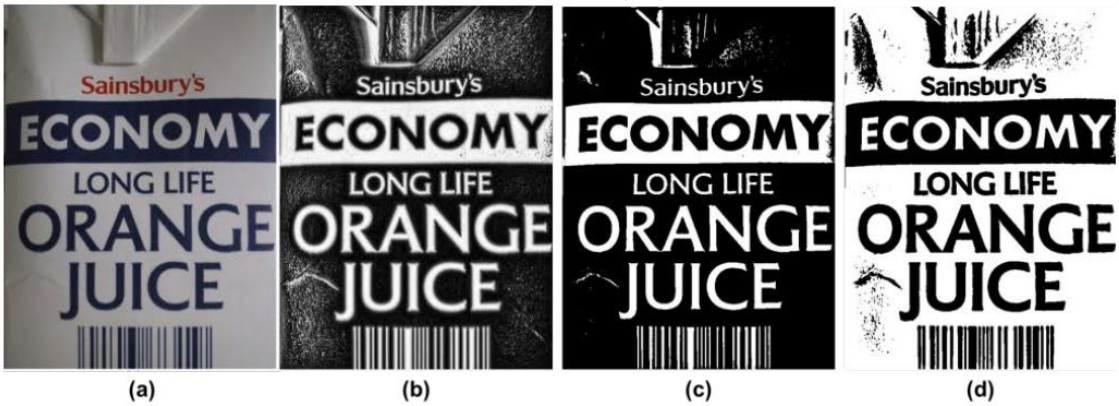


Figura 36. Ejemplo Máscaras binarias: (a) Imagen original, (b) Fase local, (c) $\mathbf{N}_{dwn}$, y (d) $\mathbf{N}_{up}$.

Cabe destacar que la información de fase local es invariable a escala y rotación. Además, la información de fase local es independiente a los cambios locales de intensidad, por lo tanto, es robusta a las variaciones de brillo, contraste, iluminación no uniforme y sombras.

Una vez que se obtienen las máscaras binarias $\mathbf{N}_{up}$ y $\mathbf{N}_{dwn}$, se extraen los Componentes Conectados (CC) de cada máscara y se procesan individualmente. Primero, a cada componente se le aplica la operación morfológica de cierre[1] para eliminar pequeños orificios y, posteriormente, se calculan las propiedades definidas en la Tabla 3. Una vez que se calcularon las propiedades de cada uno de los componentes, se lleva a cabo una etapa de filtrado con el objetivo de eliminar componentes que no pertenecen al texto, llamados componentes de no-texto.

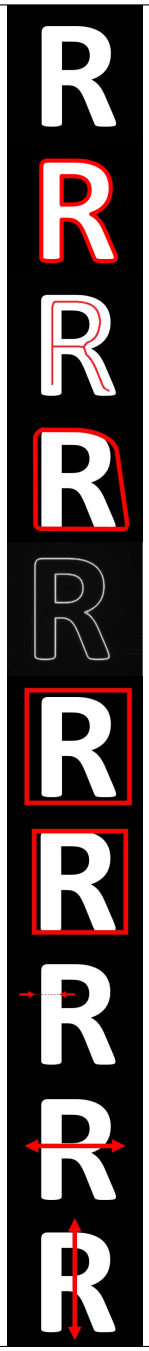
5.1.1. Filtrado de componentes conectados

Utilizando las propiedades de los componentes conectados (ver Tabla 3), se obtienen las siguientes características para el filtrado de los componentes:

- Área del rectángulo delimitador ($area(CC_{rect})$): el área del rectángulo delimitador de un componente.
- Área del componente conectado (CC_{area}): el número de píxeles conectados de un componente.

[1] El tamaño del elemento estructural fue definido experimentalmente como $\sqrt{\sqrt{CC_{area}}} \times \sqrt{\sqrt{CC_{area}}}$

Tabla 3. Propiedades Componentes Conectados *CC*

Nombre	Propiedad	Ejemplo
CC	Componente conectado	
CC_{con}	Pixeles del contorno	
CC_{skel}	Esqueleto morfológico	
CC_{envCon}	Envolvente convexa	
CC_{pc}	Congruencia de fase	
CC_{rect}	Rectángulo delimitador	
$CC_{minRect}$	Mínimo rectángulo delimitador	
CC_{SWV}	Ancho de trazo promedio	
CC_{ejeMin}	Eje mínimo	
CC_{ejeMax}	Eje máximo	

- Razón de aspecto ($CC_{aspecto}$) : la relación entre la altura y el ancho del mínimo rectángulo delimitador de un componente.
- Solidez (CC_s): relación entre la superficie de la región y la superficie del envolvente convexo ($area(CC)/area(CC_{envCon})$).
- Número de contornos (CC_{numCon}): el número de contornos cerrados de un componente.
- Puntos finales (CC_{pf}): número de puntos finales del esqueleto morfológico de un componente.
- Valor de ancho de trazo (CC_{SWV}): el valor de ancho de trazo de un componente se define como (Li *et al.*, 2014):

$$CC_{SWV} = Var(CC_{trazo})/E(CC_{trazo})^2,$$

donde $E(CC_{trazo})$ y $Var(CC_{trazo})$ son la media y la varianza del ancho de trazo (CC_{trazo}) del componente, respectivamente.

La Tabla 4 muestra los valores utilizados para el filtrado de los componentes en nuestro método (I_{area} denota el área de la imagen).

Tabla 4. Parámetros utilizados para el filtrado de los componentes.

Parámetros	Valores
$Max\ area(CC_{rect})$	30 % del I_{area}
$Min\ area(CC_{rect})$	0.3 % del I_{area}
$Max\ CC_{area}$	70 % del $Max\ area(CC_{rect})$
$Min\ CC_{area}$	30 % del $Min\ area(CC_{rect})$
$Max\ CC_S$	90 % del $Max\ area(CC_{rect})$
$Min\ CC_S$	20 % del $Min\ area(CC_{rect})$
$Min\ CC_{aspecto}$	0.10
$Max\ CC_{numCon}$	8
$Max\ CC_{pf}$	10
$Max\ CC_{SWV}$	0.30

Las áreas de los rectángulos, CC_{rect}, mínimo ($Min\ area(CC_{rect})$) y máximo ($Max\ area(CC_{rect})$) pueden ser modificados dependiendo del rango de tamaño de fuente que se requiera reconocer. Las áreas máxima y mínima de los componentes ($Max\ CC_{area}$ y $Min\ CC_{area}$) están relacionados con el área total de la imagen. La solidez del componente CC_s está

relacionada con el área de la envolvente convexa del componente. Generalmente, el número de contornos cerrados en componentes de texto (CC_{numCon}) es inferior a 4 (en el alfabeto inglés), pero este número puede variar según el alfabeto. Los componentes de texto también tienen un número limitado de puntos finales (CC_{pf}), pero pueden variar dependiendo del estilo de la fuente. Estos valores fueron obtenidos experimentalmente.

Finalmente, los componentes conectados filtrados son considerados como componentes candidatos a caracter y se filtran por su valor de congruencia de fase. Como se mencionó anteriormente (Sección 3.2), el valor de congruencia de fase local indica la importancia de la característica, uno significa el componente más significativo y cero el componente menos significativo. La congruencia de fase promedio de un componente conectado, CC_{pc}, se define utilizando el contorno del componente (CC_{con}) y los valores de PC (Ecuación 56) del contorno, de la siguiente manera:

$$CC_{pc} = \frac{1}{|CC_{con}|} \sum_{i=1}^{|CC_{con}|} PC(pt_i), \tag{62}$$

donde $|\cdot|$ denota cardinalidad y $pt_i = (x, y)$ representa al conjunto de puntos del contorno del componente ($\{pt_i \in CC_{con}\}$).

5.1.2. Resultados experimentales

Con el fin de analizar la tolerancia del método propuesto a las degradaciones de ruido e iluminación, se realizaron simulaciones por computadora utilizando imágenes sintéticas (ver Sección 2.1). Se seleccionó una imagen representativa que presenta diferentes fuentes de caracteres, tamaños y estructuras de fondo del conjunto de datos OSTD (Sección 2.2.3). La imagen fue rotada al azar entre 0 y 2π (50 imágenes). Para evaluar el desempeño del método en imágenes degradadas por ruido e iluminación no uniforme, cada imagen fue corrompida por ruido gaussiano aditivo con una desviación estándar de $\sigma = 10,\ 15\ y\ 20$; y degradada por iluminación no uniforme con $\phi = 85$, $\varphi = 60$ y $\rho = 5, 10, 20\ y\ 50$. Los resultados que se presentan a continuación son el resultado promedio de las evaluaciones realizadas.

La evaluación del desempeño del método propuesto se realizó utilizando las métricas (ver Sección 2.3) de sensibilidad (Ecuación 8), precisión (Ecuación 9) y la medida

Tabla 5. Desempeño del método propuesto bajo degradaciones de ruido aditivo.

σ	Precisión	Sensibilidad	F
0	90.3	96.6	93.3
10	95.5	96.8	96.1
15	96.3	92.7	94.4
20	97.8	78.2	86.8

Tabla 6. Desempeño del método propuesto bajo degradaciones de iluminación no uniforme.

ρ	precisión	Sensibilidad	F
5	87.6	97.0	92.1
10	88.7	96.9	92.6
20	89.7	96.9	93.2
50	89.9	97.1	93.4

F (Ecuación 10).

Las Tablas 5 y 6 muestran los resultados del desempeño del método propuesto bajo degradaciones de ruido aditivo e iluminación no uniforme, respectivamente, en términos de sensibilidad, precisión y la medida F.

Los resultados de los experimentos muestran que, en la mayoría de los casos, se detectó más del 90 % del texto. Los cambios en la iluminación no afectan al desempeño del método, mientras que la degradación por ruido aditivo afecta el desempeño del método debido a que destruye los caracteres pequeños o los fusiona con otras estructuras de fondo.

Finalmente, el método propuesto fue evaluado en el conjunto de imágenes OSTD completo (ver Sección 2.2.3). La Tabla 7 muestra los resultados del desempeño del método propuesto en términos de sensibilidad, precisión y la medida F.

Tabla 7. Desempeño del método propuesto en el conjunto de imágenes OSTD.

Precisión	Sensibilidad	*F*
61.4	85.5	69.2

Se puede observar que se detectó más del 80 % de los caracteres en las imáge-

nes del conjunto (alta sensibilidad), pero que se obtuvo un elevado número de falsos positivos (baja precisión), es decir, se consideraron caracteres a componentes que no lo son. Sin embargo, los falsos positivos pueden ser descartados en una etapa de reconocimiento de caracteres. La Figura 37 muestra algunos ejemplos de resultados del método propuesto.

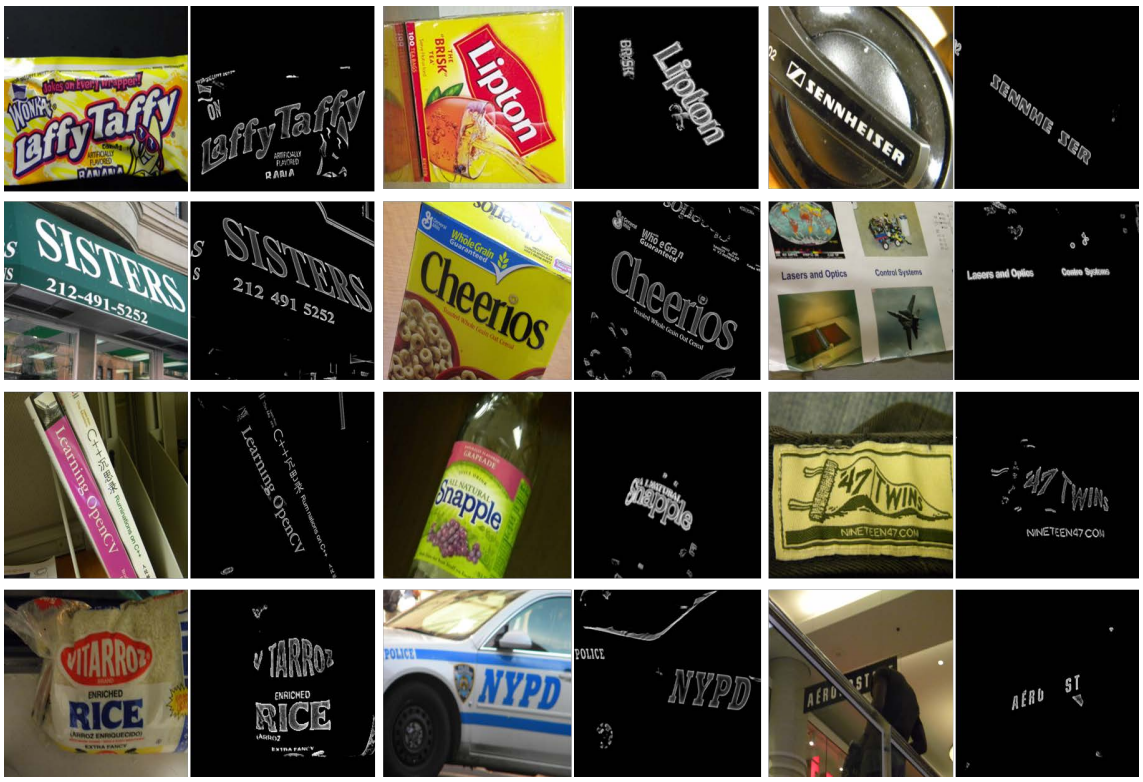


Figura 37. Ejemplo de los resultados obtenidos con el método propuesto de segmentación.

Después de analizar los resultados de los experimentos realizados en esta primera propuesta, consideramos dos mejoras para el método de detección de texto (Diaz-Escobar y Kober, 2018a,c). La primera consiste en utilizar la técnica de MSER para evitar definir sólo dos umbrales de fase local, tal como se realizó anteriormente con las máscaras binarias. La segunda mejora consiste en utilizar un clasificador para evitar el uso de reglas heurísticas en la clasificación de las regiones candidatas obtenidas. A continuación se describen brevemente la técnica MSER así como el clasificador AdaBoost que se utilizará para la clasificación de los componentes. Posteriormente se describe el segundo enfoque propuesto para la detección de texto.

5.2. Regiones Extremas Máximamente Estables

La técnica de Regiones Extremas Máximamente Estables (MSER, Maximaly Stable Extremal Regions) se introdujo por primera vez para las imágenes en escala de grises (Matas *et al.*, 2004). Básicamente, el método MSER extrae las zonas de la imagen que permanecen estables bajo un cierto número de umbrales. Formalmente se define a continuación.

Sea la imagen I un mapeo $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$. Las regiones extremas están bien definidas en la imagen si se cumplen las siguientes dos condiciones:

1. $\mathcal{S}$ es un conjunto totalmente ordenado (es reflexivo, anti-simétrico y transitivo bajo una relación binaria $\leq$).
2. Está definida una relación de adyacencia $A \subset \mathcal{D} \times \mathcal{D}$ (por ejemplo, 4-vecindad).

Se define como *Región* $\mathcal{Q}$ a un subconjunto continuo de $\mathcal{D}$, es decir, para cada $p, q \in \mathcal{Q}$ existe una secuencia $p, a_1, a_2, \ldots, a_n, q$ y $pAa_1, a_iAa_{i+1}, a_nAq$. Se define como Región frontera $\partial\mathcal{Q} = \{q \in \mathcal{D} \setminus \mathcal{Q} : \exists q \in \mathcal{Q} : qAp\}$, es decir, la frontera $\partial\mathcal{Q}$ de $\mathcal{Q}$ es el conjunto de pixeles adyacentes a al menos un pixel de $\mathcal{Q}$ pero que no pertenece a $\mathcal{Q}$. Una Región extrema $\mathcal{Q} \subset \mathcal{D}$ es una región tal que para todo $p \in \mathcal{Q}, q \in \partial\mathcal{Q} : I(p) > I(q)$ o $I(p) < I(q)$.

Sea $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_{i-1}, \mathcal{Q}_i, \ldots$ una secuencia de regiones extremas anidadas, es decir, $\mathcal{Q}_i \subset \mathcal{Q}_{i+1}$. La región extrema $\mathcal{Q}_{i^*}$ es máximamente estable si sólo si $q(i) = \frac{|\mathcal{Q}_{i+\Delta} \setminus \mathcal{Q}_{i-\Delta}|}{|\mathcal{Q}_i|}$ tiene un mínimo local i^*, donde $|\cdot|$ denota cardinalidad, y $\Delta \in S$ es un parámetro que considera la estabilidad de la región bajo un cierto número de umbrales.

5.3. Clasificador AdaBoost

AdaBoost (AdaBoost, Adaptive Boosting) (Freund y Schapire, 1997) es un enfoque de aprendizaje de máquina basado en la idea de crear una regla de predicción precisa mediante la combinación de muchas reglas relativamente débiles e inexactas.

Formalmente, el “fortalecimiento” se realiza de la siguiente manera: El fortalecedor es proporcionado con un conjunto de ejemplos etiquetados $\{(x_1, y_1), \ldots, (x_N, y_N)\}$,

donde y_i es la etiqueta asociada a la instancia x_i. En cada ronda, $t = 1, ..., T$, el fortalecedor establece una distribución D_t sobre el conjunto de ejemplos, y solicita (al clasificador débil) una hipótesis débil (o regla general) h_t con un error bajo ϵ_t con respecto a D_t. Entonces, la distribución D_t especifica la importancia relativa de cada ejemplo para la ronda actual. Finalmente, después de T rondas, el fortalecedor debe combinar las hipótesis débiles en una sola regla de predicción.

Pseudocódigo 1: Clasificador AdaBoost

Entrada: (x_1, y_1),(x_2, y_2),...,(x_N, y_N), con $x_i \in \mathcal{X}$, $y_i \in \{-1, 1\}$.

begin

  Inicializar: $D_1(i) = 1/N$ *para* $i = 1, 2, ..., N$;

  for $i \leftarrow 1$ *to* T **do**

    Entrenar un clasificador débil utilizando la distribución D_t;

    Obtener la hipótesis débil $h_t : \mathcal{X} \rightarrow \{-1, 1\}$;

    Seleccionar h_t con el valor de error ponderado más bajo: $\epsilon_t = Pr_{i\ D_t}[h_t(x_i) \neq y_i]$;

    Escoger $\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;

    Actualizar para $i = 1, 2, ..., N$:

$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

    donde Z_t es un factor de normalización (se escoge de tal forma que D_{t+1} sea una distribución);

  end

  return Hipótesis final:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

end

5.4. MSER de fase local y clasificador AdaBoost

El principal inconveniente con el método de binarización de una imagen en escala de grises es la selección de un único umbral que segmente la imagen perfectamente. Por lo general, esto no sucede y es necesario seleccionar un umbral diferente para cada tipo de imagen. En la Figura 38 se muestra un ejemplo de la binarización de una imagen utilizando tres valores de umbral diferentes.

El método MSER se introdujo por primera vez para las imágenes en escala de grises, pero puede aplicarse a otros tipos de imágenes siempre y cuando se mantengan las condiciones de orden total y la existencia de una relación de adyacencia (ver Sección 5.2). En este trabajo se propone utilizar la técnica de MSER para extraer las regiones candidatas de la información de fase local de la imagen en lugar de utilizar la imagen en escala de grises (Diaz-Escobar y Kober, 2018a,b). A diferencia de la pro-

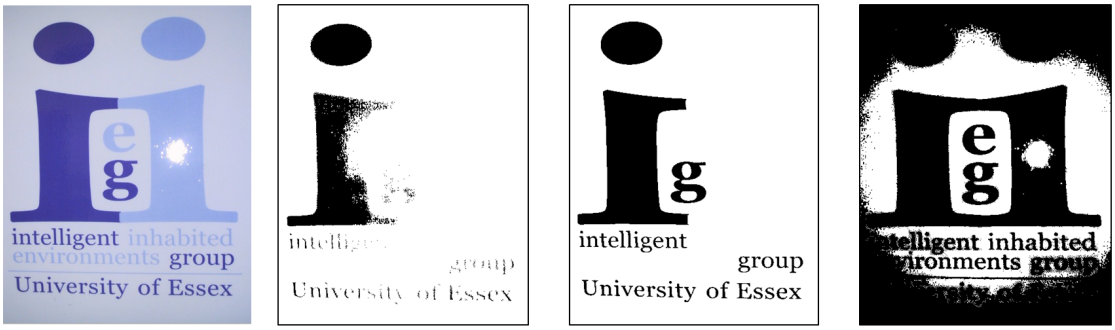


Figura 38. Ejemplo de la binarización de una imagen con los valores de umbral 50, 128, 200 (de izquierda a derecha).

puesta de máscaras binarias de fase local (Sección 5.1), en esta propuesta, se evita el uso de umbrales pre-definidos y, en su lugar, se obtienen todas las posibles regiones extremas basadas en la información de fase local de la escena. A continuación se describe el procedimiento.

5.4.1. MSER de fase local

Sea I una imagen en escala de grises y ϕ su fase local (Ecuación 47). La imagen binaria $I_{bin}^{(t)}$ se define como:

$$I_{bin}^{(t)}(x,y) = \begin{cases} 1, & \text{si } \phi(x,y) > t \\ 0, & \text{si } otro\ caso. \end{cases} \tag{63}$$

donde t denota un valor de umbral.

Una región extrema R_t en el umbral t se define como:

$$\forall\, p \in R_t,\ q \in \partial R_t \ \Rightarrow\ I_{bin}^{(t)}(p) > I_{bin}^{(t)}(q)\ \text{ o }\ I_{bin}^{(t)}(p) < I_{bin}^{(t)}(q). \tag{64}$$

La región R_{t^*} será máximamente estable si y sólo si:

$$q(t) = \frac{|R_{t+\Delta}| - |R_{t-\Delta}|}{|R_t|} \tag{65}$$

tiene un mínimo local en el umbral t^*, donde $|\cdot|$ denota cardinalidad, y $\Delta \in S$ es un parámetro que considera la estabilidad de la región bajo de un cierto número de umbrales.

Así, utilizando el enfoque MSER, los componentes conectados basados en fase se obtienen como regiones extremas máximamente estables a partir de la información de fase local, llamado Fase-MSER. La Figura 39 muestra una comparación de la técnica MSER utilizando la imagen en escala de grises y la técnica propuesta Fase-MSER.

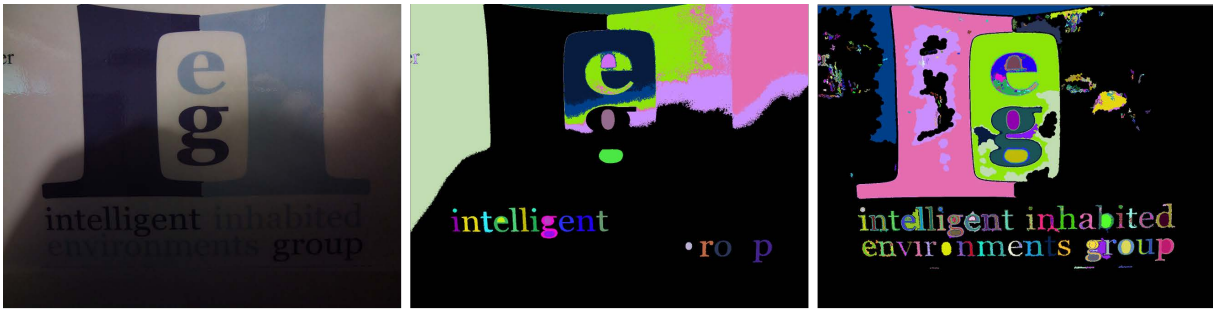


Figura 39. MSER vs Fase-MSER: (a) imagen original, (b) MSER y (c) Fase-MSER.

5.4.2. Filtrado de regiones

En esta etapa, para disminuir el número de regiones generadas a clasificar, se eliminan aquellas regiones que no cumplan con las siguientes condiciones:

1. El área del componente, para eliminar componentes de no-texto que sean muy pequeños o grandes: $max(50, 5x10^{-4} \cdot I_{area}) < CC_{area} < \frac{1}{2} \cdot I_{area}$, con I_{area} el área total de la imagen.
2. La razón de aspecto, para eliminar los componentes de no-texto que sean demasiado estrechos o anchos, $CC_{aspecto} < 0.10$.
3. El valor de congruencia de fase, para eliminar componentes de no-texto con valores de congruencia de fase baja. Si el valor PC promedio, CC_{pc} (Ecuación 62), es inferior a un umbral predefinido PC_{th} entonces, el componente es descartado. La Figura 40 muestra un ejemplo de los componentes bajo diferentes valores de umbrales PC_{th} ($0 \leq PC_{th} \leq 1$).

Cabe destacar que realizar este filtrado no es necesario, podría omitirse completamente. Sin embargo, se realiza con la intención de evitar procesar componentes que, de antemano, se sabe no son caracteres candidatos y, de este modo, evitar un costo computacional mayor.

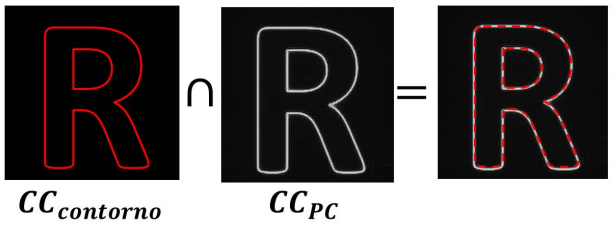
Figura 40. Filtrado de componentes candidatos bajo diferentes valores de umbrales PC_{th}.

5.4.3. Extracción de características y clasificación

Para clasificar las regiones generadas utilizando la técnica propuesta de Fase-MSER, se propone el siguiente conjunto de características:

1. Valor promedio de congruencia de fase (CC_{pc}): considera el valor promedio de congruencia de fase del componente. PC_m se calcula de la siguiente manera:

$$CC_pc = \frac{1}{|CC_{con}|} \sum_{i=1}^{|CC_{con}|} PC(pt_i)$$



con $\{pt_i \in CC_{con}\}$ y $|\cdot|$ denota cardinalidad.

2. Razón de congruencia de fase (CC_{pcr}): considera la contribución de los píxeles del contorno del componente. Uno significa una contribución completa de todos los píxeles del contorno y cero indica la contribución más baja. El PC_r se calcula como

$$PC_r = \frac{1}{|CC_{con}|} \sum_{i=1}^{|CC_{con}|} \mathbf{D}\left(PC(pt_i), PC_{th}\right) \ , \tag{66}$$

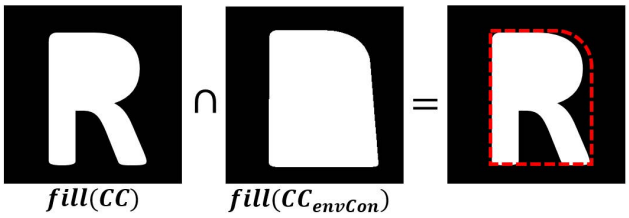
donde

$$\mathbf{D}\left(PC(pt_i), PC_{th}\right) = \begin{cases} 1 & \text{if } PC(pt_i) > PC_{th} \\ 0 & \text{otro caso.} \end{cases}$$

y PC_{th} es un umbral entre cero y uno.

3. Razón envolvente convexa: considera la convexidad del componente de la siguiente manera:

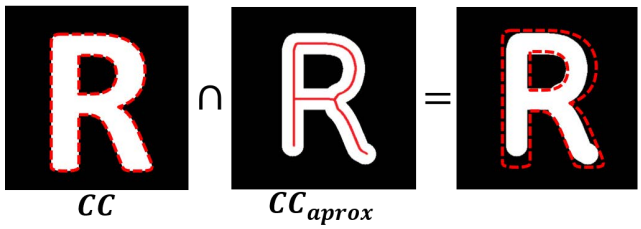
$$\frac{area(fill(CC))}{area(CC_{envCon})}$$



con $fill(CC)$ el componente sin espacios interiores vacíos.

4. Razón de área aproximada: considera la uniformidad del trazo del componente. Uno significa una uniformidad completa del trazo y cero indica una uniformidad baja. La razón de área aproximada, CC_{approx}, se calcula como:

$$\frac{abs(CC_{area} - CC_{approx})}{max(CC_{area}, CC_{aprox})}$$



donde $CC_{aprox} = CC_{trazo} \times longitud(CC_{skel})$.

5. Razón de contornos: considera la diferencia entre los contornos internos y el contorno externo del componente. Esto es para considerar la complejidad del contorno del componente. La razón de los contornos se calcula de la siguiente manera:

$$\frac{abs(long(CC_{con}) - long(CC_{conExt}))}{long(CC_{conExt})}$$

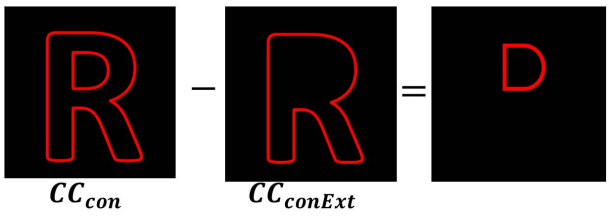


Tabla 8. Características de Componentes Conectados (CC)

Característica	**Definición**
Ocupación	$\frac{CC_{area}}{area(CC_{minRect})}$
Excentricidad	$\sqrt{1-\left(\frac{long(CC_{minAxis})}{longitud(CC_{maxAxis})}\right)^2}$
Solidez	$\frac{CC_{area}}{area(CC_{envCon})}$
Compacidad	$\frac{CC_{area}}{long(CC_{con})^2}$
Razón de área	$\frac{area(fill(CC))}{area(CC)}$
Razón de aspecto	$\frac{min(CC_{ancho},CC_{alto})}{max(CC_{ancho},CC_{alto})}$
Razón de contorno	$\frac{longitud(CC_{skel})}{longitud(CC_{con})}$
Valor medio de ancho de trazo	$\frac{var(CC_{trazo})}{E(CC_{trazo})^2}$
Razón de ancho de trazo mínimo	$\frac{CC_{trazo}}{min(CC_{ancho},CC_{alto})}$
Razón de ancho de trazo máximo	$\frac{CC_{trazo}}{max(CC_{ancho},CC_{alto})}$

donde $long(\cdot)$ y CC_{conext} representan la medida de longitud y el contorno externo del componente, respectivamente.

Además de utilizar las características propuestas y descritas anteriormente, también se consideraron las características descritas en la Tabla 8 (Neumann y Matas, 2012; Li *et al.*, 2014). Todas las características descritas anteriormente fueron utilizadas para entrenar un clasificador del tipo AdaBoost (Sección 5.3) y obtener los componentes de texto, llamados caracteres candidatos.

El clasificador $AdaBoost_c$ se entrenó utilizando el conjunto de entrenamiento de ICDAR2013 (Sección 2.2.4). Un componente es considerado como componente de texto si la suma de los votos del clasificador es positiva. Los candidatos restantes con la suma de votos negativa son considerados como Vecino Candidato, VC y son utilizados en la siguiente etapa de recuperación de caracteres.

5.4.4. Recuperación de caracteres

Durante la etapa de entrenamiento del clasificador, algunos caracteres se etiquetaron intencionalmente como componentes de no-texto (“I”, “i”, “L”, y “1”) para reducir los errores de clasificación ya que estos caracteres suelen ser similares a las estructuras de no-texto en las imágenes. La etapa de recuperación busca recuperar estos caracteres y otros que han sido clasificados erróneamente. El método de recuperación de caracteres se describe a continuación.

Para cada Caracter correctamente clasificado en la etapa anterior (Car), se define una vecindad de radio $r = 4 \cdot max(Car_{alto}, Car_{ancho})$. Todos los Vecinos Candidatos VCs dentro del radio r se consideran como vecinos candidatos de Car. Si un Car no tiene VCs posibles, entonces el caracter es descartado de la etapa de recuperación, pero continúa como un caracter más. Esto significa que los caracteres aislados no son descartados, como en muchos otros trabajos.

Posteriormente, se evalúa cada VC para determinar si se trata de un caracter clasificado erróneamente o efectivamente es un componente de no-texto. Para ello, se aplica un segundo clasificador $AdaBoost_R$. El clasificador $AdaBoost_R$ se entrenó utilizando características entre el Car y sus VCs. Estas características se describen a continuación:

1. La diferencia de área (D_{area}):

$$D_{area}(Car, VC) = \frac{abs(Car_{area} - VC_{area})}{max(Car_{area}, VC_{area})}. \tag{67}$$

2. La diferencia de mínimo rectángulo delimitador ($D_{minRect}$):

$$D_{minRect}(Car, VC) = \frac{abs(area(Car_{minRect}) - area(VC_{minRect}))}{max(area(Car_{minRect}), area(VC_{minRectt}))}. \tag{68}$$

3. La diferencia de valor de gris (D_{gris}):

$$D_{gris} = \frac{abs(Car_{gray} - VC_{gray})}{255}. \tag{69}$$

4. La diferencia de ancho de trazo promedio (D_{trazo}):

$$D_{trazo} = \frac{abs(Car_{trazo} - VC_{trazo})}{max(Car_{trazo}, VC_{trazo})}. \tag{70}$$

5. La razón de altura (D_h):

$$D_h = \frac{min(Car_{alto}, VC_{alto})}{max(Car_{alto}, VC_{alto})}. \tag{71}$$

6. La razón de anchura (D_w):

$$D_w = \frac{min(Car_{ancho}, VC_{ancho})}{max(Car_{ancho}, VC_{ancho})}. \tag{72}$$

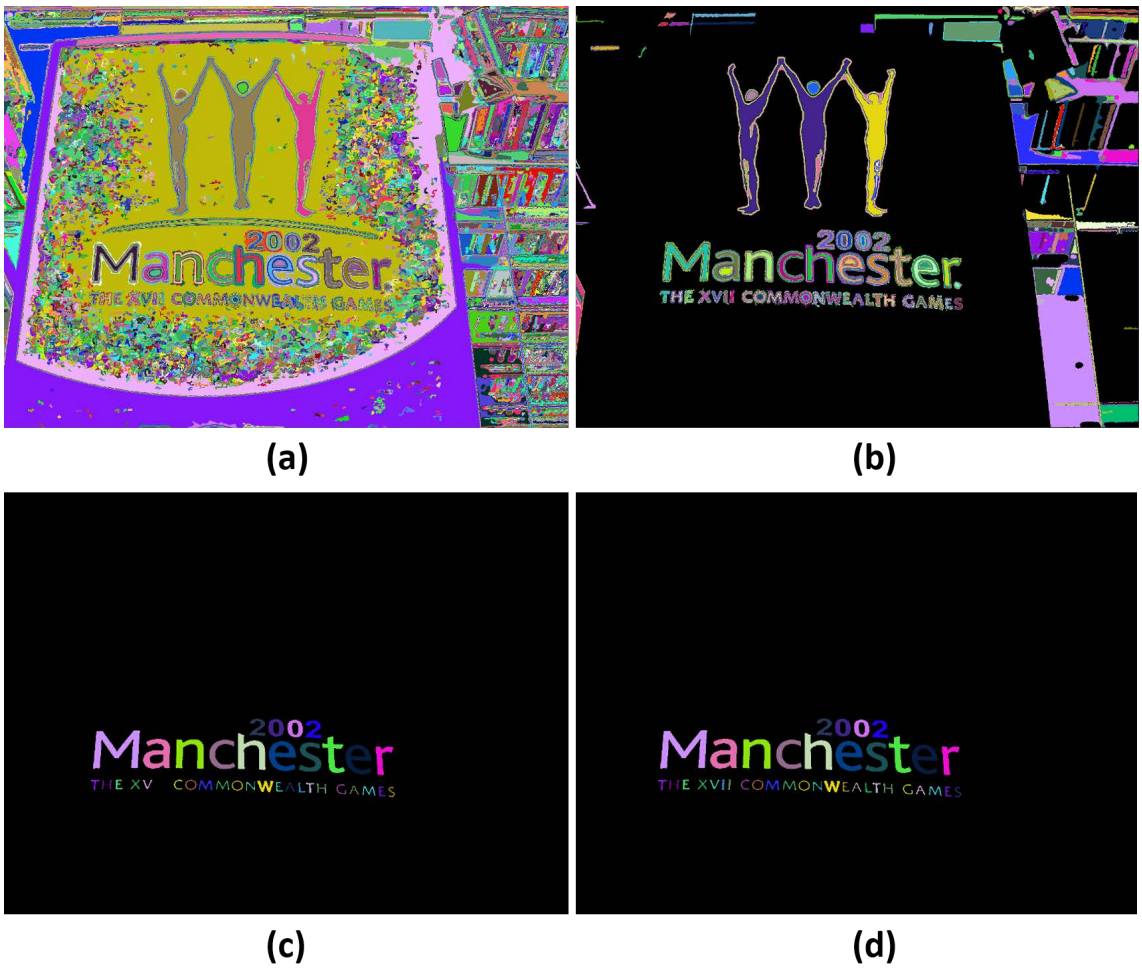


Figura 41. Ejemplo del método propuesto de segmentación de texto: (a) regiones Fase-MSER, (b) Filtrado de componentes, (c) Clasificación ($AdaBoost_C$) de regiones de texto y no-texto, (d) recuperación de caracteres ($AdaBoost_R$).

Una vez entrenado el clasificador, los vecinos candidatos son clasificados y recupe-

rados como caracteres siempre y cuando la suma de los votos del clasificador sea positiva. Además, estos nuevos caracteres recuperados a su vez se utilizan para la recuperación de sus vecinos candidatos de manera recursiva. El método se detiene cuando ningún nuevo vecino se clasifica como un nuevo caracter.

Tenga en cuenta que no se calcula ninguna característica de alineación, como en muchos trabajos relacionados. Considerar la alineación horizontal ayuda a evitar la clasificación errónea de los caracteres, pero restringe el método a texto horizontal. Por lo tanto, el método propuesto puede aplicarse a imágenes con texto multi-orientado. La Figura 41 muestra un ejemplo del método propuesto de segmentación de texto.

5.4.5. Agrupamiento de caracteres

Dado que la mayoría de los métodos de detección de texto del estado del arte evalúan la localización de las palabras en lugar de la segmentación de caracteres, se consideró una etapa de agrupamiento de caracteres en palabras. Básicamente, los caracteres similares más cercanos se agrupan y se consideran palabras candidatas. Posteriormente, se aplica la transformada Hough para obtener las líneas de la palabra final así como su orientación. El método de agrupación de caracteres se describe a continuación.

Primero, para cada caracter, se extraen todos sus Caracteres Vecinos $\{CV_i\}$ dentro de un radio $r = 4 \cdot max(Car_{alto}, Car_{ancho})$. Posteriormente, se define un valor de similitud ($sim(Car, CV_i)$) entre dos caracteres, el cual está dado por el promedio de los valores de las características definidas en la Sección 5.4.4: la diferencia de área (Ecuación 67), la diferencia de mínimo rectángulo delimitador (Ecuación 68), la diferencia de ancho de trazo promedio (Ecuación 70), la razón de altura (Ecuación 71) y la razón de anchura (Ecuación 72) de la siguiente manera:

$$sim(Car, CV_i) = \frac{1}{5} \cdot [D_{area}(Car, CV_i) + D_{minRect}(Car, CV_i) + D_{trazo}(Car, CV_i) + (1 - D_h(Car, CV_i)) + (1 - D_w(Car, CV_i))], \quad (73)$$

con $sim(Car, CV_i) \in [0, 1]$. Un valor de sim cercano a uno significa una similitud baja, mientras que un valor de similitud cercano a cero significa una similitud alta entre el caracter y el caracter vecino. Una vez que se tiene el valor de similitud, se calcula la

distancia Euclidiana mínima entre la envolvente convexa del caracter y la de sus vecinos. Finalmente esta distancia es ponderada por el valor de similitud y es seleccionada la distancia mínima obtenida, *minDist*, tal como sigue:

$$minDist = min(sim(Car, CV_I) \times Dist_E(Car, CV_i)) \tag{74}$$

El caracter y su vecino con menor distancia (*minDist*) son agrupados, formando cadenas de caracteres. Finalmente, se aplica la transformada de Hough para obtener las líneas de las palabras finales. La Figura 42 muestra un ejemplo de agrupación de caracteres.

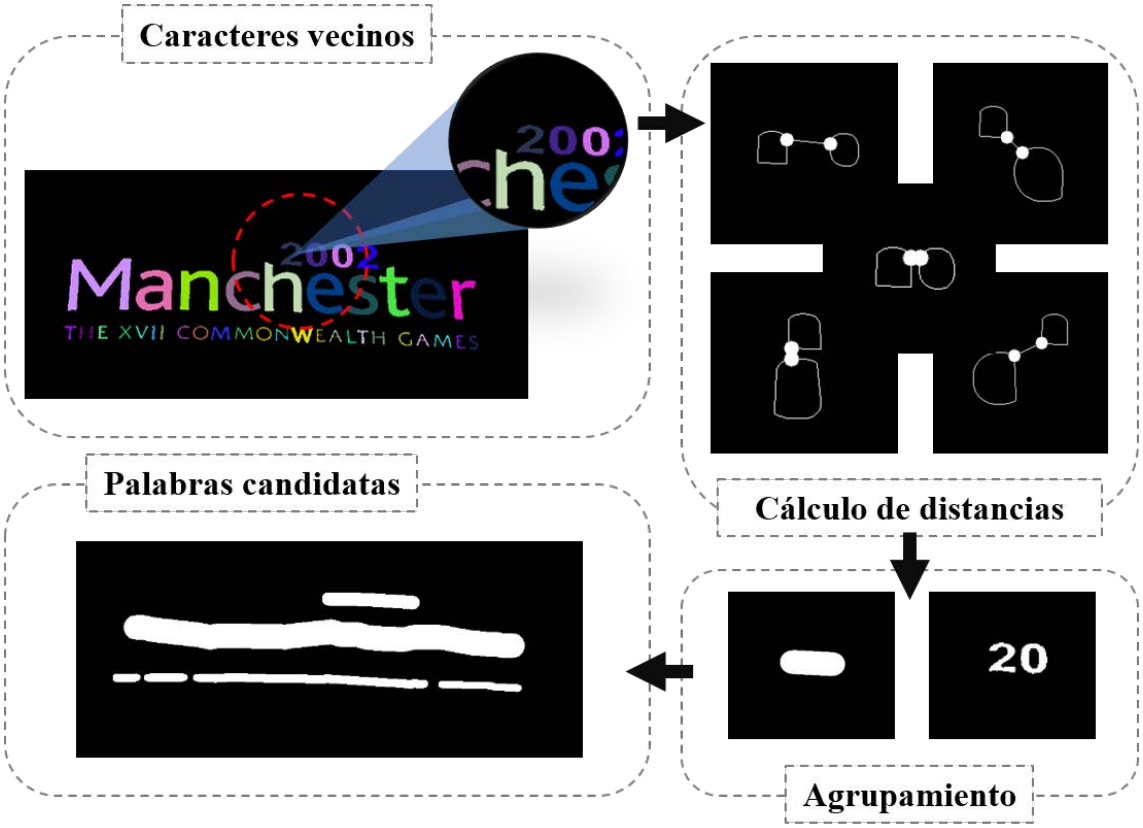


Figura 42. Ejemplo del método de agrupamiento propuesto.

5.5. Resultados experimentales

En esta Sección se analiza el desempeño del método propuesto en tres tareas diferentes: (1) la generación de regiones candidatas a partir del enfoque propuesto Fase-MSER (Sección 5.4.1); (2) la segmentación de los caracteres a nivel-pixel y a

nivel-átomo (Sección 2.3.6); y (3) la detección de texto a nivel palabra considerando el rectángulo delimitador (Sección 2.3.5).

5.5.1. Evaluación de la generación de regiones candidatas

Dado que la segmentación del texto depende de la calidad de la generación de las regiones candidatas, la primera evaluación realizada al método propuesto fue la generación de regiones basadas en fase, Fase-MSER (Sección 5.4.1). Para los experimentos, se seleccionaron diez imágenes representativas del conjunto de imágenes ICDAR2013 (Sección 2.2.4) y se generaron imágenes sintéticas con bajo contraste, alto brillo, sombras e iluminación no uniforme (Sección 2.1). Las imágenes seleccionadas contienen diferentes símbolos, tipos de fuente, colores, tamaños y fondos. Cada imagen fue escalada, rotada y degradada sintéticamente, obteniendo 1000 imágenes sintéticas por degradación (ver Figura 43). El método propuesto Fase-MSER es comparado con el método tradicional MSER utilizando los parámetros $\Delta = 4$, máxima variación $v = 0.5$ y mínima diversidad $d = 0.1$ (Saric, 2017).

La Tabla 9 muestra los resultados obtenidos en comparación con el método MSER en términos de la medida de sensibilidad-similitud (Sección 2.3.5).

El método propuesto muestra un buen desempeño en la generación de candidatos. La medida de sensibilidad-similitud fue mayor al 90 % en la mayoría de los casos, excepto en la degradación por brillo. Esto se debe a que las variaciones de brillo causaron la pérdida completa de regiones con bajo contraste (ver Figura 43, segunda fila, quinta columna). Además, el método Fase-MSER propuesto muestra un desempeño mayor al alcanzado por la técnica MSER tradicional, de hasta un 30 % para la iluminación no uniforme y sombras, y una mejora en el desempeño de hasta el 10 % para las variaciones de brillo y contraste.

Una vez evaluado el método utilizando imágenes sintéticas, se realizó una segunda evaluación utilizando el conjunto de imágenes ICDAR2013 completo y se comparó con métodos del estado del arte. La Tabla 10 muestra los resultados obtenidos en términos de la medida de sensibilidad-similitud y el promedio de regiones candidatas generadas. Como se puede observar en la Tabla 10, el método propuesto obtuvo un menor número de regiones candidatas con un porcentaje alto de sensibilidad-similitud

Figura 43. Ejemplo de imágenes sintéticas. De arriba hacia abajo: bajo contraste, alto brillo, sombras e iluminación no homogénea.

Tabla 9. Resultados a nivel-caracter en imágenes sintéticas (sensibilidad-similitud (%)).

Método	Contraste	Brillo	Iluminación	Sombras
MSER	76.1	74.0	65.0	66.2
Fase-MSER	**98.3**	**82.1**	**95.2**	**94.2**

comparado con los otros métodos. Esto es importante ya que existe un compromiso entre el número de regiones obtenidas y la complejidad computacional del sistema, ya que a mayor número de regiones generadas mayor es el tiempo de procesamiento. Además, nuestro método supera los resultados obtenidos en (Wu *et al.*, 2016) y (Sung *et al.*, 2015), incluso cuando los métodos utilizan distintos canales como escala de grises, RGB, Cb y Cr para la generación de regiones candidatas.

Por otro lado, a pesar de que el método en cascada que utiliza CNNs (Zheng *et al.*, 2017) reporta buenos resultados de sensibilidad-similitud para este mismo conjunto de imágenes, el promedio de candidatos generados por imagen es demasiado alto, casi 30 veces más que el método propuesto.

Tabla 10. Resultados de la generación de regiones candidatas en ICDAR2013.

Método	S-similitud(%)	Regiones candidatas
MSER (gris)	92.9	754
Zheng (Gray+H+S+Cb) (Zheng *et al.*, 2017)	98.6	6,651
Sung (gris+Cr+Cb) (Sung *et al.*, 2015)	87.7	401
Saric (gris)(Saric, 2017)	89.9	77
Wu *et al.* (RGB) (Wu *et al.*, 2016)	90.0	1,226
Fase-MSER (gris)	**91.0**	**220**

5.5.2. Evaluación de la segmentación a nivel-pixel y nivel-estructura

Para la evaluación de la segmentación del texto a nivel-pixel y nivel-estructura, se utilizaron las métricas de sensibilidad (R) y precisión (P), así como la medida F (F) (Sección 2.3.6). La Tabla 11 muestra los resultados del método propuesto en el conjunto de imágenes ICDAR13. Como podemos observar, el método propuesto supera los métodos de Yin *et al.* (2014) y Saric (2017), los cuales utilizan imágenes en escala de grises.

Tabla 11. Resultados de la segmentación de texto en el conjunto ICDAR13 (%).

	Nivel-pixel (%)			Nivel-átomo (%)		
Método	R	P	F	R	P	F
USTB_FuStar (Yin *et al.*, 2014)	69.5	74.4	71.9	68.0	72.4	70.1
Saric (Saric, 2017)	65.9	77.3	70.8	67.7	80.2	72.8
Propuesto	**69.9**	**85.2**	**76.1**	**68.0**	**80.1**	**73.5**

Ambos resultados, tanto la generación de regiones candidatas como la segmentación de texto, muestran que el método propuesto obtiene menos regiones candidatas con un mayor desempeño comparado con otros métodos del estado del arte.

Dado que la mayoría de los métodos existentes presentan la evaluación de la localización de textos en lugar de la segmentación de caracteres, realizamos la evaluación de localización de texto. La tabla 12 muestra los resultados obtenidos y el desempeño de métodos basados en MSER. El método propuesto muestra mejores resultados de medida F que la mayoría de los otros métodos, con excepción de (Sung *et al.*, 2015) y (Tian *et al.*, 2017), en los que se utilizan múltiples canales de la imagen. Sin embargo, el método propuesto por Sung *et al.* (2015) está diseñado sólo para texto horizontal, lo que le permite evitar la clasificación errónea de caracteres, pero lo limita a sólo texto horizontal. Por otro lado, el método propuesto por Tian *et al.* (2017) obtiene un

desempeño menor que nuestro método cuando utiliza solo imágenes en escala de grises. Además, este último método obtiene un rendimiento inferior en el conjunto de imágenes multi-orientado USTB en comparación con el método propuesto (véase la Tabla 13).

Tabla 12. Resultados de la localización de texto en el conjunto ICDAR2013 (%).

Método	R	P	F
Tian (gris) (Tian *et al.*, 2017)	67.8	81.2	73.9
Tian (RGB+V) (Tian *et al.*, 2017)	83.9	83.6	83.4
Saric (gris) (Saric, 2017)	67.7	80.2	72.8
Wu (RGB)(Wu *et al.*, 2016)	70.0	84.0	76.0
Neumann (RGB+I+H+S)(Neumann y Matas, 2016)	71.3	82.1	76.3
Yin (gris) (Yin *et al.*, 2015)	65.1	83.9	73.3
Sung (gris+Cr+Cb)(Sung *et al.*, 2015)	72.0	87.6	79.0
Yin (gris) (Yin *et al.*, 2014)	68.2	86.2	76.2
Propuesto (gris)	**73.9**	**82.7**	**78.0**

A continuación, con el fin de evaluar el desempeño del método propuesto en imágenes de texto multi-orientado, se utilizaron los conjuntos de imágenes multi-orientadas MSRA-TD500 (MSRA), USTB-SV1K (USTB) y OSTD (Sección 2.2.6). el método propuesto se evaluó utilizando el protocolo de evaluación utilizado en (Yao *et al.*, 2012) para una comparación justa. La tabla 13 muestra los resultados de la evaluación y los resultados del conjunto ICDAR2013 para una mejor visualización.

El método propuesto obtiene los mejores resultados en los conjuntos de imágenes USTB y OSTD en comparación con los otros métodos. Para el caso del conjunto MSRA, es importante destacar que el método propuesto fue entrenado considerando solamente caracteres latinos, por lo que su desempeño en este conjunto no es el mejor

Tabla 13. Resultados de detección de texto en ICDAR, MSRA, USTB y OSTD (%).

	ICDAR			MSRA			USTB			OSTD		
Método	R	P	F	R	P	F	R	P	F	R	P	F
Ma *et al.* (2018)	88.0	95.0	**91.0**	69.0	82.0	75.0	-	-	-	-	-	-
Wei *et al.* (2018)	81.1	87.3	84.3	-	-	-	55.9	54.1	55.0	76.2	75.4	75.8
He *et al.* (2017)	81.0	92.0	86.0	70.0	77.0	74.0	-	-	-	-	-	-
Tian *et al.* (2017)	83.9	83.6	83.8	-	-	-	48.7	53.8	51.1	-	-	-
Saric (2017)	66.1	76.5	70.6	-	-	-	31.8	44.6	37.1	45.4	49.8	47.5
Yin *et al.* (2015)	66.0	83.7	73.8	63.0	81.0	71.0	45.4	49.8	47.5	-	-	-
Li *et al.* (2014)	62.0	80.0	70.0	-	-	-	-	-	-	60.0	72.0	61.0
Yao *et al.* (2012)	66.0	69.0	67.0	63.0	63.0	60.0	-	-	-	73.0	77.0	74.0
Propuesto	73.9	82.7	78.0	63.9	74.3	65.6	58.8	68.8	**63.1**	89.0	90.1	**88.0**
Propuesto*	73.9	82.7	78.0	73.9	81.7	**75.7**	58.8	68.8	63.1	89.0	90.1	88.0

de todos. Sin embargo, muestra un desempeño mayor que la mayoría de los métodos presentes. Además, a pesar de que el método basado en el aprendizaje profundo propuesto por Ma *et al.* (2018) superó a todos los métodos en el conjunto MSRA, los mismos autores reportan que el desempeño de su método presenta una disminución de hasta un 25 % al ser entrenado usando sólo el conjunto de entrenamiento MSRA (300 imágenes), obteniendo una medida F más baja que nuestro método. Las Figuras 44, 45 y 46 muestran ejemplos del desempeño del método propuesto en los conjuntos de imágenes USTB-SV1K y MSRA-TD500.

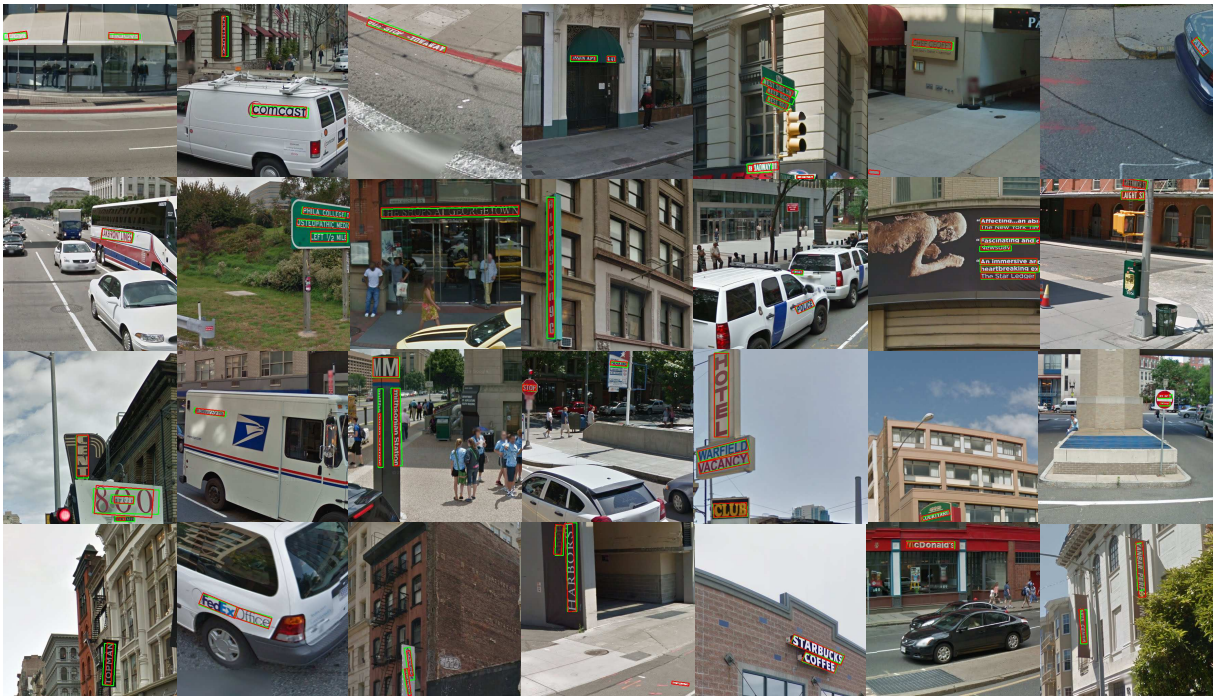


Figura 44. Resultados de detección de texto del método propuesto en el conjunto USTB-SV1K. Rectángulo verde: texto verdadero; rectángulo rojo: texto detectado por el método propuesto.

En la Figura 44, se muestran ejemplos de detección correcta de texto en imágenes del conjunto USTB-SV1K; mientras que la Figura 45 muestra algunos ejemplos de errores comunes que presentó el método propuesto en el conjunto USTB-SV1K. Estos errores pueden agruparse en tres tipos: (1) El error del logotipo de Google (primera fila), en el que el método propuesto reconoce la marca de agua de Google en las imágenes; (2) el error de texto sin marcar (segunda fila), en el que el método propuesto reconoce el texto pero no se considera texto según las etiquetas definidas en el conjunto de imágenes; y (3) los errores de falsos positivos y falsos negativos (tercera fila).

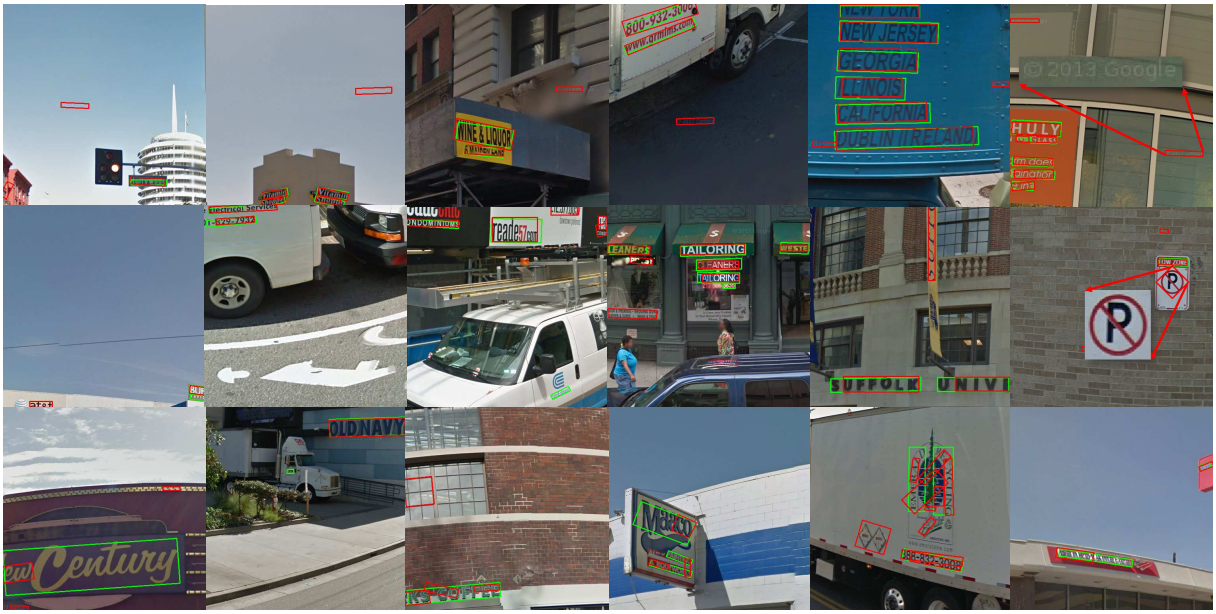


Figura 45. Errores de detección de texto del método propuesto en el conjunto USTB-SV1K. Rectángulo verde: texto verdadero; rectángulo rojo: texto detectado por el método propuesto.

Por otro lado, en la primera fila de la Figura 46, se muestran ejemplos de la detección correcta de texto en imágenes con texto sólo en idioma inglés; la segunda fila muestra ejemplos de detección correcta de texto en imágenes con texto en los idiomas inglés y chino; y la tercera fila muestra ejemplos de errores de detección de texto.

Figura 46. Resultados de detección de texto del método propuesto en el conjunto MSRA-TD500. Rectángulo verde: texto verdadero; rectángulo rojo: texto detectado por el método propuesto.

En este Capítulo se propuso un nuevo método de detección y segmentación de texto multi-orientado, inspirado en el sistema de visión humana. El método se basa en el modelo energía local y el marco teórico de espacio-escala de la señal monogénica. El método propuesto se divide en cuatro etapas: generación de regiones candidatas, clasificación de componentes de texto y no-texto, recuperación de caracteres y agrupación de caracteres en palabras. Las regiones candidatas basadas en fase se extraen aplicando el método propuesto de Fase-MSER, mientras que la recuperación y agrupación de caracteres se realiza aplicando clasificadores Adaboost para evitar el uso de reglas heurísticas. El método propuesto demostró ser robusto a las distorsiones geométricas, variaciones de fuentes, fondos complejos, bajo contraste, alto brillo, sombras y cambios de iluminación. El método logró un alto desempeño de segmentación de caracteres con un bajo número de regiones extraídas. El método supera los métodos del estado del arte en conjuntos de imágenes comunes en términos de segmentación de caracteres, localización de texto y número de regiones candidatas. Además, nuestro método no se limita a textos horizontales, como la mayoría de los métodos existentes, sino también a textos multi-orientado. Por último, el método propuesto puede utilizarse para la detección de texto en diferentes idiomas o textos escritos a mano.

Capítulo 6. Reconocimiento de Caracteres

En este Capítulo se presenta y describe el método propuesto para el reconocimiento de caracteres. Básicamente el método utiliza el descriptor LUIFT propuesto, descrito en el Capítulo 4 y el enfoque de bolsa de características para crear un descriptor representativo de cada caracter. Finalmente estos descriptores son clasificados utilizando una máquina de soporte vectorial. A continuación se describen brevemente el enfoque de bolsa de características y el clasificador de máquina de soporte vectorial utilizados.

6.1. Bolsa de Características

El modelo de “bolsa de características” (BOF, Bags of Features), también conocido como “bolsa de palabras” (BOW, Bag Of Words), fue introducido por primera vez para el análisis de documentos (Fei-Fei y Perona, 2005). Este modelo considera el número de repeticiones de palabras que aparecen en un documento en lugar de sus posiciones espaciales.

En la actualidad, el modelo BOF es ampliamente utilizado para tareas de reconocimiento de objetos. La idea general de este modelo consiste en representar un objeto como un histograma normalizado de recuentos de características. Básicamente, este método incluye tres pasos: extracción de características, generación visual de vocabulario (libro de códigos) y representación de características en histograma de frecuencias.

El vocabulario visual se construye agrupando las características extraídas de un conjunto de imágenes de entrenamiento, cada una de las cuales es una palabra visual. Posteriormente, dada una nueva imagen, se detectan las características y se asignan a los términos más cercanos que coincidan con el vocabulario visual. El descriptor final es entonces el histograma normalizado de las características cuantificadas detectadas en la imagen (O’Hara y Draper, 2011). La Figura 47 muestra un diagrama del enfoque BOF.

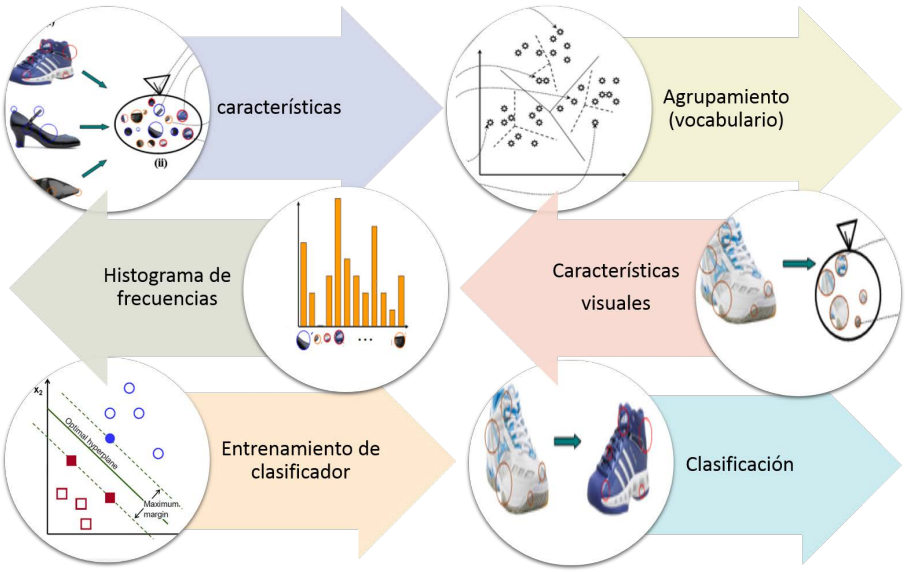


Figura 47. Diagrama del enfoque BOF.

6.2. Máquina de soporte vectorial

El clasificador de máquina de soporte vectorial (SVM, Support Vector Machine) es un algoritmo de aprendizaje de máquina supervisado propuesto por Cortes y Vapnik (1995), el cual puede ser utilizado tanto para problemas de regresión como de clasificación. Básicamente, el clasificador SVM mapea el espacio de características n-dimensional (donde la dimensión depende de el número de características que se consideran) a un espacio de dimensión mayor donde el objetivo es crear un hiper-plano o un conjunto de hiper-planos que dividan las clases de manera óptima.

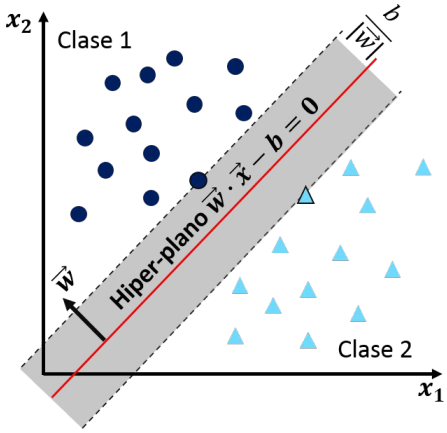


Figura 48. Máquina de soporte vectorial.

Formalmente, dado un conjunto de datos de entrenamiento

$$(\overrightarrow{r}_1, s_1), (\overrightarrow{r}_2, s_2), \ldots, (\overrightarrow{r}_3, s_3)$$

con $\overrightarrow{r}_i$ el vector n-dimensional de características y $s_i = \pm 1$ (dos clases) la clase a la que pertenece el vector $\overrightarrow{r}_i$. El clasificador SVM busca encontrar el hiper-plano que divida al grupo de puntos característica $\overrightarrow{r}_i$ etiquetados como $s_i = 1$ de los etiquetados como $s_i = -1$, y que, a su vez, maximice la distancia entre el hiper-plano y el punto más cercano $\overrightarrow{r}_i$ de cualquiera de los dos grupos:

$$\overrightarrow{w} \cdot \overrightarrow{r} - b = 0, \tag{75}$$

con $\overrightarrow{w}$ el vector normal (no necesariamente normalizado) hacia el hiperplano. El parámetro $\frac{b}{|\overrightarrow{w}|}$ determina el desplazamiento del hiperplano desde el origen a lo largo del vector normal (ver Figura 48).

6.3. Reconocimiento de caracteres

Debido a que el texto en imágenes naturales presenta diferentes tamaños de fuente y no necesariamente se encuentra alineado horizontalmente, es necesario aplicar un pre-procesamiento a los caracteres para eliminar transformaciones como escalamiento, rotación y cizallamiento.

6.3.1. Normalización de caracteres

Una vez que el texto es detectado y segmentado en palabras, la etapa de normalización se encarga de normalizar el tamaño y la geometría de los caracteres. A continuación se resumen los pasos para la etapa de normalización de caracteres.

1. Primero, se aplica una transformación de rotación $\mathbf{T}_{\boldsymbol{\theta}}$ (Sección 2.1.1) utilizando el ángulo θ de rotación del mínimo rectángulo delimitador de la palabra segmentada. Una vez rectificada la orientación de la palabra, se extraen los caracteres segmentados y se procesan por individual.

2. Para eliminar la distorsión geométrica de cizallamiento, se utiliza una aproximación de los valores $c_x = -mu_{11}/mu_{02}$ y $c_x = -mu_{11}/mu_{20}$ basados en los mo-

mentos centrales del caracter (Mukundan y Ramakrishnan, 1998). Aplicando la transformación de cizallamiento $\mathbf{T}_{\mathbf{c}_{x,y}}$ (Sección 2.1.1), obtenemos el caracter rectificado.

3. Finalmente se aplica una transformación de escalamiento $\mathbf{T}_{\mathbf{s}_{x,y}}$ a un tamaño de 64×64.

La Figura 49 muestra un ejemplo del proceso de normalización.

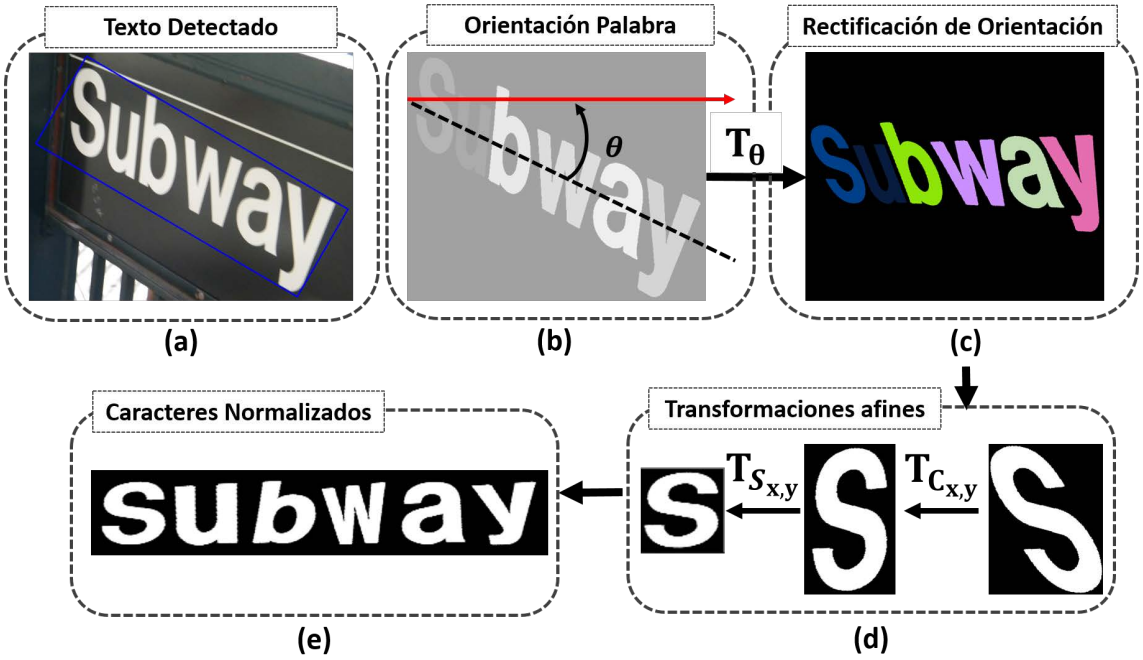


Figura 49. Ejemplo normalización de caracteres: (a) rectángulo delimitador de la palabra detectada; (b) orientación de la palabra con referencia del eje x; (c) palabra rectificada después de aplicar la trasformación $\mathbf{T}_{\boldsymbol{\theta}_{x,y}}$; (d) normalización de los caracteres aplicando las transformaciones $\mathbf{T}_{\mathbf{c}_{x,y}}$ y $\mathbf{T}_{\mathbf{s}_{x,y}}$; (e) caracteres normalizados.

6.3.2. Clasificación de caracteres

En esta sección se describe la metodología para el reconocimiento de caracteres. Se asume la segmentación del texto y sólo se consideran los caracteres segmentados. Básicamente, los puntos de interés se obtienen de la imagen del caracter como aquellos con mayor valor de congruencia de fase (Ecuación 56) del caracter. Después, para cada punto de interés, se calcula el descriptor LUIFT (Sección 4.2.2). Posteriormente, se utiliza el enfoque de BOF (Sección 6.1) y se crea un descriptor de “contornos”.

Finalmente, se utiliza una SVM (Sección 6.2) para la clasificación. A continuación se resumen los pasos para el reconocimiento de caracteres.

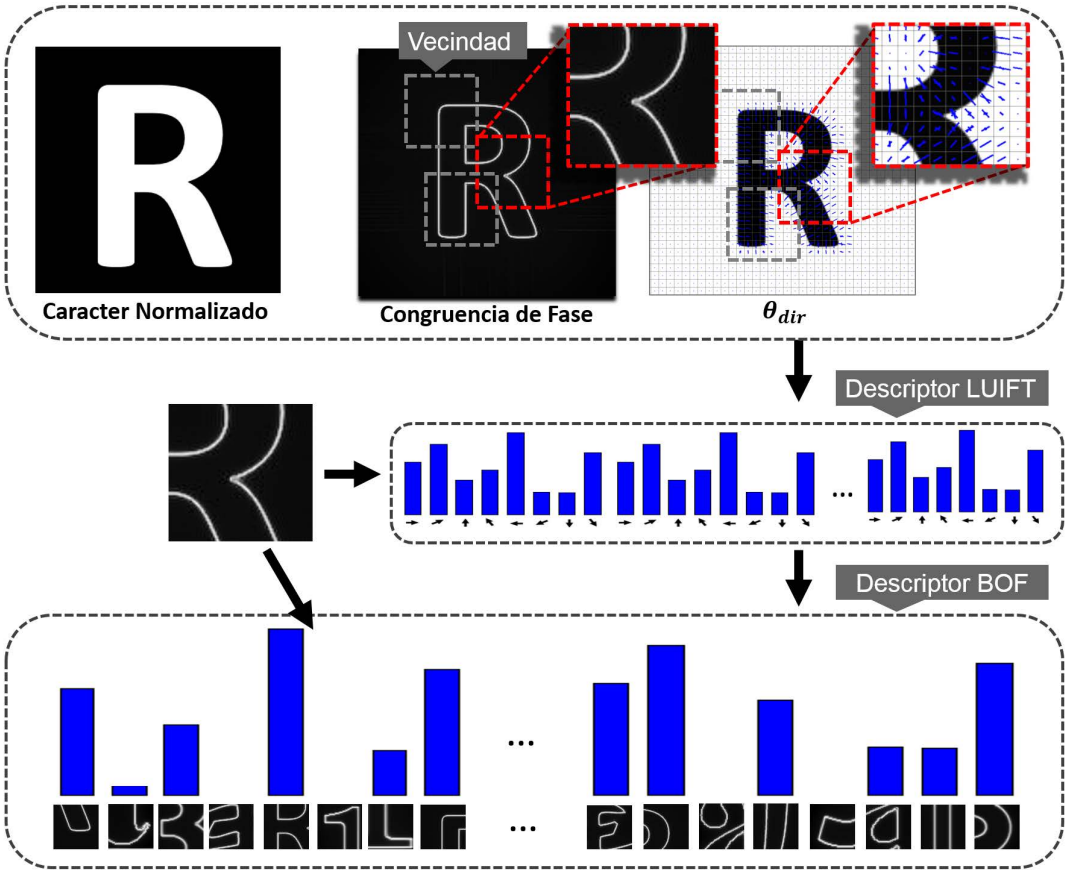


Figura 50. Ejemplo del descriptor propuesto para el reconocimiento de caracteres.

1. Primero, utilizando el espacio-escala de la señal monogénica de la imagen (Sección 3.3.3), se calcula la fase local (Ecuación 47), la orientación local (Ecuación 45), la dirección local (Ecuación 46) y la congruencia de fase local (Ecuación 56) de la imagen.

2. Utilizando la información de congruencia de fase y la orientación local, se aplica un algoritmo de supresión no máxima (González y Woods, 2006) para obtener los puntos de interés del contorno del caracter.

3. Una vez que obtenemos los puntos de interés, por cada punto, se calcula un descriptor LUIFT en una vecindad de tamaño 32×32.

4. Posteriormente, se utiliza el modelo BOF. El vocabulario visual se crea utilizando

un clasificador k-medias. Se considera un vocabulario de 300 palabras, con 62 y 49 clases. Finalmente se entrena un SVM con un núcleo de base radial (Muller *et al.*, 2001) para su clasificación.

La Figura 50 ilustra el procedimiento para el reconocimiento de caracteres.

6.4. Resultados experimentales

En esta Sección, se evalúa el método propuesto para el reconocimiento de caracteres y se compara con algunos métodos del estado del arte.

6.4.1. Evaluación en el conjunto Chars74k

El desempeño del sistema propuesto se compara con los resultados presentados por De Campos *et al.* (2009) en el conjunto de imágenes Chars74K (Sección 2.2.5) en términos del porcentaje de reconocimiento, el cual se define como el número de caracteres reconocidos correctamente entre el total de los caracteres. Para obtener una comparación justa, consideramos las mismas 15 imágenes de entrenamiento y de prueba que las reportadas en el trabajo de De Campos *et al.* (2009). Además, para mejorar los resultados de la clasificación, sólo se consideraron 49 clases, eliminando aquellas en las que los caracteres en minúsculas y mayúsculas tienen la misma forma, como por ejemplo: “p” y “P”, “z” y “Z”, “o” y “O”.

Tabla 14. Resultados del reconocimiento de caracteres en el conjunto Chars74K (%)

Descriptor	Fnt	Img
GB (Berg *et al.*, 2005)	69.71	47.09
SC (Belongie *et al.*, 2002)	64.83	34.41
SIFT (Lowe, 1999)	46.94	20.75
PCH (Varma y Zisserman, 2003)	44.93	21.40
SPIN (Lazebnik *et al.*, 2005)	28.75	11.83
MR8 (Varma y Zisserman, 2002)	30.71	10.43
ABBYY	66.05	30.77
MKL (De Campos *et al.*, 2009)	-	55.26
Propuesto (62 clases)	65.97	49.52
Propuesto (49 clases)	**74.19**	**58.14**

La Tabla 14 muestra los resultados obtenidos en ambos subconjuntos Fnt y Img del conjunto de imágenes Chars74K, así como los resultados obtenidos en De Campos

et al. (2009). Como se puede observar, el método propuesto obtuvo mejores resultados que la mayoría de los otros métodos (en el caso de 62 clases), mientras que obtuvo los mejores resultados considerando 49 clases. Cabe destacar que tanto el número de imágenes de entrenamiento como el de prueba fue muy bajo, tan sólo 15 imágenes por caracter.

Un segundo entrenamiento del clasificador fue realizado utilizando 1000 imágenes por caracter. Las imágenes de entrenamiento pertenecen al conjunto de entrenamiento de las imágenes sintéticas Fnt del conjunto Chars74K (Sección 2.2.5). Se evaluó el conjunto de imágenes Img del conjunto Chars74k pero esta vez considerando todas las imágenes del conjunto. La Tabla 15 muestra los resultados obtenidos y la comparación de resultados con el estado del arte.

Tabla 15. Resultados de la clasificación de caracteres conjunto Img-Chars74K (%)

Descriptor	49 clases	62 clases
Strokelets (Bai *et al.*, 2016)	-	62.0
SIFT+BOF (Yi y Tian, 2014)	-	72.6
PCH+BOF (Zhu *et al.*, 2016a)	76.4	-
Propuesto	**78.0**	**63.2**

Cabe destacar que el conjunto Chars74K no es uniforme, es decir, cada caracter contiene un conjunto con un número diferente de imágenes, desde 30 hasta 500 imágenes por caracter.

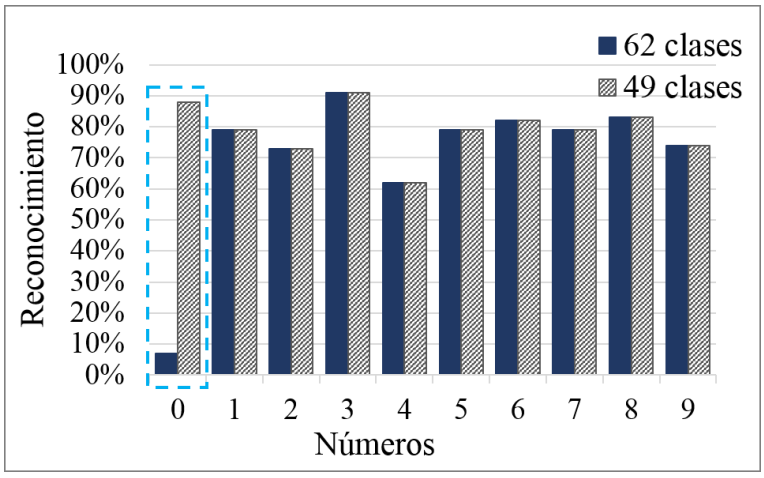


Figura 51. Resultado del reconocimiento de números en el conjunto Img-Chars74K.

Las Figuras 51, 52 y 53 muestran el desempeño del método propuesto por caracter,

utilizando 49 y 62 clases. Los rectángulos punteados muestran las clases agrupadas.

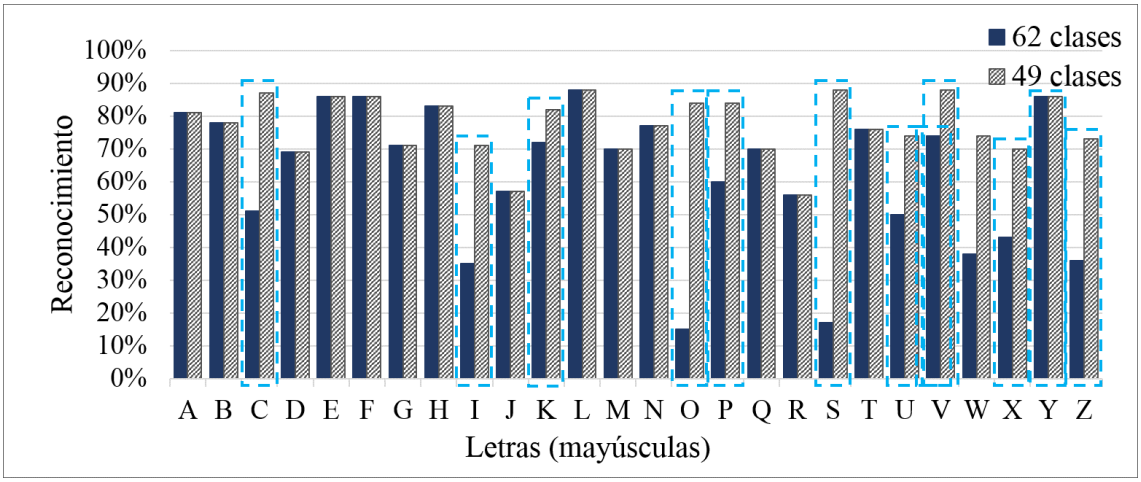


Figura 52. Resultado del reconocimiento de letras en mayúsculas en el conjunto Img-Chars74K.

Como se puede observar en la Figura 51, la clase “cero” obtuvo el menor porcentaje de reconocimiento debido a su gran parecido geométrico con los caracteres “o” y “O”. Agrupando estos tres caracteres en una sola clase se puede alcanzar hasta un 88 % de reconocimiento. En el caso de la clase “cuatro”, el número de imágenes de prueba era muy bajo (47 imágenes) y algunas eran caracteres escritos “a mano”, por lo que el reconocimiento disminuyó considerablemente.

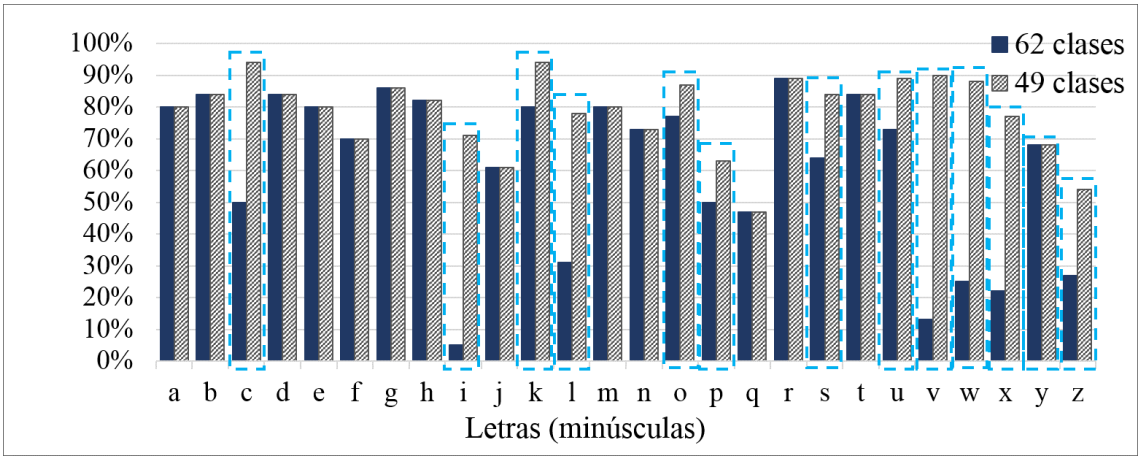


Figura 53. Resultado del reconocimiento de letras en minúsculas en el conjunto Img-Chars74K.

Para el caso de letras en mayúsculas con 62 clases (Figura 52), la mayoría de los caracteres obtuvieron un reconocimiento del 70 %, exceptuando aquellos que son semejantes a sus iguales en minúsculas (C, I, K, O, S, U, V, W, X, Y, Z). Agrupando las clases de los caracteres geométricamente semejantes (49 clases), el reconocimiento

de los caracteres aumentó casi hasta un 50 %. Se puede observar un comportamiento semejante para el caso de letras en minúsculas (Figura 53).

En este Capítulo se propuso un nuevo método para el reconocimiento de caracteres utilizando el descriptor propuesto LUIFT basado en la información de fase local de la imagen y el enfoque de bolsa de características. Los resultados de la evaluación permiten concluir que el método propuesto muestra robustez a ligeras distorsiones geométricas, ruido, contraste, desenfoque, oclusión, así como a diferentes tipos de fuente.

Capítulo 7. Sistema de *Principio-a-Fin*

En este Capítulo, se presenta la evaluación final del sistema de principio-a-fin de detección y reconocimiento de texto en imágenes naturales, así como su comparación con el estado del arte. Como se mencionó anteriormente, los métodos de detección y reconocimiento de texto de principio-a-fin constan de tres etapas, detección, reconocimiento y corrección de errores. En los Capítulos 5 y 6 se abordaron y describieron las etapas de detección y reconocimiento de texto propuestas, respectivamente. Sin embargo, la etapa de corrección de errores puede estar o no presente en el sistema. En este trabajo de investigación, se incluyó una etapa de corrección de errores la cual se describe a continuación.

7.1. Corrección de errores

La etapa de corrección de errores se encarga de corregir los errores de las etapas anteriores, tales como el reconocimiento incorrecto de un caracter, el agrupamiento de dos o más palabras como una sola, o la detección de un componente de no-texto. Estos errores pueden ser corregidos haciendo uso de diccionarios y técnicas de comparación de secuencias de caracteres.

Básicamente, en la etapa de corrección de errores, se comparan las palabras obtenidas con las palabras pertenecientes a un diccionario o conjunto de diccionarios utilizando tres diferentes métricas: distancia de edición ($Dist_E$) (Levenshtein, 1966), distancia de Hamming ($Dist_H$) y valor de frecuencia de palabra ($Dist_F$). Para nuestro sistema, utilizamos dos diccionarios, un diccionario con un listado de palabras y su número de frecuencia[1] y un diccionario genérico[2]. A continuación se resumen los pasos para la corrección de errores.

7.2. Etapa de corrección de errores

1. La palabra a evaluar se compara con todas las palabras del diccionario.
2. Si la palabra evaluada existe en el diccionario ($Dist_H$), regresa la palabra original y termina el algoritmo, en caso contrario continúa con el siguiente paso.

[1] https://www.wordfrequency.info/intro.asp
[2] http://www.robots.ox.ac.uk/ vgg/data/text/

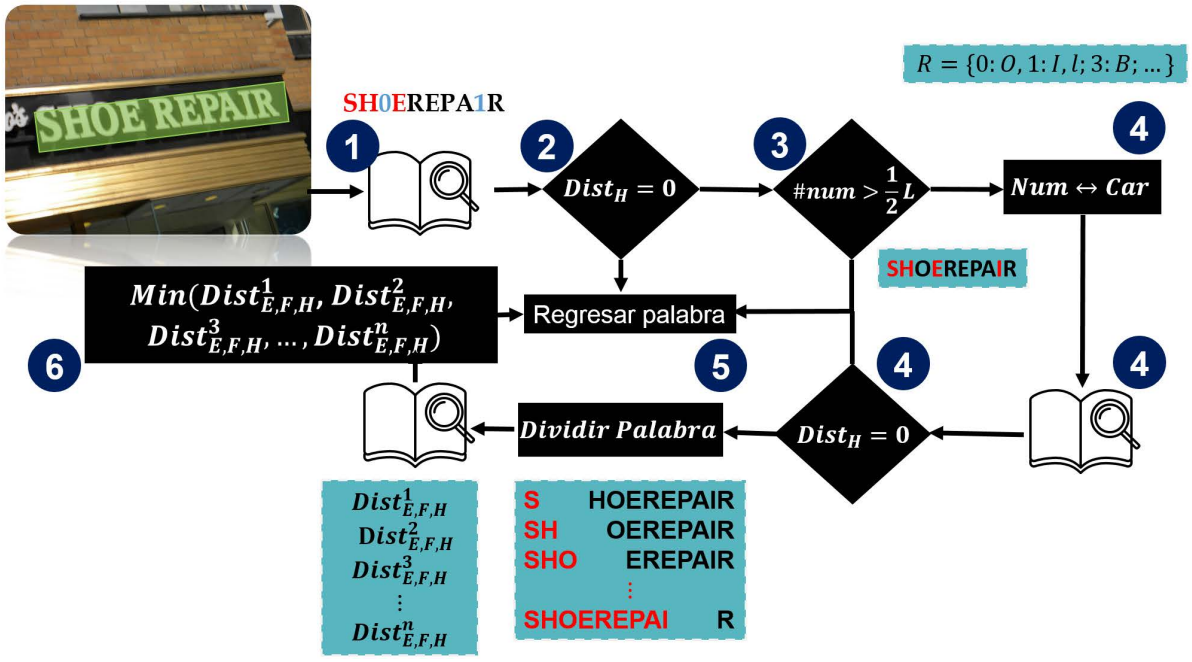


Figura 54. Ejemplo del algoritmo para la etapa de corrección de errores.

3. Revisa si el porcentaje de números contenidos en las palabra evaluada supera un umbral $thr_{num} = 0.6$. En caso afirmativo, regresa la palabra original y termina el algoritmo, en caso contrario continúa con el siguiente paso.

4. Si el porcentaje de números contenidos en la palabra evaluada no supera el umbral thr_{num}, entonces se reemplazan los números por los caracteres especificados en la lista subs = {1:i, 1:l, 5:s, 6:G, 8:B, 0:o} creando una nueva palabra con las sustituciones aplicadas, nuevamente se revisa si la palabra se encuentra en el diccionario. En caso de que la palabra sea encontrada (distancia de Levenshtein cero), la palabra se devuelve y termina el algoritmo, en caso contrario continúa con el siguiente paso.

5. La palabra se divide en dos sub-cadenas, haciendo todas las divisiones no vacías posibles. Para cada sub-cadena se repiten los pasos anteriores, sólo que en lugar de finalizar el algoritmo, por cada sub-cadena, se escoge el par de correspondencias con menor distancia de Levenshtein conjunta y se almacenan las mejores sub-cadenas en una nueva lista con su distancia de Levenshtein, su distancia Hamming y su número de frecuencia ($Dist^i_{E,F,H}, i = 1, 2, \ldots, N$).

6. Si la lista anterior no está vacía, se ordena la nueva lista de opciones y se de-

vuelve la mejor correspondencia según el orden[3]. Si está vacía, se devuelve la palabra original antes de la sustitución numérica y termina el algoritmo.

La Figura 54 muestra un ejemplo del algoritmo de corrección de errores.

7.3. Sistema propuesto

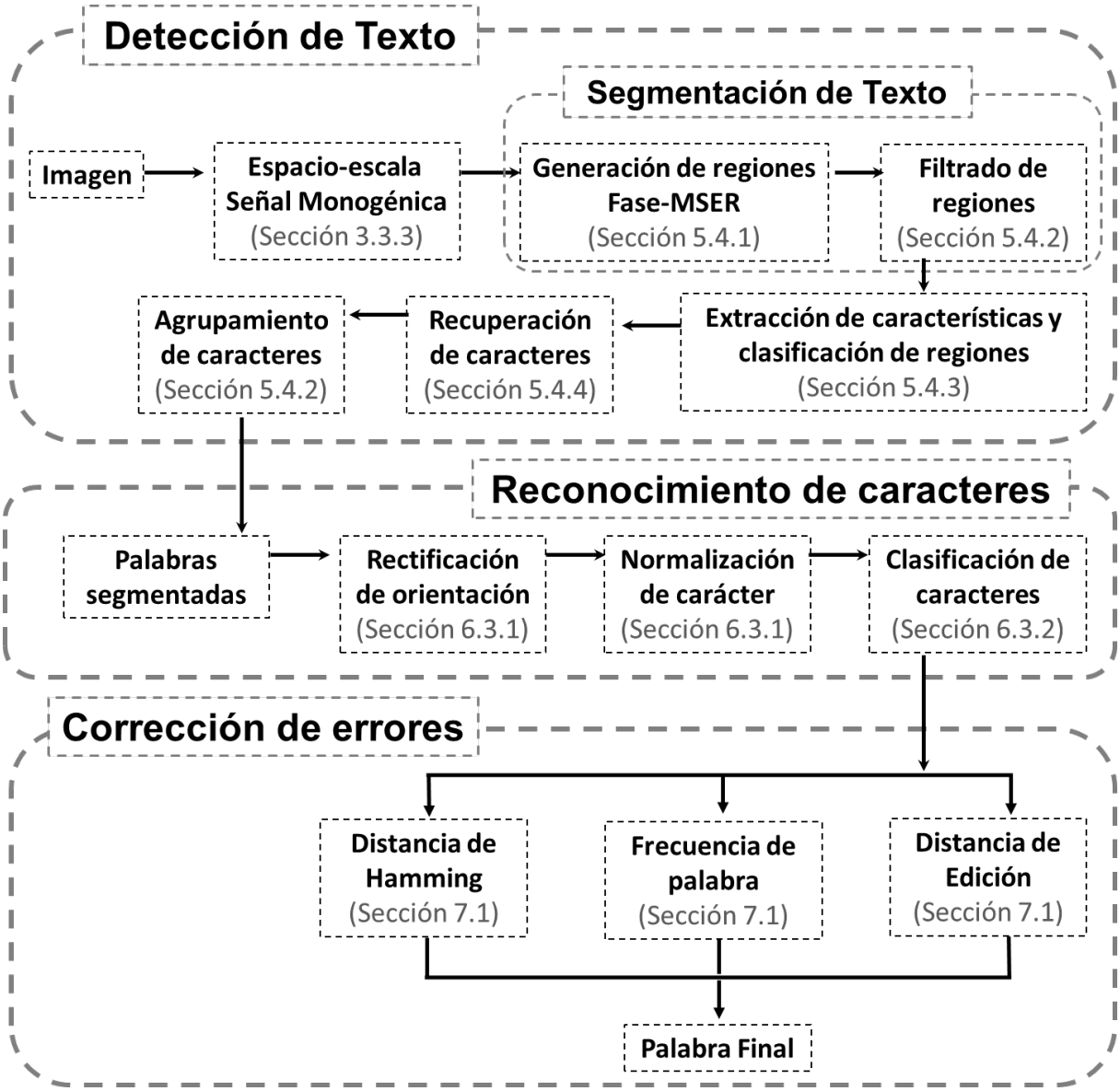


Figura 55. Diagrama del sistema propuesto de detección y reconocimiento de texto.

Básicamente, el sistema de principio-a-fin propuesto consta de tres etapas prin-

[3]el orden es por distancia de levenshtein ascendente, conteo en el corpus descendente, distancia de Hamming ascendente.

cipales: (1) la etapa de detección de texto, en la que los componentes del texto se segmentan y clasifican utilizando la información de fase local y la técnica MSER (Capítulo 5); (2) la etapa de reconocimiento de palabras, en la que los caracteres se reconocen utilizando el descriptor propuesto LUIFT y el enfoque BOF (Capítulo 6); y (3) la etapa de corrección de errores, en donde se corrigen errores de las etapas anteriores utilizando diccionarios. La Figura 55 muestra un diagrama del sistema propuesto.

7.4. Resultados

El sistema de principio-a-fin propuesto fue evaluado en los conjuntos de imágenes OSTD (Sección 2.2.3) y el conjunto ICDAR2013 (Sección 2.2.4). La Tabla 16 muestra los resultados de la evaluación del sistema por etapa, en el conjunto de imágenes OSTD en términos de sensibilidad (R), precisión (P) y medida F (F).

Tabla 16. Resultados del sistema propuesto en el conjunto OSTD (%)

Etapa	R	P	F
Segmentación	91.3	92.2	90.0
Localización	94.5	89.8	90.0
Reconocimiento	41.5	40.0	40.2
Corrección de errores	71.0	67.5	68.1

El método propuesto obtiene una desempeño del 90 % en las etapas de segmentación y localización del texto. Sin embargo, en la etapa de reconocimiento de palabras disminuye, debido a que sólo se consideraron para el reconocimiento de caracteres las transformaciones afines (escalamientos y rotaciones). Por lo que, los caracteres que presentan distorsiones de perspectiva no son bien reconocidos. Finalmente, la etapa de corrección de errores mejora el desempeño del método hasta un 20 %. La Figura. 56 muestra ejemplos del desempeño del sistema propuesto en el conjunto de imágenes OSTD.

Finalmente se evaluó el sistema propuesto en el conjunto de imágenes ICDAR2013. La Tabla 15 muestra los resultados obtenidos y la comparación de resultados con el estado del arte en términos de la medida F (F).

La Tabla 17 muestra los resultados del sistema propuesto en el conjunto ICDAR2013 comparado con algunos métodos del estado del arte. El método propuesto supera a

Figura 56. Ejemplos de resultados de sistema propuesto en conjunto OSTD.

Tabla 17. Comparación con el estado del arte en el conjunto ICDAR2013 (%)

Método	F
Bušta *et al.* (2017)	77.0
Neumann y Matas (2016)	54.0
Jaderberg *et al.* (2016)	76.0
Propuesto	**65.0**

todos los métodos comparados. A pesar de que los métodos Jaderberg *et al.* (2014) y Bušta *et al.* (2017) obtuvieron un mejor desempeño, estos métodos están basados en DNNs, sólo funcionan para texto en horizontal y no reconocen caracteres aislados. Además el número de imágenes de entrenamiento es mucho mayor al utilizado en nuestro método.

La Figura 57 muestra algunos ejemplos de los resultados obtenidos con el sistema propuesto en el conjunto de imágenes ICDAR2013.

Por otro, lado la Figura 58 muestra algunos ejemplos de los errores obtenidos con el método propuesto en el mismo conjunto. La mayoría de los errores que presenta el método propuesto se deben a la presencia de caracteres conectados o cercanos

Figura 57. Ejemplos de resultados de sistema propuesto en conjunto ICDAR2013.

(letra manuscrita), al tamaño de la fuente (menor que el valor mínimo definido en el sistema) o la presencia de reflejos o iluminación directa (flash) que hacen que se pierda la imagen por completo.

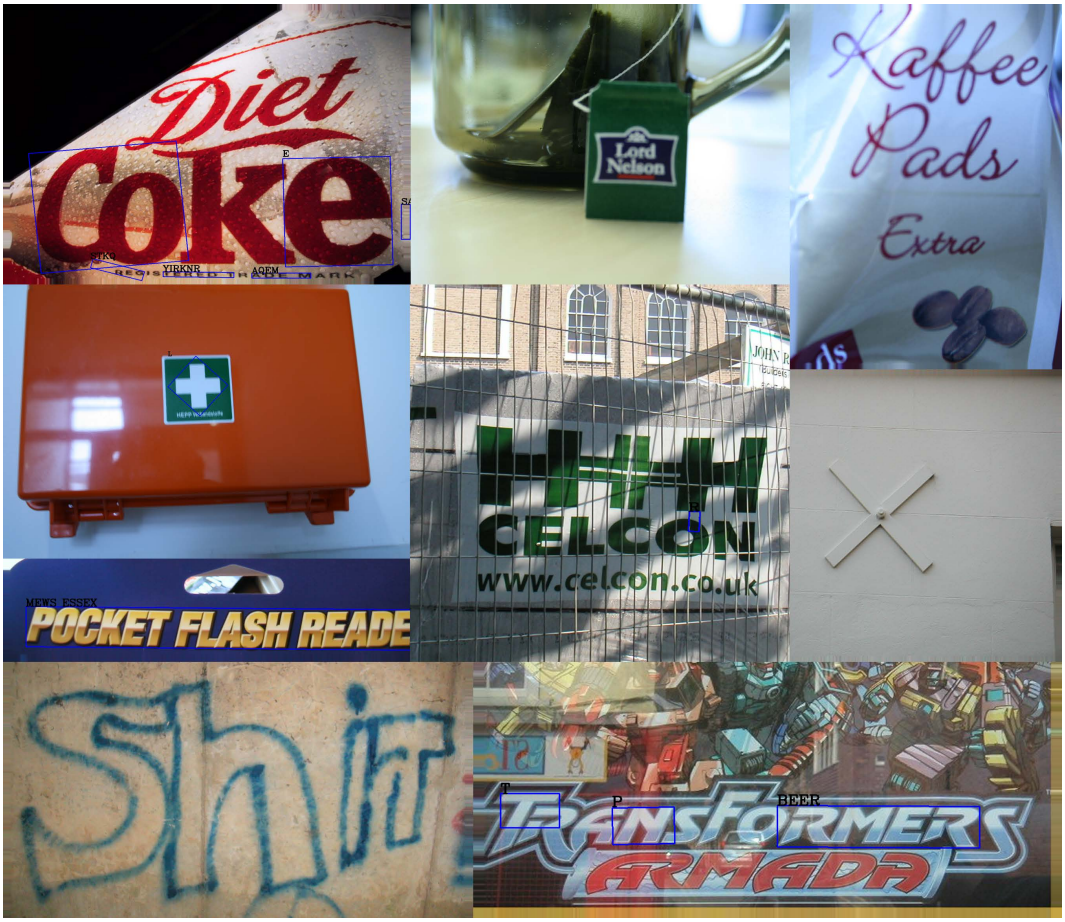


Figura 58. Ejemplos de errores del sistema propuesto en conjunto ICDAR2013.

Capítulo 8. Conclusiones

A pesar de la existencia en la literatura de diversos sistemas de principio-a-fin, la tarea de reconocimiento y detección de texto en imágenes naturales sigue siendo un problema abierto. Muchos de los sistemas actuales consideran imágenes “perfectas” en su diseño. Sin embargo, las imágenes utilizadas en aplicaciones actuales, esto no se cumple, ya que las imágenes naturales suelen presentar distorsiones, degradaciones y/o fondos complicados debido a la naturaleza de los dispositivos de captura que existen hoy en día. Por esta razón, en este trabajo de tesis se estudió el problema de detección y reconocimiento de texto multi-orientado en imágenes naturales. A lo largo de esta tesis, se investigaron y desarrollaron distintos métodos para atacar cada uno de los sub-problemas que componen al problema completo. Primero, se atacó el problema de detección y segmentación de texto (Capítulo 5). Después, se atacó el problema del reconocimiento de caracteres y se propuso un descriptor llamado LUminance Invariant Feature Transform (LUIFT) (Capítulo 6). Posteriormente, se desarrolló un algoritmo para la corrección de errores producidos en la etapa de reconocimiento de caracteres y agrupación de palabras. Finalmente, se integro el sistema completo de principio-a-fin utilizando las etapas desarrollas anteriormente (Capítulo 7).

La etapa de detección y segmentación de texto multi-orientado se basa en el modelo de energía local y la técnica MSER. El sistema consiste de tres etapas: segmentación, recuperación y agrupamiento de caracteres en palabras. El sistema propuesto demostró ser robusto a distorsiones geométricas, variaciones de fuentes, fondos complejos, bajo contraste, alto brillo, sombras y cambios de iluminación. También logró un alto desempeño de segmentación de caracteres con un número pequeño de regiones extraídas. El método superó los algoritmos del estado del arte en conjuntos de imágenes comunes en términos de segmentación de caracteres, localización de texto y número de regiones candidatas. Además, nuestro método no se limita a sólo texto alineado horizontalmente como la mayoría de los métodos existentes, sino también a textos multi-orientados.

También se atacó el problema de reconocimiento de caracteres (Capítulo 6). Para este problema, se propuso un descriptor de características robusto a cambios en iluminación y ligeras distorsiones geométricas, llamado LUIFT (Capítulo 4). A diferencia de los descriptores existentes que utilizan los valores de intensidad de la imagen para

su diseño, el descriptor propuesto LUIFT se basa en la información de fase local, lo cual brinda una mayor robustez a cambios en la iluminación de la imagen, al desenfoque y a la presencia de ruido aditivo. Este descriptor fue utilizado para representar a los caracteres y, utilizando el modelo de bolsa de características, clasificar a los caracteres. El método sugerido demostró ser robusto a ligeras distorsiones geométricas, oclusión y degradaciones tales como iluminación no uniforme,desenfoque, ruido y baja resolución. Además, el método propuesto obtuvo un alto desempeño de reconocimiento utilizando pocas imágenes de entrenamiento.

Como última etapa, se diseñó un algoritmo para la corrección de errores de reconocimiento utilizando diccionarios. Esto con el objetivo de corregir palabras o eliminar regiones reconocidas erróneamente. Finalmente se desarrolló un sistema de principio-a-fin para la detección y reconocimiento de texto multi-orientado en imágenes naturales (Capítulo 7). El sistema propuesto consta de tres etapas principales: (1) la etapa de detección, segmentación y localización del texto; (2) la etapa de reconocimiento de caracteres; y (3) la etapa de corrección de errores. El método fue evaluado utilizando conjuntos de imágenes comunes y comparado con técnicas del estado del arte. El sistema propuesto demostró ser robusto a distorsiones geométricas, estilos y tamaños de fuente diferentes, baja resolución y contraste, iluminación no uniforme y sombras.

El sistema desarrollado en este trabajo de investigación logró obtener resultados competitivos e incluso mejores que métodos basados en aprendizaje profundo. A diferencia de dichos métodos, las técnicas propuestas fueron diseñadas basándonos en conocimiento teórico y desarrolladas eficientemente con el objetivo de utilizar poca información pero obtener una mayor robustez. Consideramos que nuestros métodos permiten tener otra visión del desarrollo de técnicas que extraigan la información relevante de las imágenes sin necesidad de utilizar grandes conjuntos para lograr un buen desempeño, como es el caso del enfoque de aprendizaje profundo.

Como trabajo futuro, el sistema propuesto podría utilizarse para texto en otros alfabetos o textos escritos a mano. Incluso, podría ser utilizado para el reconocimiento de otros objetos. El sistema podría ser mejorado realizando descomposición piramidal de la imagen y/o utilizar otros espacios de color. Finalmente, el sistema podría implementarse en GPU para su ejecución en tiempo real.

Literatura citada

Agrawal, M., Konolige, K., y Blas, M. R. (2008). Censure: Center surround extremas for realtime feature detection and matching. En: *Conference on Computer Vision*. Springer, pp. 102–115.

Aguilar-González, P. M., Kober, V., y Díaz-Ramírez, V. H. (2014). Adaptive composite filters for pattern recognition in nonoverlapping scenes using noisy training images. *Pattern Recognition Letters*, **41**: 83–92.

Alahi, A., Ortiz, R., y Vandergheynst, P. (2012). Freak: Fast retina keypoint. En: *Conference on Computer vision and pattern recognition*. IEEE, pp. 510–517.

Alcantarilla, P. F., Bartoli, A., y Davison, A. J. (2012). Kaze features. En: *Conference on Computer Vision*. Springer, pp. 214–227.

Angadi, S. y Kodabagi, M. (2009). A texture based methodology for text region extraction from low resolution natural scene images. *International Journal of Image Processing (IJIP)*, **3**(5): 229.

Attneave, F. (1954). Some informational aspects of visual perception. *Psychological review*, **61**(3): 183.

Bai, X., Yao, C., y Liu, W. (2016). Strokelets: A learned multi-scale mid-level representation for scene text recognition. *IEEE Transactions on Image Processing*, **25**(6): 2789–2802.

Bay, H., Tuytelaars, T., y Van Gool, L. (2006). Surf: Speeded up robust features. En: *Conference on Computer Vision*. Springer, pp. 404–417.

Bay, H., Ess, A., Tuytelaars, T., y Van Gool, L. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, **110**(3): 346–359.

Beaudet, P. (1987). Rotationally invariant image operators. En: *International Joint Conference on Pattern Recognition*. pp. 579–583.

Bellavia, F., Tegolo, D., y Valenti, C. (2011). Improving harris corner selection strategy. *IET Computer Vision*, **5**(2): 87–96.

Belongie, S., Malik, J., y Puzicha, J. (2002). Shape matching and object recognition using shape contexts. Reporte técnico, CALIFORNIA UNIV SAN DIEGO LA JOLLA DEPT OF COMPUTER SCIENCE AND ENGINEERING.

Berg, A. C., Berg, T. L., y Malik, J. (2005). Shape matching and object recognition using low distortion correspondences. En: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, Vol. 1, pp. 26–33.

Bissacco, A., Cummins, M., Netzer, Y., y Neven, H. (2013). Photoocr: Reading text in uncontrolled conditions. En: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 785–792.

Bosch, A., Zisserman, A., y Munoz, X. (2007). Image classification using random forests and ferns. En: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, pp. 1–8.

Busta, M., Neumann, L., y Matas, J. (2015). Fastext: Efficient unconstrained scene text detector, 2015. En: *IEEE International Conference on Computer Vision (ICCV)*. Vol. 1.

Bušta, M., Neumann, L., y Matas, J. (2017). Deep textspotter: An end-to-end trainable scene text localization and recognition framework. En: *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, pp. 2223–2231.

Calonder, M., Lepetit, V., Strecha, C., y Fua, P. (2010). Brief: Binary robust independent elementary features. En: *Conference on Computer Vision*. Springer, pp. 778–792.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6): 679–698.

Carneiro, G. y Jepson, A. D. (2002). Phase-based local features. En: *European Conference on Computer Vision*. Springer, pp. 282–296, [doi:10.1007/3-540-47969-4_19].

Casasent, D. (1984). Unified synthetic discriminant function computational formulation. *Applied Optics*, **23**(10): 1620–1627.

Clavelli, A., Karatzas, D., y Lladós, J. (2010). A framework for the assessment of text extraction algorithms on complex colour images. En: *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, pp. 19–26.

Cormen, T. H. (2009). *Introduction to algorithms*. MIT press.

Cortes, C. y Vapnik, V. (1995). Support-vector networks. *Machine learning*, **20**(3): 273–297.

Cristianini, N., Shawe-Taylor, J., *et al.* (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Dalal, N. y Triggs, B. (2005). Histograms of oriented gradients for human detection. En: *Conference on Computer Vision and Pattern Recognition*. IEEE, Vol. 1, pp. 886–893.

De Campos, T. E., Babu, B. R., Varma, M., *et al.* (2009). Character recognition in natural images. *VISAPP (2)*, **7**.

Deselaers, T., Keysers, D., y Ney, H. (2008). Features for image retrieval: an experimental comparison. *Information Retrieval*, **11**(2): 77–107.

Diaz-Escobar, J. y Kober, V. (2015a). Optical character recognition based on phase features. En: *Computing Systems and Telematics (ICCSAT), 2015 International Conference on*. IEEE, pp. 1–5.

Diaz-Escobar, J. y Kober, V. (2015b). Optical character recognition of camera-captured images based on phase features. En: *Applications of Digital Image Processing XXXVIII*. International Society for Optics and Photonics, Vol. 9599, p. 959903.

Diaz-Escobar, J. y Kober, V. (2016a). A robust hog-based descriptor for pattern recognition. En: *Applications of Digital Image Processing XXXIX*. International Society for Optics and Photonics, Vol. 9971, p. 99712A.

Diaz-Escobar, J. y Kober, V. (2016b). Text detection in digital images captured with low resolution under nonuniform illumination conditions. En: *Mexican Conference on Pattern Recognition*. Springer, pp. 3–12.

Diaz-Escobar, J. y Kober, V. (2017). Text detection in natural scenes with phase congruency approach. En: *Applications of Digital Image Processing XL*. International Society for Optics and Photonics, Vol. 10396, p. 1039637.

Diaz-Escobar, J. y Kober, V. (2018a). Natural scene text detection and recognition with a three-stage local phase-based algorithm. En: *Applications of Digital Image Processing XLI*. International Society for Optics and Photonics.

Diaz-Escobar, J. y Kober, V. (2018b). Scene text segmentation based on local image phase information and mser method. En: *Mexican Conference on Pattern Recognition*. Springer, pp. 211–220.

Diaz-Escobar, J. y Kober, V. (2018c). Natural scene text detection and segmentation using phase-based regions and character retrieval. *Neurocomputing (Under Review)*.

Diaz-Escobar, J., Kober, V., y Karnaukhov, V. (2015). Character recognition in degraded document images using morphological and phase-only filtering. *Journal of Communications Technology and Electronics*, **60**(12): 1360–1365.

Diaz-Escobar, J., Kober, V., y Gonzalez-Fraga, J. (2018a). Luift: Luminance invariant feature transform. *Mathematical Problems in Engineering*.

Diaz-Escobar, J., Kober, V., Karnaukhovb, V., y Gonzalez-Fraga, J. (2018b). A new invariant to illumination feature descriptor for pattern recognition. *Mathematical Model and Computational Methods*.

Diaz-Ramirez, V. H. y Kober, V. (2009). Target recognition under nonuniform illumination conditions. *Applied optics*, **48**(7): 1408–1418.

Doh, Y.-H., Kim, J.-C., Kim, J.-W., Choi, K.-H., Kim, S.-J., y Alam, M. S. (2004). Distortion-invariant pattern recognition based on a synthetic hit-miss transform. *Optical Engineering*, **43**(8): 1798–1803.

Epshtein, B., Ofek, E., y Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. En: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pp. 2963–2970.

Fei-Fei, L. y Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. En: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, Vol. 2, pp. 524–531.

Felsberg, M. (2002). Low-level image processing with the structure multivector.

Felsberg, M. y Sommer, G. (2000). A new extension of linear signal processing for estimating local properties and detecting features. En: *Mustererkennung 2000*. Springer, pp. 195–202.

Felsberg, M. y Sommer, G. (2001). The monogenic signal. *IEEE Transactions on Signal Processing*, **49**(12): 3136–3144. [doi:10.1109/78.969520].

Felsberg, M. y Sommer, G. (2004). The monogenic scale-space: A unifying approach to phase-based image processing in scale-space. *Journal of Mathematical Imaging and vision*, **21**(1): 5–26.

Fitch, J., Coyle, E. J., y Gallagher Jr, N. C. (1984). Median filtering by threshold decomposition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, **32**(6): 1183–1188.

Fortun, D., Bouthemy, P., y Kervrann, C. (2015). Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*, **134**: 1–21.

Freund, Y. y Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, **55**(1): 119–139.

Gladilin, E. y Eils, R. (2015). On the role of spatial phase and phase correlation in vision, illusion, and cognition. *Frontiers in computational neuroscience*, **9**: 45.

González, R. y Woods, R. (2006). *Digital Image Processing, (3a ed.)*. Prentice-Hall Inc. Upper Saddle River, NJ, USA.

González-Fraga, A. J., Kober, V. I., y Álvarez-Borrego, J. (2006). Adaptive synthetic discriminant function filters for pattern recognition. *Optical Engineering*, **45**(5): 057005.

Granlund, G. H. y Knutsson, H. (2013). *Signal processing for computer vision*. Springer Science & Business Media.

Gupta, R., Patil, H., y Mittal, A. (2010). Robust order-based methods for feature description. En: *Conference on Computer Vision and Pattern Recognition*. IEEE.

Han, J., Pei, J., y Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Harris, C. y Stephens, M. (1988). A combined corner and edge detector. En: *Alvey Vision Conference*. Citeseer, Vol. 15, pp. 10–5244.

He, W., Zhang, X.-Y., Yin, F., y Liu, C.-L. (2017). Deep direct regression for multi-oriented scene text detection. *arXiv preprint arXiv:1703.08289*.

Heikkilä, M., Pietikäinen, M., y Schmid, C. (2009). Description of interest regions with local binary patterns. *Pattern Recognition*, **42**(3): 425–436.

Horn, B. K. (1990). Height and gradient from shading. *International journal of computer vision*, **5**(1): 37–75.

Huang, W., Lin, Z., Yang, J., y Wang, J. (2013). Text localization in natural images using stroke feature transform and text covariance descriptors. En: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1241–1248.

Hubel, D. H. y Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, **160**(1): 106–154. [doi:10.1113/jphysiol.1962.sp006837].

Hubel, D. H., Wensveen, J., y Wick, B. (1995). *Eye, brain, and vision*. Scientific American Library New York.

Jaderberg, M., Vedaldi, A., y Zisserman, A. (2014). Deep features for text spotting. En: *European conference on computer vision*. Springer, pp. 512–528.

Jaderberg, M., Simonyan, K., Vedaldi, A., y Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, **116**(1): 1–20.

Jain, A., Ross, A. A., y Nandakumar, K. (2011). *Introduction to biometrics*. Springer Science & Business Media.

Karaoglu, S., Fernando, B., y Trémeau, A. (2010). A novel algorithm for text detection and localization in natural scene images. En: *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*. IEEE, pp. 635–642.

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L. G., Mestre, S. R., Mas, J., Mota, D. F., Almazan, J. A., y De Las Heras, L. P. (2013). Icdar 2013 robust reading competition. En: *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, pp. 1484–1493.

Kaur, A., Dhir, R., y Lehal, G. S. (2017). A survey on camera-captured scene text detection and extraction: towards gurmukhi script. *International Journal of Multimedia Information Retrieval*, **6**(2): 115–142.

Kim, K. I., Jung, K., y Kim, J. H. (2003). Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(12): 1631–1639.

Kovesi, P. (1999). Image features from phase congruency. *Videre: Journal of computer vision research*, **1**(3): 1–26.

Kovesi, P. (2000). Phase congruency: A low-level image invariant. *Psychological research*, **64**(2): 136–148. [doi:10.1007/s004260000024].

Kovesi, P. *et al.* (2002). Edges are not just steps. En: *Proceedings of the Fifth Asian Conference on Computer Vision*. Melbourne, Vol. 8, pp. 22–8.

Kumar, B. V., Mahalanobis, A., y Juday, R. D. (2005). *Correlation pattern recognition*, Vol. 27. Cambridge University Press Cambridge.

Kumar, B. V. V., Savvides, M., y Xie, C. (2006). Correlation pattern recognition for face recognition. *Proceedings of the IEEE*, **94**(11): 1963–1976.

Laerty, J., McCallum, A., y Pereira, F. (2001). Conditional random elds: Probabilistic models for segmenting and labeling sequence data. En: *Proceedings of ICML*.

Lazebnik, S., Schmid, C., y Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(8): 1265–1278.

Lee, C.-Y. y Osindero, S. (2016). Recursive recurrent nets with attention modeling for ocr in the wild. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2231–2239.

Leutenegger, S., Chli, M., y Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. En: *Conference on Computer Vision*. IEEE, pp. 2548–2555.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. En: *Soviet physics doklady*. Vol. 10, pp. 707–710.

Levi, G. y Hassner, T. (2016). Latch: learned arrangements of three patch codes. En: *Conference on Applications of Computer Vision*. IEEE, pp. 1–9.

Li, Y., Jia, W., Shen, C., y van den Hengel, A. (2014). Characterness: An indicator of text in the wild. *IEEE transactions on image processing*, **23**(4): 1666–1677.

Liu, S. y Bai, X. (2012). Discriminative features for image classification and retrieval. *Pattern Recognition Letters*, **33**(6): 744–751.

Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., y Yan, J. (2018). Fots: Fast oriented text spotting with a unified network. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5676–5685.

Liu, Z. y Sarkar, S. (2008). Robust outdoor text detection using text intensity and shape features. En: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, pp. 1–4.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. En: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, Vol. 2, pp. 1150–1157.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**(2): 91–110.

Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., y Xue, X. (2018). Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*.

Maragos, P. (1989). Morphological correlation and mean absolute error criteria. *ASSP, Int. Conf. on. IEEE*, pp. 1568–15721.

Maragos, P. (2004). Morphological filtering for image enhancement and feature detection. *analysis*, **19**: 18.

Matas, J. y Zimmermann, K. (2005). A new class of learnable detectors for categorisation. *Image Analysis*, pp. 541–550.

Matas, J., Chum, O., Urban, M., y Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, **22**(10): 761–767.

Mikolajczyk, K. y Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, **60**(1): 63–86.

Mikolajczyk, K. y Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(10): 1615–1630.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., y Van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, **65**(1-2): 43–72.

Mishra, A., Alahari, K., y Jawahar, C. (2012). Top-down and bottom-up cues for scene text recognition. En: *CVPR-IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Molina, E., Diaz, J., Hidalgo-Silva, H., y Chávez, E. (2017). Algoritmos de binarizacion robusta de imágenes con iluminación no uniforme. *Revista Iberoamericana de Automática e Informática industrial*.

Moravec, H. P. (1979). Visual mapping by a robot rover. En: *Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., pp. 598–600.

Moreels, P. y Perona, P. (2007). Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, **73**(3): 263–284.

Morrone, M. C. y Burr, D. (1988). Feature detection in human vision: A phase-dependent energy model. *Proceedings of the Royal Society of London B: Biological Sciences*, **235**(1280): 221–245.

Morrone, M. C. y Owens, R. A. (1987). Feature detection from local energy. *Pattern recognition letters*, **6**(5): 303–313.

Muja, M. y Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, **2**(331-340): 2.

Mukundan, R. y Ramakrishnan, K. (1998). *Moment functions in image analysis-theory and applications*. World Scientific.

Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., y Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, **12**(2): 181–201.

Neumann, L. y Matas, J. (2012). Real-time scene text localization and recognition. En: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 3538–3545.

Neumann, L. y Matas, J. (2016). Real-time lexicon-free scene text localization and recognition. *IEEE transactions on pattern analysis and machine intelligence*, **38**(9): 1872–1885.

Nguyen, A., Yosinski, J., y Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 427–436.

Novikova, T., Barinova, O., Kohli, P., y Lempitsky, V. (2012). Large-lexicon attribute-consistent text recognition in natural images. En: *European Conference on Computer Vision*. Springer, pp. 752–765.

Ntirogiannis, K., Gatos, B., y Pratikakis, I. (2008). An objective evaluation methodology for document image binarization techniques. En: *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*. IEEE, pp. 217–224.

O'Hara, S. y Draper, B. A. (2011). Introduction to the bag of features paradigm for image classification and retrieval. *arXiv preprint arXiv:1101.3354*.

Oppenheim, A. V. y Lim, J. S. (1981). The importance of phase in signals. *Proceedings of the IEEE*, **69**(5): 529–541.

Pal, C., Sutton, C., y McCallum, A. (2006). Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. En: *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, Vol. 5, pp. V–V.

Pan, Y.-F., Hou, X., y Liu, C.-L. (2008). A robust system to detect and localize texts in natural scene images. En: *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*. IEEE, pp. 35–42.

Papari, G. y Petkov, N. (2011). Edge and line oriented contour detection: State of the art. *Image and Vision Computing*, **29**(2): 79–103.

Redmon, J. y Farhadi, A. (2017). Yolo9000: better, faster, stronger. *arXiv preprint*.

Robbins, B. y Owens, R. (1997). 2d feature detection via local energy. *Image and Vision Computing*, **15**(5): 353–368.

Rosten, E. y Drummond, T. (2006). Machine learning for high-speed corner detection. En: *Conference on Computer Vision*. Springer, pp. 430–443.

Saoi, T., Goto, H., y Kobayashi, H. (2005). Text detection in color scene images based on unsupervised clustering of multi-channel wavelet features. En: *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE, pp. 690–694.

Saric, M. (2017). Scene text segmentation using low variation extremal regions and sorting based character grouping. *Neurocomputing*.

Schapire, R. E. y Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine learning*, **37**(3): 297–336.

Simo-Serra, E., Torras, C., y Moreno-Noguer, F. (2015). Dali: deformation and light invariant descriptor. *International Journal of Computer Vision*, **115**(2): 136–154.

Smith, S. M. y Brady, J. M. (1997). Susan—a new approach to low level image processing. *International Journal of Computer Vision*, **23**(1): 45–78.

Sonka, M., Hlavac, V., y Boyle, R. (2014). *Image processing, analysis, and machine vision*. Cengage Learning.

Sung, M.-C., Jun, B., Cho, H., y Kim, D. (2015). Scene text detection with robust character candidate extraction method. En: *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, pp. 426–430.

Tang, F., Lim, S. H., Chang, N. L., y Tao, H. (2009). A novel feature descriptor invariant to complex brightness changes. En: *Conference on Computer Vision and Pattern Recognition*. IEEE.

Tang, P., Yuan, Y., Fang, J., y Zhao, Y. (2015). A novel similar background components connection algorithm for colorful text detection in natural images. En: *Signal Processing, Communications and Computing (ICSPCC), 2015 IEEE International Conference on*. IEEE, pp. 1–5.

Tang, S., Andriluka, M., y Schiele, B. (2014). Detection and tracking of occluded people. *International Journal of Computer Vision*, **110**(1): 58–69.

Tian, C., Xia, Y., Zhang, X., y Gao, X. (2017). Natural scene text detection with mc–mr candidate extraction and coarse-to-fine filtering. *Neurocomputing*, **260**: 112–122.

Tian, Z., Huang, W., He, T., He, P., y Qiao, Y. (2016). Detecting text in natural image with connectionist text proposal network. En: *European conference on computer vision*. Springer, pp. 56–72.

Tola, E., Lepetit, V., y Fua, P. (2008). A fast local descriptor for dense matching. En: *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Trier, O. D., Jain, A. K., Taxt, T., *et al.* (1996). Feature extraction methods for character recognition-a survey. *Pattern recognition*, **29**(4): 641–662.

Tuytelaars, T. y Mikolajczyk, K. (2008). Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, **3**(3): 177–280.

Unser, M., Sage, D., y Van De Ville, D. (2009). Multiresolution monogenic signal analysis using the riesz–laplace wavelet transform. *IEEE Transactions on Image Processing*, **18**(11): 2402–2418.

Varma, M. y Zisserman, A. (2002). Classifying images of materials: Achieving viewpoint and illumination independence. En: *European Conference on Computer Vision*. Springer, pp. 255–271.

Varma, M. y Zisserman, A. (2003). Texture classification: Are filter banks necessary? En: *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*. IEEE, Vol. 2, pp. II–691.

Venkatesh, S. y Owens, R. (1989). An energy feature detection scheme. En: *ICIP'89: IEEE International Conference on Image Processing: conference proceedings, 5-8 September 1989, Singapore*. IEEE.

Verdie, Y., Yi, K., Fua, P., y Lepetit, V. (2015). Tilde: a temporally invariant learned detector. En: *Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 5279–5288.

Vonikakis, V., Chrysostomou, D., Kouskouridas, R., y Gasteratos, A. (2013). A biologically inspired scale-space for illumination invariant feature detection. *Measurement Science and Technology*, **24**(7): 074024.

Wang, K. y Belongie, S. (2010). Word spotting in the wild. En: *European Conference on Computer Vision*. Springer, pp. 591–604.

Wang, K., Babenko, B., y Belongie, S. (2011a). End-to-end scene text recognition. En: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, pp. 1457–1464.

Wang, T., Wu, D. J., Coates, A., y Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. En: *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, pp. 3304–3308.

Wang, Z., Fan, B., y Wu, F. (2011b). Local intensity order pattern for feature description. En: *International Conference on Computer Vision*. IEEE, pp. 603–610.

Wei, Y., Shen, W., Zeng, D., Ye, L., y Zhang, Z. (2018). Multi-oriented text detection from natural scene images based on a cnn and pruning non-adjacent graph edges. *Signal Processing: Image Communication*, **64**: 89–98.

Weinman, J. J., Learned-Miller, E., y Hanson, A. R. (2009). Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, **31**(10): 1733–1746.

Wolf, C. y Jolion, J.-M. (2006). Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, **8**(4): 280–296.

Wu, H., Zou, B., Zhao, Y.-q., Chen, Z., Zhu, C., y Guo, J. (2016). Natural scene text detection by multi-scale adaptive color clustering and non-text filtering. *Neurocomputing*, **214**: 1011–1025.

Yao, C., Bai, X., Liu, W., Ma, Y., y Tu, Z. (2012). Detecting texts of arbitrary orientations in natural images. En: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 1083–1090.

Yao, C., Bai, X., y Liu, W. (2014). A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, **23**(11): 4737–4749.

Ye, Q. y Doermann, D. (2015). Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, **37**(7): 1480–1500.

Yi, C. y Tian, Y. (2011). Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, **20**(9): 2594–2605.

Yi, C. y Tian, Y. (2014). Scene text recognition in mobile applications by character descriptor and structure configuration. *IEEE transactions on image processing*, **23**(7): 2972–2982.

Yi, C., Yang, X., y Tian, Y. (2013). Feature representations for scene text character recognition: A comparative study. En: *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, pp. 907–911.

Yi, K. M., Trulls, E., Lepetit, V., y Fua, P. (2016). Lift: Learned invariant feature transform. En: *Conference on Computer Vision*. Springer, pp. 467–483.

Yin, X.-C., Yin, X., Huang, K., y Hao, H.-W. (2014). Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence*, **36**(5): 970–983.

Yin, X.-C., Pei, W.-Y., Zhang, J., y Hao, H.-W. (2015). Multi-orientation scene text detection with adaptive clustering. *IEEE transactions on pattern analysis and machine intelligence*, **37**(9): 1930–1937.

Yu, C., Song, Y., y Zhang, Y. (2016). Scene text localization using edge analysis and feature pool. *Neurocomputing*, **175**: 652–661.

Zhang, H., Zhao, K., Song, Y.-Z., y Guo, J. (2013). Text extraction from natural scene image: A survey. *Neurocomputing*, **122**: 310–323.

Zhao, Z., Fang, C., Lin, Z., y Wu, Y. (2015). A robust hybrid method for text detection in natural scenes by learning-based partial differential equations. *Neurocomputing*, **168**: 23–34.

Zheng, Y., Li, Q., Liu, J., Liu, H., Li, G., y Zhang, S. (2017). A cascaded method for text detection in natural scene images. *Neurocomputing*, **238**: 307–315.

Zhou, G., Liu, Y., Meng, Q., y Zhang, Y. (2011). Detecting multilingual text in natural scene. En: *Access Spaces (ISAS), 2011 1st International Symposium on*. IEEE, pp. 116–120.

Zhu, A., Wang, G., y Dong, Y. (2016a). Good initialization model with constrained body structure for scene text recognition. *Journal of Electronic Imaging*, **25**(5): 053018.

Zhu, Y., Yao, C., y Bai, X. (2016b). Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, **10**(1): 19–36.

Zitova, B. y Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, **21**(11): 977–1000.