



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**DIRECCIÓN DE FORMACIÓN POSTGRADUADA**

**Algoritmo basado en el análisis local de la Transformada Radon para  
corregir la inclinación en imágenes de documentos complejos**

**Tesis para optar por el Título de Máster en Informática Aplicada**

**Autor:** Ing. Rafael L. Cardero Álvarez  
**Tutor:** Msc. Héctor R. González Díez

**La Habana, julio de 2011**

**“Año 53 de la Revolución”**



## **DECLARACIÓN JURADA DE AUTORÍA**

Yo Rafael Leodan Cardero Álvarez, con carnet de identidad 84122624669, declaro que soy el único autor del resultado que se expone en esta tesis. La misma se titula "Algoritmo basado en el análisis local de la Transformada Radon para corregir la inclinación en imágenes de documentos complejos". Se presenta con el objetivo de optar por el título de Máster en Informática Aplicada.

La investigación cuyos resultados se plasman en este documento, fue desarrollada en el periodo 2010-2011.

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional. Para que así conste firmo la presente Declaración Jurada de Autoría, en La Habana, a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

(firma)

## SÍNTESIS CURRICULARES

**Ing. Rafael L. Cardero Álvarez:** Graduado de Ingeniero en Ciencias Informáticas por la Universidad de las Ciencias Informáticas, en el año 2008. Instructor en la docencia. Trabaja en la Facultad 6 de la Universidad de las Ciencias Informáticas. Ha recibido varios cursos sobre procesamiento de imágenes digitales. Imparte en la universidad los cursos de pregrado “Procesamiento de imágenes digitales I” y “Procesamiento de imágenes digitales II”. Ha participado en varios eventos realizados en Cuba, vinculados al procesamiento de imágenes digitales. Sus intereses de investigación son el procesamiento de imágenes digitales (en general), el análisis de imágenes de documentos, la segmentación de textos en fotogramas de video, y la dirección de industrias de la informática y las comunicaciones. Correo electrónico: [rlcordero@uci.cu](mailto:rlcordero@uci.cu).

**Msc. Héctor Raúl González Díez:** Graduado de Licenciado en Física Nuclear en el ISCTN. Asistente en la docencia. Trabaja en la Facultad 6 de la Universidad de las Ciencias Informáticas como Subdirector de Investigación y Postgrado del Centro GEySED. Ha recibido varios cursos sobre procesamiento de imágenes digitales. Imparte en la universidad los cursos de pregrado “Procesamiento de imágenes digitales I” y “Procesamiento de imágenes digitales II”. Ha participado en varios eventos realizados en Cuba, vinculados al procesamiento de imágenes digitales. Sus intereses de investigación son el análisis de videos, la segmentación temporal de videos, y el reconocimiento de patrones en videos. Correo electrónico: [hglez@uci.cu](mailto:hglez@uci.cu).



## **AGRADECIMIENTOS**

La energía que me ha impulsado durante la elaboración de esta tesis, proviene de manantiales profundos que emanan aguas de carácter personal, y no profesional. Debo ser honesto: no he escrito esta tesis para hacerme Máster en Informática Aplicada. He escrito esta tesis para rendir honor y amor, a quienes mi honor y amor merecen; y sobre todo, para corregir el más grande error que he cometido en los últimos 26 años. Es por ello que los agradecimientos no serán escritos aquí, en papel. Para ello he elegido otro medio, que es el mejor de todos. Los agradecimientos de esta tesis serán escritos durante el acto de defensa, en el corazón de los agradecidos.

**Rafael**

**Una vez más,**  
*a mis padres: por haberme educado, con tanto esfuerzo, del modo que hoy soy.*

**Con sobradas razones y emociones,**  
*a Yojita: por ti y para ti, he escrito esta tesis.*





## RESUMEN

Una imagen de documento complejo es aquella en la que el texto puede ser manuscrito o escrito a máquina, mostrar distintos tipos de letra, tamaños de letra e interlineados, y estar dispuesto en una o varias columnas. Además puede contener figuras, gráficos, tablas, ecuaciones, decoraciones, bordes de página y bordes laterales mezclados junto al texto. La corrección de la inclinación es un hito relevante durante el análisis de imágenes de documentos y los algoritmos existentes para corregir la inclinación brindan malos resultados al procesar imágenes de documentos complejos, lo cual impacta negativamente en varios sectores.

En esta tesis se propone un nuevo algoritmo que evoluciona del modo que sigue: 1) mejorar la imagen mediante un ajuste del contraste; 2) segmentar la imagen mediante el método de Otsu; 3) eliminar las componentes conectadas con mayor área; 4) estimar la inclinación analizando localmente la Transformada Radon de las regiones más significativas; 5) rotar la imagen original. Además se propone una metodología que sirve como guía al diseñar nuevos algoritmos que corrijan la inclinación. El algoritmo se probó en 11600 imágenes de documentos complejos con texto manuscrito y escrito a máquina. En las imágenes con texto manuscrito el porcentaje de aciertos fue del 100% y en las imágenes con texto escrito a máquina fue del 99,71%. Estos resultados demostraron que el algoritmo es capaz de arrojar buenos resultados al corregir la inclinación en imágenes de documentos complejos y que es apto para ser utilizado en las aplicaciones que lo requieran.

**Palabras claves:** análisis, corrección, DIA, documento, imagen, inclinación, radon, transformada.

## **ABSTRACT**

A complex document image contains handwritten or typewritten text which format varies in aspects such as style, size, spacing and column layout. The text could be mixed with objects like images, graphs, tables, equations, decorations, noisy borders and page borders. Skew correction is a crucial step during document image analysis and algorithms proposed up to date to correct the skew suffer many problems processing complex document images. This situation impacts negatively in many application fields.

In this thesis a new skew correction algorithm is proposed. It evolves as follows: 1) enhance the image via a contrast adjusting technique; 2) segment the image using the Otsu's method; 3) remove connected components having a big area; 4) estimate the skew analyzing locally the Radon Transform of the more representative regions; 5) rotate the original image. A methodology to apply during the skew correction process is also proposed. This methodology is very useful to guide the design of new skew correction algorithms. The proposed algorithm was tested on a set of 11600 complex document images with handwritten and typewritten text. In the images with handwritten text the algorithm yielded a 100% of extended hits percent. In the images with typewritten text the value of such percent was 99,71%. These results demonstrated that the proposed algorithm is able to effectively correct the skew of complex document images, giving good results. Moreover the results suggested that the algorithm is ready to be used in the software applications requiring it.

**Keywords:** analysis, correction, DIA, document, image, radon, skew, transform.

## TABLA DE CONTENIDOS

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1. MARCO TEÓRICO</b> .....	<b>7</b>
1.1. <i>Algoritmos para procesar imágenes digitales</i> .....	7
1.1.1. Transformadas Radon y Hough.....	7
1.1.2. Histograma .....	9
1.1.3. Proyecciones vertical y horizontal.....	9
1.1.4. Operaciones lógicas en imágenes binarias.....	9
1.1.5. Ajuste del brillo y del contraste .....	9
1.1.6. Segmentación basada en umbral .....	11
1.1.7. Detección de bordes.....	11
1.1.8. Dilatación y erosión .....	12
1.1.9. Análisis de las componentes conectadas .....	12
1.1.10. Rotación .....	13
1.2. <i>Algoritmos para corregir la inclinación en imágenes de documentos</i> .....	14
<b>CONCLUSIONES PARCIALES</b> .....	22
<b>CAPÍTULO 2. METODOLOGÍA Y ALGORITMO PROPUESTOS</b> .....	<b>24</b>
2.1. <i>Metodología para corregir la inclinación</i> .....	24
2.2. <i>Nuevo algoritmo para corregir la inclinación</i> .....	25
2.3. <i>Valoración preliminar</i> .....	27
<b>CONCLUSIONES PARCIALES</b> .....	28
<b>CAPÍTULO 3. RESULTADOS Y DISCUSIÓN</b> .....	<b>29</b>
3.1. <i>Implementación en MATLAB del algoritmo propuesto</i> .....	29
3.2. <i>Composición del banco de imágenes utilizado en las pruebas</i> .....	29
3.3. <i>Resultados obtenidos mediante el algoritmo propuesto</i> .....	31
3.4. <i>Discusión</i> .....	32
<b>CONCLUSIONES PARCIALES</b> .....	36
<b>CONCLUSIONES GENERALES</b> .....	<b>37</b>
<b>TRABAJO FUTURO</b> .....	<b>38</b>
<b>REFERENCIAS</b> .....	<b>39</b>
<b>BIBLIOGRAFÍA</b> .....	<b>41</b>

**ANEXOS** .....43

*Anexo 1. Resultados retornados por el algoritmo propuesto*.....43

*Anexo 2. Resultados retornados por otros algoritmos.* .....45

*Anexo 3. Evolución del algoritmo propuesto*.....54

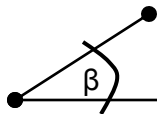
*Anexo 4. Codificación en MATLAB del algoritmo propuesto*.....61

*Anexo 5. Organigrama de algoritmos* .....66

**GLOSARIO DE TÉRMINOS**.....67

*Cuando algo se puede leer y entender sin mucho esfuerzo, es porque mucho esfuerzo ha sido puesto en su redacción.*



**Enrique Jardiel Poncela**

## **INTRODUCCIÓN**

En la actualidad existen dudas sobre si la utilización de las computadoras ha aumentado o disminuido la utilización de los documentos de papel. Hoy los documentos son creados y modificados principalmente de manera digital, pero se siguen imprimiendo millones de ellos para ser leídos o para servir como evidencia legal ante algún hecho, por sólo mencionar dos ejemplos. Además, el contenido en un documento de papel generalmente complementa al contenido en un documento digital, y viceversa.

La complementación existente entre ambos tipos de documentos ha provocado que el repetido lema de los 1980s, “la sociedad sin papel”, haya perdido relevancia para darle paso a un nuevo enfoque. Este enfoque consiste en lograr que ambos tipos de documentos coexistan de manera eficiente e integrada, lográndose sinergia entre los mismos. Para que ello sea posible se necesita que las computadoras sean capaces de leer el contenido escrito en papel, con la misma precisión que leen el contenido escrito de manera digital. En consecuencia, de una vez y por todas, se dispondría de un medio para plasmar la información que puede ser entendido simultáneamente por los humanos y por las computadoras: el papel.

El objetivo de la rama de la ciencia denominada “Análisis de imágenes de documentos” (DIA: Document Image Analysis) es “lograr que las computadoras sean capaces de entender el texto y los componentes gráficos existentes en imágenes digitales, y extraer la información de interés del mismo modo que un humano lo haría” [1].

El análisis de imágenes de documentos es aplicado en disímiles entornos. En la banca se analizan imágenes de documentos para reconocer automáticamente los números de cuenta, el monto de dinero, y los nombres escritos en cheques. En el correo postal se analizan imágenes de documentos para leer la dirección del destinatario, lo cual posibilita encaminar automáticamente las cartas y otros artefactos postales. Las bibliotecas digitales constituyen otro sector de aplicación, y es uno de los que tiene mayor impacto. En este sector se digitalizan los materiales físicos existentes en una biblioteca, se indexa su contenido, y se publican a la comunidad. La ventaja es que los usuarios no tienen que visitar la biblioteca para consultar un material. Basta con tener una conexión a Internet. Esto posibilita que una persona pueda consultar un material disponible en una biblioteca distante geográficamente. Otra ventaja es que los documentos digitales no se deterioran con el paso del tiempo ni con la manipulación; los documentos de papel, sí. Por tanto las bibliotecas digitales posibilitan que los usuarios puedan consultar documentos históricos creados hace cientos de años.

El análisis de imágenes de documentos también es aplicado en el procesamiento de formularios y en el procesamiento automático de encuestas. En el primero se reconocen e interpretan automáticamente los datos escritos en los campos de un formulario. En el segundo se reconoce el contenido plasmado en planillas de encuestas, y se emite automáticamente un resultado global. Existe otro sector que se está desarrollando aceleradamente en los últimos tiempos. Consiste en el reconocimiento e interpretación automática del texto manuscrito en dispositivos móviles. Todos los ejemplos mencionados anteriormente tienen como núcleo a otra aplicación del análisis de imágenes de documentos. Dicha aplicación es el reconocimiento óptico de caracteres (OCR).

Se plantea que la imagen de un documento está inclinada si las líneas de texto presentes en la misma forman un ángulo distinto de cero respecto a un eje de referencia, generalmente el eje horizontal. Esta inclinación se produce casi siempre durante el proceso de digitalización ya que es muy tedioso para una persona colocar la hoja totalmente alineada en la bandeja del escáner. El problema anterior se incrementa notablemente cuando una misma persona debe digitalizar cientos de documentos cada día, como parte de su carga laboral.

La corrección de la inclinación es una de las primeras tareas que se realizan durante el análisis de la imagen de un documento [1]. Incluso aunque no se vaya a analizar la imagen del documento, es

recomendable corregir la inclinación para facilitar la lectura del contenido de la imagen por parte de un observador humano. En caso de que sí se vaya a analizar la imagen, es preciso corregir la inclinación ya que la mayoría de los algoritmos a aplicar durante el análisis asumen como entrada una imagen sin inclinación. Algunos de estos algoritmos son la localización del texto en la imagen; la descomposición del texto en columnas, líneas, palabras, y símbolos; la representación y descripción de los símbolos, y el reconocimiento de los mismos [2]. Si la imagen que sirve como entrada a dichos algoritmos está inclinada, lo más probable es que se obtengan malos resultados.

Por tanto teniendo en cuenta que generalmente las imágenes de documentos están inclinadas debido a errores humanos durante el proceso de digitalización, y que la mayoría de los algoritmos utilizados en el análisis asumen que las mismas no están inclinadas, la corrección de la inclinación es un hito vital de cuyo resultado depende notablemente el éxito del proceso en general [1].

Los algoritmos que corrigen la inclinación en imágenes de documentos están organizados principalmente en seis categorías:

1. Algoritmos basados en el análisis de las Transformadas Hough o Radon [1] [3] [4] [5].
2. Algoritmos basados en el análisis de proyecciones [1] [2] [3].
3. Algoritmos basados en el análisis de las componentes conectadas [1] [2] [3] [6].
4. Algoritmos basados en la aproximación de funciones [7] [8].
5. Algoritmos basados en máscaras [3].
6. Algoritmos basados en el gradiente [9].

Los algoritmos desarrollados hasta el momento para corregir la inclinación asumen como entrada imágenes de documentos en los que el texto es escrito a máquina, está dispuesto en una columna, no se mezcla con objetos como figuras y gráficos, y tiene relativamente el mismo tipo y tamaño de letra en todos sus símbolos.

Lamentablemente para estos algoritmos el formato de la mayoría de los documentos que hoy proliferan no es tan sencillo. Hoy existen potentes procesadores de texto que permiten organizar el texto en varias columnas; insertar gráficos, imágenes, tablas y ecuaciones junto al texto; controlar el interlineado, la sangría, el tipo de letra y el tamaño de letra, de una sección de texto determinada; establecer bordes de página e incluir decoraciones junto al texto; por sólo mencionar algunas funcionalidades. A lo anterior súmese la existencia de imágenes de documentos con texto manuscrito. Este tipo de imágenes aporta nuevas dificultades para los algoritmos existentes.

La principal debilidad de los algoritmos propuestos hasta el momento es que procesan toda la imagen para estimar la inclinación presente en la misma. Este procesamiento de toda la imagen es el principal factor que los incapacita de ser robustos ante las complejidades ya mencionadas. Se plantea esto ya que para estimar la inclinación de una imagen es recomendable tener en cuenta solamente la inclinación de las líneas de texto. Objetos como las figuras, los gráficos, las decoraciones y los bordes de página, no son significativos para estimar la inclinación de la imagen. Por el contrario, estos objetos son catalogados como ruidosos ya que pueden desviar a la inclinación estimada de la inclinación real. De la debilidad anterior se deriva un problema relacionado con el tiempo de cómputo. Los algoritmos existentes no tienen en cuenta el hecho de que generalmente todo el texto en una imagen tiene la misma inclinación. Por tanto no es necesario procesar todas las regiones de texto. Basta con procesar las regiones de texto que mejor representen la inclinación de la imagen. Esto último disminuiría notablemente el tiempo de cómputo requerido para corregir la inclinación en cada imagen.

En todo lo mencionado anteriormente se manifiesta una clara contradicción entre la teoría y la práctica. Por un lado (la práctica) se crean documentos con varias complejidades, empleando modernos procesadores de texto. Estos documentos generan imágenes que seguramente deberán ser analizadas por algunas de las aplicaciones ya mencionadas, u otras. Por otro lado (la teoría) la corrección de la inclinación es un hito vital durante el análisis de imágenes de documentos y los algoritmos existentes para corregir la inclinación no tienen en cuenta las complejidades planteadas, arrojando malos resultados ante

las mismas (véase Anexo 2). Esta contradicción provoca un impacto negativo en la práctica cotidiana.

En lo que sigue asúmase que:

- Una “imagen de documento complejo” es aquella en la que el texto puede ser manuscrito o escrito a máquina; mostrar distintos tipos de letra, tamaños de letra, e interlineados; y estar dispuesto en una o varias columnas. Además es aquella que contiene figuras, gráficos, tablas, ecuaciones y decoraciones, mezclados junto al texto. También puede contener bordes de página y bordes laterales (Fig 1).
- Cuando se utilice la palabra “algoritmo”, sin indicar su tipo, se refiere a un algoritmo para corregir la inclinación en imágenes de documentos.
- Un “resultado satisfactorio” significa que un algoritmo es capaz de brindar un por ciento extendido de aciertos superior al 95%, al procesar un conjunto de imágenes. Este por ciento extendido se calcula multiplicando por 100 la cantidad de imágenes en las que la diferencia absoluta entre la inclinación real y la inclinación estimada por el algoritmo fue menor o igual que un grado, y dividiendo el resultado del producto entre el total de imágenes procesadas.

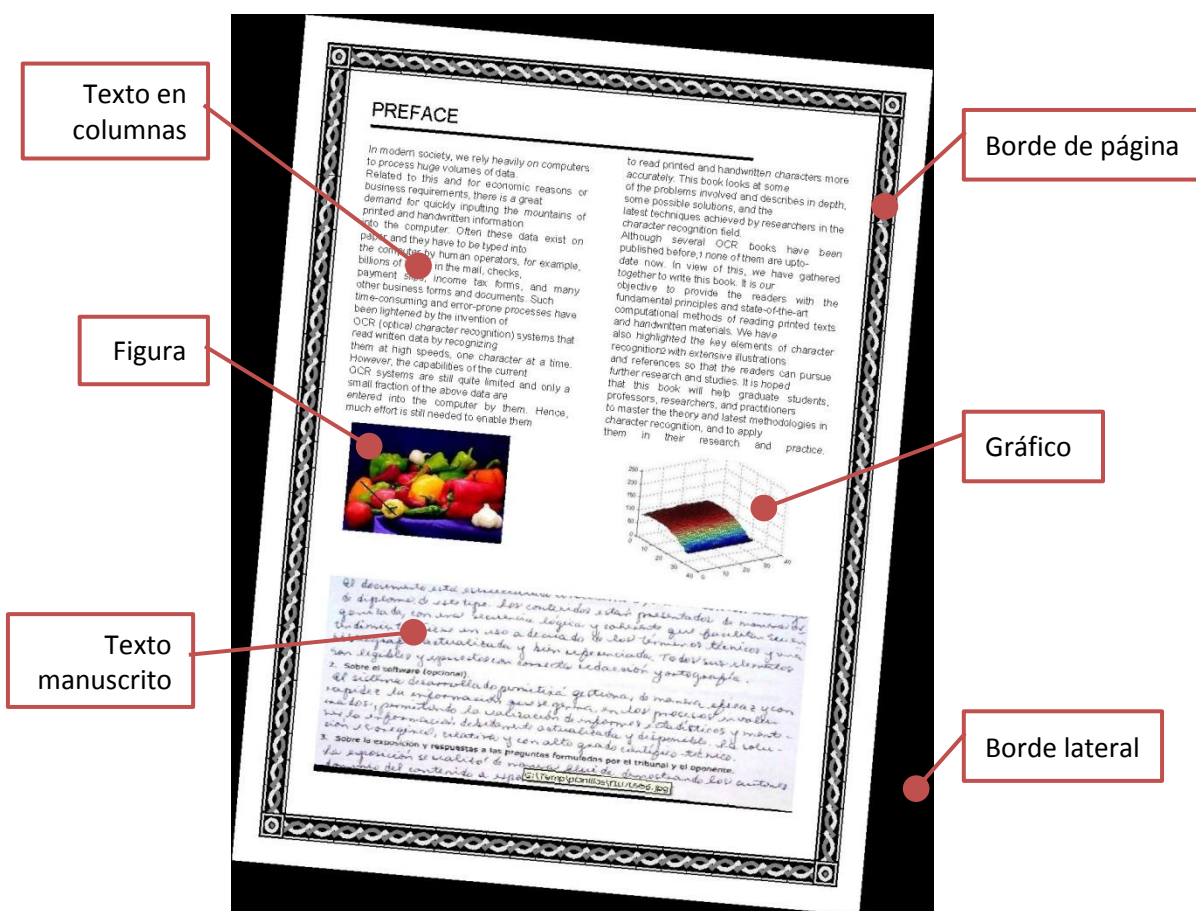


Figura 1. Imagen de un documento complejo, inclinada  $-5^{\circ}$ .

El problema científico que originó esta investigación fue “los algoritmos desarrollados hasta la actualidad no tienen en cuenta las características de las imágenes de documentos complejos, lo cual los incapacita de brindar resultados satisfactorios al corregir la inclinación en este tipo de imágenes”. El objeto que se estudió fue “la corrección de la inclinación en imágenes de documentos” y se planteó como objetivo general “diseñar un algoritmo que brinde resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos”. Este objetivo general delimitó al campo de acción “selección y análisis de las regiones significativas para estimar la inclinación en imágenes de documentos”, dentro del objeto de estudio de la investigación.

La hipótesis formulada como posible solución al problema, y que sirvió como guía durante la investigación, fue: “un algoritmo que elimine de la imagen del documento las regiones con mayor acumulación de puntos de objeto, y que estime la inclinación analizando localmente la Transformada Radon de las regiones más significativas, brinda resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos”.

Los objetivos específicos de la investigación fueron:

1. Elaborar el marco teórico de la investigación, evidenciando una revisión crítica de los algoritmos existentes.
2. Diseñar un algoritmo que brinde resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos.
3. Ejecutar algoritmos precedentes (y el algoritmo diseñado) en un banco de imágenes de prueba, recopilar los resultados obtenidos e interpretarlos.

Para lograr el objetivo general y los objetivos específicos anteriormente planteados, durante la investigación se ejecutaron las siguientes tareas:

1. Elaboración de un banco de imágenes de documentos, necesario para hacer las pruebas.
2. Revisión crítica de los algoritmos existentes.
3. Implementación en MATLAB y prueba de algoritmos existentes.
4. Diseño de un nuevo algoritmo, que elimine de la imagen del documento las regiones con mayor acumulación de puntos de objeto y que estime la inclinación analizando localmente la Transformada Radon de las regiones más significativas.
5. Implementación en MATLAB y prueba del algoritmo diseñado.
6. Comparación e interpretación de los resultados generados por algoritmos precedentes y por el algoritmo diseñado.

La unidad de estudio de la investigación fue “el algoritmo para corregir la inclinación en imágenes de documentos”. En cada unidad de estudio las variables “elimina las regiones con mayor acumulación de puntos de objeto”, “estima la inclinación analizando localmente la Transformada Radon de las regiones más significativas”, y “genera resultados satisfactorios”, tomaron distintos valores. Al respecto de estas tres variables, todas son dicotómicas ya que toman uno de dos valores posibles (Sí, No). Además las dos primeras son independientes y la tercera es dependiente.

La técnica que se utilizó para seleccionar de la población (integrada por todas las unidades de estudio posibles) una muestra representativa fue “el muestreo accidental”. El uso de esta técnica no probabilística se evidenció al revisar arbitrariamente en Internet los algoritmos propuestos por otros autores. Las principales fuentes de información que se utilizaron fueron artículos de revista, memorias de eventos, y libros, todas relacionadas con la corrección de la inclinación y con el análisis de imágenes de documentos. Se tuvo que utilizar una técnica de muestreo no probabilística ya que no se pudo lograr el principio de equiprobabilidad de todas las unidades de estudio en la población, para integrar la muestra. Sobre la estrategia de investigación que se utilizó, fue explicativa o experimental. Se plantea esto último ya que se estudiaron las causas que incapacitan a los algoritmos existentes de brindar resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos.



Los métodos científicos se dividen en teóricos y empíricos. Los métodos teóricos utilizados durante la investigación fueron el hipotético-deductivo, el sistémico, el dialéctico, y el histórico. El método hipotético-deductivo se utilizó durante la verificación de la hipótesis. El método sistémico se empleó para estudiar los hitos básicos que intervienen en la corrección de la inclinación. El método dialéctico fue útil para estudiar las contradicciones existentes. Respecto al método histórico, se utilizó para analizar la evolución del análisis de imágenes de documentos y de la corrección de la inclinación.

Como métodos empíricos se emplearon la observación, la medición, y el experimento. El primero se utilizó al observar el comportamiento de los algoritmos ante imágenes con distintas complejidades. El segundo para medir el valor de las variables en las unidades de estudio. Respecto a los experimentos, se hicieron para conocer el comportamiento de algoritmos existentes, y del propuesto, ante las imágenes en el banco de prueba.

La investigación descrita en este documento es novedosa en cuanto su resultado permite obtener resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos, lo cual no era posible con los algoritmos precedentes. Además el problema fue abordado desde un punto de vista diferente al propuesto por otros autores. Hasta el momento los autores han concentrado sus esfuerzos en proponer nuevas técnicas para estimar la inclinación, y en mejorar las existentes. Es por ello que en la actualidad las técnicas para estimar la inclinación están bien establecidas y fundamentadas, cada una con sus ventajas y desventajas inherentes.

En esta investigación se puso el énfasis en mejorar notablemente la calidad de los datos que sirven como entrada a una técnica de estimación determinada. La técnica de estimación que se eligió fue el análisis de la Transformada Radon. El nuevo algoritmo fue diseñado de modo tal que se aprovecharan las ventajas de la estimación basada en el análisis de la Transformada Radon, y a la vez se disminuyeran notablemente las desventajas de este tipo de estimación. En el diseño del algoritmo se evidencian aspectos novedosos que posibilitaron lograr la relación mencionada anteriormente.

La investigación abordada en esta tesis hace dos aportes teóricos. El primero es una metodología que describe las operaciones a realizar en cualquier algoritmo que corrija la inclinación en imágenes de documentos. Dicha metodología es útil en cuanto sirve como guía para diseñar nuevos algoritmos, lo cual fue verificado durante la concepción del algoritmo propuesto en esta investigación. El segundo aporte teórico es el nuevo algoritmo que se propone. Dicho algoritmo es novedoso en cuanto al método de preprocesamiento que se utiliza para eliminar los objetos ruidosos, en cuanto al método que se utiliza para seleccionar las regiones significativas, y en cuanto al preprocesamiento adicional basado en operaciones morfológicas que se aplica a cada región significativa antes de calcular y analizar la Transformada Radon.

El aporte práctico de la investigación radica en un componente implementado en MATLAB, que expone el algoritmo propuesto en esta tesis. Esto posibilita que el resultado teórico de esta investigación pueda ser introducido inmediatamente en la práctica cotidiana. Nótese que utilizando las capacidades de exportación de código brindadas por MATLAB, el resultado teórico de esta investigación también puede ser utilizado desde aplicaciones desarrolladas en Java, .NET y C/C++ por sólo citar tres ejemplos. Por tanto esta investigación contribuye a desarrollar la industria del software en Cuba, en cuanto el problema de la corrección de la inclinación está presente en disímiles aplicaciones y los algoritmos precedentes padecen de los problemas ya planteados.

Lo que resta de este documento está organizado cómo sigue. En el Capítulo 1 se describen algunos algoritmos básicos para procesar imágenes digitales. Además se analizan algoritmos que existen para corregir la inclinación en imágenes de documentos. Por cada algoritmo analizado se emite un juicio crítico respecto a sus fortalezas y debilidades. En el capítulo además se caracterizan brevemente las tendencias identificadas y se plantean las debilidades comunes de los algoritmos estudiados.

En el Capítulo 2 se describe la metodología propuesta en esta investigación. Dicha metodología tiene un enfoque integral, abarcando desde la digitalización de la imagen del documento hasta la obtención de la imagen libre de inclinación. También se describe el algoritmo propuesto. Dicho algoritmo implementa cada paso de la metodología y está diseñado de modo tal que brinde resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos.

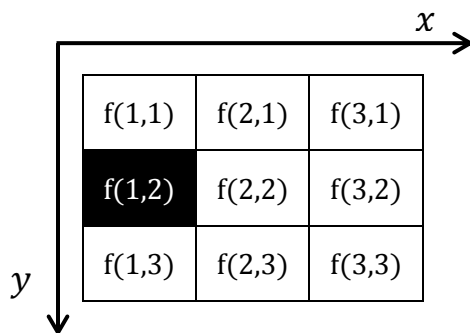
En el Capítulo 3 se describen los archivos de código fuente obtenidos al implementar en MATLAB el algoritmo propuesto. También se describe la composición del banco de imágenes utilizado en las pruebas, el cual se conforma de 11600 imágenes. En el capítulo se muestran los resultados arrojados por el nuevo algoritmo y se demuestra su superioridad respecto a algoritmos precedentes. Téngase en cuenta que el nuevo algoritmo arrojó un por ciento extendido de aciertos de 99,71% para el caso de las imágenes con texto escrito a máquina, y del 100% para las imágenes con texto manuscrito.

Luego se plantean las conclusiones generales de la investigación y se presentan las tareas que requerirán esfuerzos en el futuro para darle continuidad a la misma. A las secciones anteriores siguen las referencias bibliográficas y la bibliografía.

El Anexo 1 contiene tablas que muestran detalladamente los resultados generados por el algoritmo propuesto al procesar las imágenes en el banco de prueba. Los resultados generados por seis algoritmos precedentes, representativos de las categorías de algoritmos que existen para corregir la inclinación, son expuestos en el Anexo 2. En el Anexo 3 se demuestra la evolución del algoritmo propuesto, mediante la utilización de una imagen de prueba. En el Anexo 4 se presenta el código fuente obtenido al implementar en MATLAB el algoritmo propuesto en la investigación. En el Anexo 5 se muestra un organigrama que refleja jerárquicamente las categorías de los algoritmos que corrigen la inclinación. Al final del documento se muestra un glosario de términos. Dicho glosario contiene el significado atribuido a términos que se utilizan repetidamente a lo largo del documento.

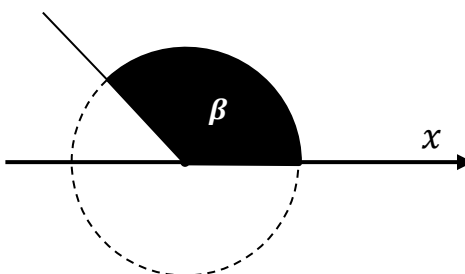
**Al leer el cuerpo de este documento téngase en cuenta que:**

- En el sistema de ejes de coordenadas utilizado para indicar la posición de los elementos de una imagen, el eje  $x$  (horizontal) crece de izquierda a derecha y el eje  $y$  (vertical) crece desde arriba hacia abajo. La posición de un elemento en la imagen está dada por el par  $(x, y)$ . Por ejemplo la notación  $f(1,2)$  se refiere al pixel de la imagen  $f$  que está en la intersección de la primera columna con la segunda fila (Fig 2).

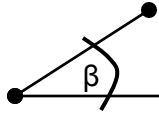


**Figura 2.** Sistema de eje de coordenadas.

- Los ángulos son expresados en grados y se miden en sentido contrario a las manecillas del reloj, a partir del eje  $x$  (Fig 3).



**Figura 3.** Convención adoptada para medir los ángulos.



## CAPÍTULO 1. MARCO TEÓRICO

En este capítulo se describen algunos algoritmos básicos para procesar imágenes digitales. Además se analizan varios algoritmos para corregir la inclinación en imágenes de documentos. Por cada algoritmo analizado, se emite un juicio crítico respecto a sus fortalezas y debilidades. En el capítulo se caracterizan brevemente las tendencias identificadas y se plantean las debilidades comunes de los algoritmos estudiados.

### 1.1. Algoritmos para procesar imágenes digitales

Una imagen digital es una función bidimensional  $f(x, y)$  donde  $x$ ,  $y$ , y  $f(x, y)$  son cantidades finitas. Se llama al valor de  $f$  en el punto con coordenadas  $(x, y)$ , la intensidad de la imagen en dicho punto [10].

Las imágenes digitales se clasifican en cuatro clases de acuerdo al tipo de sus elementos o píxeles. Las imágenes “binarias” son aquellas en las que los píxeles almacenan un valor, de dos posibles. Las imágenes “en escala de grises” son aquellas en las que los píxeles almacenan un valor numérico (escalar), perteneciente a un conjunto acotado. Los píxeles de una imagen en “colores” almacenan un vector generalmente conformado por tres componentes escalares. El significado de cada componente depende del modelo de color utilizado. Algunos modelos de color son RGB, CMY y CIELAB [10]. Las imágenes “indexadas” almacenan en sus píxeles un valor numérico que se utiliza como índice. Dicho índice se emplea para recuperar el color del píxel desde un mapa o paleta de colores [11].

El término “Procesamiento de imágenes digitales” denota al campo cuyo propósito es procesar imágenes digitales mediante la utilización de computadoras. Este campo comprende los procesos que reciben una imagen y retornan otra imagen, reciben una imagen y extraen rasgos a partir de la misma, y clasifican los objetos presentes en una imagen [10].

En esta sección se describen algunos algoritmos básicos para procesar imágenes digitales. Dichos algoritmos son utilizados intensivamente en los algoritmos que corrigen la inclinación en imágenes de documentos.

#### 1.1.1. Transformadas Radon y Hough

Cualquier línea recta puede ser descrita mediante la ecuación normal (Ec 1.1). En dicha ecuación el parámetro  $\rho$  es la longitud del menor segmento comprendido desde el origen hasta la recta, y  $\theta$  indica la inclinación del segmento.

$$\rho = x \cos \theta + y \sin \theta \quad (1.1)$$

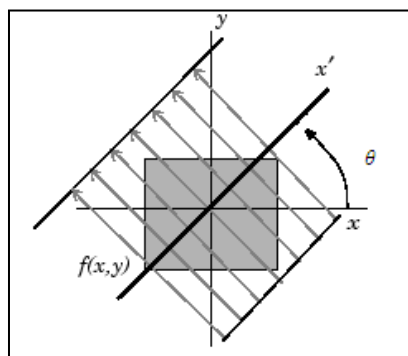
La Transformada Radon  $\overline{g(\rho, \theta)}$  de una señal bidimensional continua  $g(x, y)$  es el resultado de integrar los valores de  $g$  a lo largo de líneas rectas inclinadas [12]. Fue introducida en el año 1917 por el matemático austriaco Johann Radon. En la ecuación 1.2 se muestra la formulación matemática de esta transformada. Nótese que todas las rectas pueden ser descritas seleccionando  $0^0 \leq \theta < 180^0$  y  $-\rho_{\max} \leq \rho \leq \rho_{\max}$ .

$$\overline{g(\rho, \theta)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \delta(\rho - x \cos \theta - y \sin \theta) dx dy \quad (1.2)$$

donde  $\delta(k)$  es la delta de Dirac.

En el procesamiento de imágenes digitales, el caso discreto de la Transformada Radon (RT) es la proyección de los valores de la imagen a lo largo de rectas inclinadas a un ángulo específico (Fig 1.1). Generalmente se representa digitalmente mediante una matriz bidimensional. Las columnas de la matriz representan los valores de  $\theta$  y las filas representan los valores de  $\rho$ .

Una aplicación de la RT es la detección de las líneas más visibles en la imagen. Esto puede lograrse calculando la RT asociada a la imagen y buscando los valores máximos en la matriz de la RT. Las coordenadas  $(\rho, \theta)$  que indican la posición de los máximos encontrados describen a las líneas rectas más visibles (véase ecuación 1.1). En [12] se aborda de manera exquisita la teoría y la implementación de la RT.



**Figura 1.1.** La Transformada Radon de una imagen digital. La imagen es proyectada a lo largo de una línea  $x'$  inclinada  $\theta$  grados.

La Transformada Hough (HT), del modo que es conocida actualmente, fue propuesta en el año 1972 por Richard O. Duda y Peter E. Hart [13]. Esta transformada permite detectar líneas rectas en imágenes digitales, de modo equivalente a la RT. Para este caso (detección de líneas), ambas transformadas explotan la ecuación normal de la recta. También utilizan un arreglo de votos para almacenar la cantidad de puntos de la imagen que están alineados a lo largo de una recta caracterizada por el par  $(\rho, \theta)$ .

La ventaja de la HT es que ha sido extendida para localizar otras formas en la imagen, como círculos y elipses. Esta extensión se logra ajustando el arreglo de votos para que la cantidad de dimensiones del mismo sea igual a la cantidad de parámetros de la forma buscada. Por ejemplo para encontrar un círculo (Ec 1.3) en la imagen, el arreglo de votos debe tener tres dimensiones. Una para los posibles valores de  $a$ , otra para los posibles valores de  $b$ , y otra para los posibles valores del  $r$ . El problema en utilizar la HT para detectar formas de orden superior (e.g. círculos, elipses) radica en que se incrementa notablemente la complejidad computacional del algoritmo. Es por ello que el uso más común de la HT es para encontrar líneas rectas en la imagen, al igual que la RT.

$$(x - a)^2 + (y - b)^2 = r^2 \tag{1.3}$$

donde  $(a, b)$  es el centro de círculo y  $r$  es el radio del mismo.

La HT es abordada en varios libros de procesamiento de imágenes ya que es un algoritmo clásico y muy utilizado. Se destacan las fuentes [10], [14] y [15] por su rigor y claridad. Se sugiere la lectura de [16] para conocer la historia de la transformada, contada por uno de sus autores. Dicha lectura resulta muy interesante porque desde 1962, cuando P.V.C Hough presentó su patente [17], la evolución de la HT ha estado marcada por varios hitos.

Tanto la RT como la HT, son calculadas a partir de imágenes binarias y en escala de grises. En el caso de las imágenes binarias, el valor FALSE se asume como cero y el valor TRUE se asume como uno.

### 1.1.2. Histograma

El histograma es un rasgo que se calcula generalmente a partir de imágenes binarias y en escala de grises. En el caso de las imágenes binarias, el histograma es un vector de dos elementos. El primer elemento del vector almacena la cantidad de ocurrencias de un valor binario en la imagen. El segundo elemento almacena la cantidad de ocurrencias del valor binario restante.

Para el caso de las imágenes en escala de grises, el histograma es un vector dado por la función  $h(i) = L_i$  con  $(i = 0, \dots, N)$ , donde  $L_i$  es la cantidad de ocurrencias del nivel de intensidad  $i$  en la imagen, y  $N$  es el máximo valor escalar que puede tomar un pixel.

### 1.1.3. Proyecciones vertical y horizontal

La proyección vertical y la proyección horizontal son rasgos que se obtienen al proyectar los valores de la imagen en el eje horizontal ( $x$ ) o en el vertical ( $y$ ), respectivamente. Ambas proyecciones se calculan generalmente a partir de imágenes binarias y en escala de grises. Para el caso de las imágenes binarias, el valor FALSE se asume como cero y el valor TRUE se asume como uno. En la ecuación 1.4 se muestra la formulación de la proyección vertical y en la ecuación 1.5 se muestra la de la proyección horizontal

$$P_V(x) = \sum_{y=1}^{\text{alto}} f(x, y) \quad (1.4)$$

donde  $x = 1, \dots, \text{ancho}$ .

$$P_H(y) = \sum_{x=1}^{\text{ancho}} f(x, y) \quad (1.5)$$

donde  $y = 1, \dots, \text{alto}$ .

Debe decirse que además de las proyecciones vertical y horizontal, existe otra en la cual la imagen es proyectada diagonalmente. Esta proyección “diagonal” es utilizada raramente.

### 1.1.4. Operaciones lógicas en imágenes binarias

Las operaciones lógicas elementales AND, OR, NOT también son utilizadas en el procesamiento de imágenes digitales. Sean  $f$  y  $g$  dos imágenes binarias de tamaño  $M \times N$ :

- La operación  $f$  AND  $g$  retorna una nueva imagen binaria  $w$  de igual tamaño, cuyos pixeles son TRUE si y sólo si los pixeles correspondientes en  $f$  y  $g$  son ambos TRUE.
- La operación  $f$  OR  $g$  retorna una nueva imagen binaria  $w$  de igual tamaño, cuyos pixeles son TRUE si y sólo si al menos uno de los pixeles correspondientes en  $f$  y  $g$  es TRUE.

Respecto a la operación NOT, retorna una nueva imagen binaria donde el valor de cada pixel es el complemento del valor binario almacenado en el pixel correspondiente de la imagen de entrada. En [10] y en [18] se aborda de manera detallada la aplicación de operaciones lógicas en imágenes digitales.

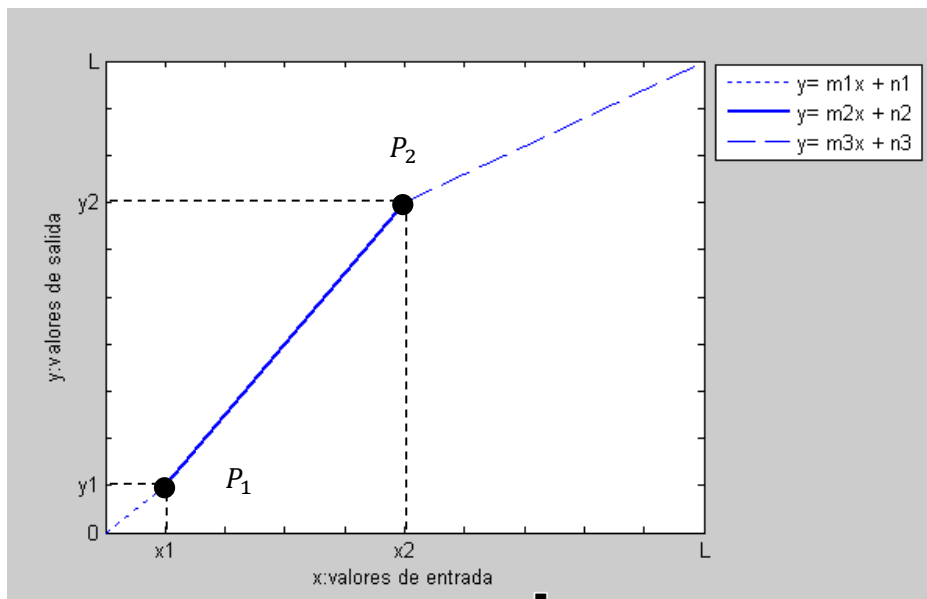
### 1.1.5. Ajuste del brillo y del contraste

El ajuste del brillo y el ajuste del contraste son algoritmos de preprocesamiento de imágenes digitales. Específicamente, dentro de esta categoría (preprocesamiento), son algoritmos de mejora basados en el procesamiento puntual [10].

El brillo controla cuán blanca, o cuán oscura, es la imagen. Mientras mayor sea el valor de los pixeles, mayor es el brillo y más blanca es la imagen; mientras menor sea el valor de los pixeles, menor es el brillo y más oscura es la imagen. El contraste controla la distribución de los valores de la imagen a lo largo del espectro de posibles valores. Mientras mayor sea el contraste, más distribuidos están los valores de la imagen a lo largo de espectro, y por tanto mejor se refleja la diferencia entre los pixeles. En una imagen

con bajo contraste sus valores están distribuidos en un subconjunto reducido del espectro, y por tanto es difícil notar la diferencia entre sus píxeles.

En la figura 1.2 se modela el ajuste del contraste. En dicha figura el eje  $x$  está asociado a los valores de entrada y el eje  $y$  está asociado a los valores de salida.  $L$  es el máximo valor del espectro ( $L \geq 0$ ). Nótese en dicha figura que el efecto del ajuste del contraste está dado por la ubicación de los puntos  $P_1(x_1, y_1)$  y  $P_2(x_2, y_2)$ . La interpretación de los puntos es la siguiente: los valores en el rango  $[x_1, x_2]$  de la imagen de entrada, serán mapeados a valores en el rango  $[y_1, y_2]$  de la imagen de salida.



**Figura 1.2.** Modelación gráfica del ajuste del contraste.

La ecuación 1.6 muestra la formulación matemática del ajuste del brillo y la ecuación 1.7 muestra la del ajuste de contraste. Véase que la ecuación 1.6 es un caso especial de la ecuación 1.7, si  $m_1 = m_2 = m_3 = 1$  y  $n_1 = n_2 = n_3 = K$ .

$$\begin{aligned}
 & y = x + K \tag{1.6} \\
 & \text{donde } K \text{ controla el efecto de ajustar el brillo. Si } K > 0 \text{ se aumenta el brillo. Si } K < 0 \text{ se disminuye el brillo.}
 \end{aligned}
 \left| \right.
 \begin{aligned}
 y = \begin{cases} m_1x + n_1 & \text{si } x \in [0, x_1] \\ m_2x + n_2 & \text{si } x \in [x_1, x_2] \\ m_3x + n_3 & \text{si } x \in [x_2, L] \end{cases} \tag{1.7}
 \end{aligned}$$

En el caso del ajuste del brillo debe tenerse en cuenta que se pueden generar valores fuera de rango. La estrategia a seguir para manejar esta situación consiste en establecer dichos valores, al mínimo o al máximo del espectro, según corresponda.

El ajuste del brillo y el ajuste del contraste son aplicados generalmente a imágenes en escala de grises. Con algunas modificaciones, pueden aplicarse a imágenes en colores. Se puede obtener información detallada sobre estos algoritmos en los libros [10] y [18].

### 1.1.6. Segmentación basada en umbral

La segmentación es una operación que divide la imagen en regiones no solapadas, por lo que cada pixel de la imagen segmentada pertenece a una y sólo a una región [10]. La calidad de la segmentación es muy importante ya que de ella depende el éxito en las etapas posteriores de un sistema de procesamiento de imágenes digitales.

La segmentación basada en umbral (thresholding) es un tipo de segmentación en el cual se utiliza un umbral ( $T$ ) para agrupar los valores de la imagen en dos clases. Los pixeles cuyo valor sea menor que  $T$  constituyen una región o clase. Los pixeles restantes constituyen la otra clase. Por tanto, el resultado de la segmentación basada en umbral es una imagen binaria. Las imágenes de entrada pueden ser en colores y en escala de grises, aunque generalmente este tipo de segmentación se utiliza con más frecuencia en las últimas.

La ventaja de la segmentación basada en umbral radica en su sencillez, su facilidad de implementación, y su baja complejidad computacional. Sin embargo, no es una panacea. Encontrar el umbral que segmente adecuadamente la imagen en dos clases, puede ser muy difícil. Es esta la principal desventaja de este tipo de segmentación: la dependencia total del umbral. Para encontrar dicho umbral se han propuesto varios métodos. Uno de los más conocidos es el método de Otsu [19]. El umbral retornado por este método es aquel para el cual se minimiza la diferencia entre los pixeles de una misma clase (intra-class-variance) y se maximiza la diferencia entre los pixeles de clases distintas (inter-class-variance). La fuente [20] es un excelente artículo en el cual se revisan y comparan 40 métodos para estimar el umbral. Se sugiere su lectura. También se sugiere la lectura de las secciones relacionadas con la segmentación de imágenes digitales en los libros [10], [18] y [21].

### 1.1.7. Detección de bordes

Un borde es el conjunto de puntos conectados mediante un criterio dado, en los cuales el valor de la primera derivada bidimensional es mayor que un umbral  $T$ . Desde otro punto de vista es el conjunto de pixeles conectados, en los cuales el valor de la segunda derivada bidimensional es cero. El objetivo de la rama del procesamiento de imágenes digitales conocida como “detección de bordes”, es encontrar los bordes existentes en una imagen digital [10].

En la detección de bordes se utiliza el vector gradiente para calcular la primera derivada, y el laplaciano para calcular la segunda derivada. En el caso del vector gradiente su magnitud y dirección en un punto determinado, son calculadas mediante las ecuaciones 1.8 y 1.9 respectivamente. La dirección del vector gradiente en un punto  $(x, y)$  tiene una propiedad muy importante. Consiste en que la dirección del borde que pasa por el punto es perpendicular a la dirección del vector gradiente en dicho punto [10]. Esta simple relación es aprovechada en muchísimas aplicaciones.

$$\nabla f = \sqrt{(G_x^2 + G_y^2)} \quad (1.8)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad (1.9)$$

**Nota:** El ángulo es medido respecto al eje  $x$ .

Donde  $G_x$  es la derivada parcial respecto al eje horizontal, y  $G_y$  es la derivada parcial respecto al eje vertical. Recuérdese que ambas derivadas parciales son calculadas en cada punto de la imagen.

Existen varias vías para hallar las derivadas parciales involucradas en las ecuaciones 1.8 y 1.9. La más sencilla es el operador de Roberts. Otras son el operador de Prewitt y el operador de Sobel [10]. También existen varias vías para calcular el valor del umbral  $T$ . Una de ellas, la utilizada por la función `edge` de MATLAB, es descrita en la sección 16.2 del libro [14].

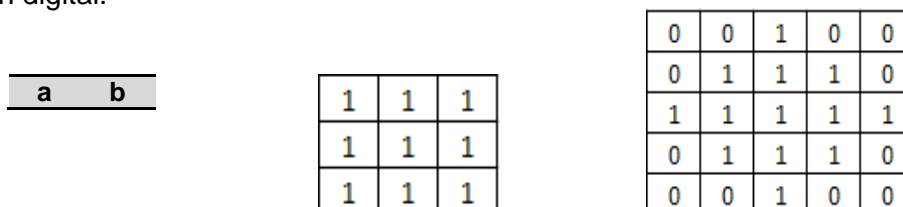
La detección de bordes puede aplicarse a imágenes en colores y en escala de grises, aunque lo más común es que se aplique a las últimas. El resultado que se produce es una imagen binaria en la que un punto es TRUE si pertenece a algún borde. Esta rama (básica) del procesamiento de imágenes es

abordada en varios libros. Por su alto rigor y claridad se recomienda [10]. Otras fuentes muy interesantes son [14] y [21].

### 1.1.8. Dilatación y erosión

El término “Morfología”, dentro de procesamiento de imágenes digitales, aglutina a las operaciones que procesan imágenes digitales basándose en las formas (e.g. cuadrados, círculos).

Las operaciones morfológicas le aplican un elemento de estructuración (SE: structuring element) a la imagen de entrada para producir una imagen de salida. El tamaño de ambas imágenes es el mismo. El valor de cada pixel en la imagen de salida se obtiene analizando la vecindad<sup>1</sup> del pixel correspondiente en la imagen de entrada. El SE controla la forma y el tamaño de la vecindad (Fig 1.3) mientras que el criterio empleado en el análisis determina el tipo de operación morfológica que se está aplicando. Se aclara que la forma del SE determina que la operación morfológica será sensible ante los objetos con dicha forma en la imagen digital.



**Figura 1.3.** Ejemplos de elementos de estructuración (los puntos que componen el SE se etiquetan con “1” y los demás con “0”). a) Cuadrado de lado 3; b) Diamante de radio 2.

La dilatación y la erosión son las dos operaciones morfológicas elementales. En la Tabla 1.1 se muestra el funcionamiento de las mismas. Nótese que en el caso de la dilatación se adicionan pixeles a la frontera de los objetos presentes en la imagen, por lo que aumenta el tamaño de los objetos. En el caso de la erosión se eliminan puntos de la frontera, por lo que disminuye el tamaño de los objetos. La cantidad de puntos adicionados o eliminados depende de la forma y el tamaño del SE.

**Tabla 1.1.** Funcionamiento de la dilatación y la erosión.

<b>Dilatación</b>	El pixel de salida es el valor máximo de la vecindad correspondiente en la imagen de entrada. En el caso de las imágenes binarias, el resultado es TRUE si algún pixel en la vecindad es TRUE.
<b>Erosión</b>	El pixel de salida es el valor mínimo de la vecindad correspondiente en la imagen de entrada. En el caso de las imágenes binarias, el resultado es FALSE si algún pixel en la vecindad es FALSE.

La dilatación y la erosión pueden aplicarse a imágenes binarias, en escala de grises, y en colores. Sin embargo, lo más común es que se apliquen a imágenes binarias. En [10] se analizan estas operaciones morfológicas con alto rigor matemático. También se abordan otras operaciones como la clausura y la apertura. Otra fuente muy interesante es [22]. Se caracteriza por su claridad y sentido práctico.

### 1.1.9. Análisis de las componentes conectadas

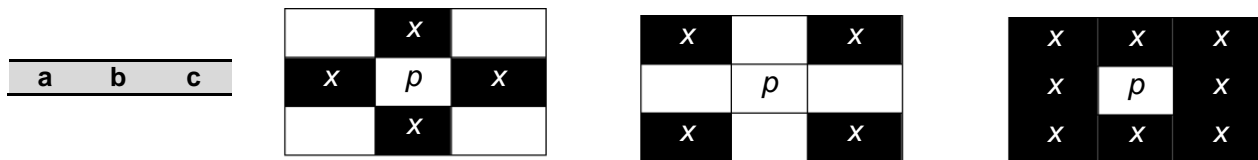
Alrededor de un pixel  $p$  ubicado en las coordenadas  $(x, y)$ , existen principalmente tres tipos de vecindad. La vecindad  $N_4(p)$  agrupa a los pixeles en las coordenadas  $(x, y - 1)$ ,  $(x + 1, y)$ ,  $(x, y + 1)$  y  $(x - 1, y)$ . La vecindad  $N_D(p)$  agrupa a los pixeles en las coordenadas  $(x - 1, y - 1)$ ,  $(x + 1, y - 1)$ ,

<sup>1</sup> Se asume que la vecindad de un pixel lo incluye a sí mismo (caso en el cual la distancia entre el pixel y el vecino es cero).



$(x + 1, y + 1)$  y  $(x - 1, y + 1)$ . La vecindad  $N_8(p)$  constituye la unión de las vecindades  $N_4(p)$  y  $N_D(p)$ . En la figura 1.4 se muestran gráficamente las vecindades anteriores.

Dos pixeles  $p$  y  $q$  son 4-adyacentes si  $q \in N_4(p)$ . Similarmente, los pixeles  $p$  y  $q$  son 8-adyacentes si  $q \in N_8(p)$ . Un camino entre los pixeles  $p_1$  y  $p_n$  es la secuencia de pixeles  $p_1, p_2, \dots, p_{n-1}, p_n$  de modo tal que  $p_k$  es adyacente a  $p_{k+1}$ , para todos los valores de  $k$  ( $1 \leq k < n$ ). Nótese que el camino está determinado por el criterio de vecindad empleado (e.g.  $N_4, N_8$ ).



**Figura 1.4.** Principales tipos de vecindad de un píxel  $p$  en las coordenadas  $(x, y)$ . a)  $N_4(p)$  agrupa a los dos vecinos verticales y a los dos horizontales; b)  $N_D(p)$  agrupa a los cuatro vecinos diagonales; c)  $N_8(p)$  es la unión de las vecindades  $N_4(p)$  y  $N_D(p)$ .

Dos pixeles de objeto  $p$  y  $q$  están 4-conectados si existe un camino que los une, basado en la vecindad  $N_4$  y compuesto enteramente por puntos de objeto. De modo similar, se dice que los puntos de objeto  $p$  y  $q$  están 8-conectados si existe un camino que los une. Dicho camino debe estar basado en la vecindad  $N_8$  y compuesto enteramente por puntos de objeto.

Habiéndose planteado todo lo anterior, para cualquier punto de objeto  $p$  en una imagen digital, el conjunto compuesto por todos los puntos de objeto conectados a  $p$  se define como la “componente conectada que contiene a  $p$ ” [23]. Definida de manera diferente, pero con igual significado, una componente conectada basada en un criterio de vecindad dado es el conjunto de puntos de objeto donde para cualquier par de puntos  $p$  y  $q$  ( $p \neq q$ ) en el conjunto, existe un camino que los une basado en el criterio de vecindad utilizado.

El análisis de las componentes conectadas (CCA: Connected Component Analysis) agrupa al conjunto de algoritmos que permiten encontrar las componentes conectadas existentes en una imagen digital. Generalmente estos algoritmos reciben como entrada imágenes binarias. En algunos casos, se permiten imágenes en escala de grises. Nótese que en el centro del CCA está el criterio de vecindad utilizado. La cantidad de componentes conectadas encontradas depende del criterio de vecindad que se utilice. Al respecto lo más común es que se utilicen los criterios  $N_4(p)$  y  $N_8(p)$ . En [23] se aborda de manera muy clara y precisa el análisis de las componentes conectadas. Por otro lado en [24] se describen cinco algoritmos para encontrar las componentes conectadas presentes en una imagen digital. Entre ellos se describe el algoritmo implementado por la función `bwlabel` de MATLAB.

### 1.1.10. Rotación

La rotación es un algoritmo de preprocesamiento de imágenes digitales. Exactamente, es un algoritmo de preprocesamiento basado en transformaciones geométricas.

Los algoritmos de transformación geométrica modifican las características geométricas de la imagen, moviendo el punto en las coordenadas  $(x, y)$  hacia una nueva posición  $(x', y')$ . La ecuación que modela el movimiento depende del tipo de transformación geométrica que se aplique. En los algoritmos de transformación geométrica generalmente se utilizan métodos de interpolación ya que es muy común que se obtengan valores no enteros para las variables  $x'$  e  $y'$ , los que no pueden asociarse directamente a posiciones en la imagen digital. En el capítulo 4 del libro [25] se aborda de manera exquisita el problema de la interpolación, como caso particular de la aproximación de funciones.

La ecuación 1.10 muestra la formulación matemática de la rotación. Recuérdese que luego de aplicar

esta ecuación es necesario aplicar un algoritmo de interpolación para garantizar que las coordenadas finales sólo estén compuestas por valores enteros.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (1.10)$$

donde  $(x, y)$  es la posición original y  $\theta$  es el ángulo de la rotación.

Otros algoritmos de transformación geométrica en imágenes digitales son la traslación, el escalado (zoom), y los espejos. En [26] se abordan de manera muy práctica dichos algoritmos.

## 1.2. Algoritmos para corregir la inclinación en imágenes de documentos

Ya fue planteado en este documento que la corrección de la inclinación es una tarea crítica que debe realizarse en los primeros pasos del análisis automático de la imagen del documento. Incluso, aunque no se vaya a analizar automáticamente la imagen, es preciso corregir la inclinación para facilitar la lectura del contenido por parte de un observador humano.

Los algoritmos que corrigen la inclinación se componen básicamente de dos etapas: la estimación, y la rotación. El objetivo de la estimación es calcular el ángulo que representa la inclinación del contenido. En la otra etapa se aplica una transformación geométrica de rotación en la imagen del documento, para eliminar la inclinación existente. Debido a que la rotación de una imagen es un asunto sencillo, el esfuerzo de los investigadores se ha concentrado en cómo estimar fielmente la inclinación.

Para estimar la inclinación en la imagen de un documento se han propuesto varios algoritmos. En [2] se abordan dos categorías de algoritmos: los algoritmos basados en el análisis de las componentes conectadas, y los algoritmos basados en proyecciones.

Los algoritmos basados en el análisis de las componentes conectadas tienen como suposición que las mismas son equivalentes a los símbolos en el texto. Por tanto para estimar el ángulo de la inclinación se encuentran las componentes conectadas presentes en la imagen, se calcula el centroide de cada una, y por cada centroide se halla la inclinación del segmento que lo une con el centroide más cercano. Normalmente el centroide más cercano está ubicado en la misma línea de texto. Luego hallan el promedio del conjunto de ángulos de inclinación, y dicho valor se asume como la inclinación de la imagen. El problema de lo anterior es que se asume que en la segmentación cada símbolo fue separado del fondo como un objeto único, lo cual en la práctica es poco común debido a factores como la baja calidad de la imagen y la existencia de elementos ruidosos (e.g. manchas, tachaduras). Además pudiesen existir problemas en el caso de texto manuscrito, donde como mínimo los caracteres en una misma palabra derivarían en una sola componente conectada, ya que es posible que el centroide más cercano no sea el del siguiente objeto en la misma línea de texto. Otro problema es que el promedio es muy sensible a valores extremadamente ruidosos, conocidos como "outliers". La ventaja de estos algoritmos es que no se necesita conocer a priori el conjunto de posibles ángulos de la inclinación.

Según [2], los algoritmos basados en el análisis de proyecciones proyectan la imagen del documento en varios ángulos y calculan la varianza de cada proyección obtenida. Luego seleccionan la proyección para la cual se obtuvo mayor varianza. El ángulo asociado a dicha proyección es asumido como el valor de la inclinación. La fundamentación de este algoritmo es que la proyección asociada al ángulo de la inclinación tendrá picos cuando cruce por las líneas de texto, y vacíos cuando cruce por el espacio entre líneas. Por tanto, su varianza será elevada. El primer problema de lo anterior es que requiere conocer los ángulos en los que se hallará la proyección. Además tiene implícita una aproximación en los ángulos distintos de  $(0^\circ + 90^\circ K)$ , con  $K$  perteneciente a los enteros, que puede generar errores en la estimación. Otro problema es que una varianza elevada no es un rasgo que describe fielmente al ángulo de la inclinación (la existencia de picos y vacíos espaciados uniformemente, sí). Estos algoritmos también pueden enfrentar problemas ante la presencia de objetos como figuras y decoraciones en la imagen del documento, y ante la existencia de texto dispuesto en varias columnas.

En [1] se abordan tres tipos de algoritmos para estimar la inclinación: los algoritmos basados en proyecciones, los algoritmos basados en la HT, y los algoritmos basados en el vecino más cercano. En el caso de los algoritmos basados en proyecciones, se parte de que la proyección horizontal de un texto dispuesto horizontalmente (sin inclinación) tendrá picos cuando la misma cruce por las líneas de texto y valles cuando cruce por el espacio entre líneas. Además, el ancho de los picos sería aproximado a la altura de los caracteres. Teniendo en cuenta estas suposiciones se proyecta la imagen en varios ángulos, en cada proyección se calcula un indicador que mida la dispersión de la misma, y se selecciona la proyección en la cual se maximizó el indicador. El ángulo asociado a la proyección seleccionada es tomado como el valor de la inclinación. Este algoritmo es similar al propuesto en [2]. Por tanto, ambos comparten las mismas fortalezas y debilidades.

En [1] también se abordan adaptaciones para los algoritmos basados en proyecciones. Una es encontrar las componentes conectadas y representar cada una de ellas por el punto “inferior-central” del rectángulo mínimo que la encierra. Luego se hace una estimación basada en proyecciones (del modo ya abordado), pero utilizando como datos de entrada la nube de puntos y no los píxeles de la imagen. La ventaja de esta variante es que puede ser más robusta, ante la existencia de figuras, que los algoritmos que calculan la proyección directamente a partir de los píxeles de la imagen. La desventaja es que implica un costo computacional adicional en la detección de las componentes conectadas.

Otra adaptación presente en [1], un poco distinta a la concepción clásica de los algoritmos basados en proyecciones, consiste en dividir la imagen en regiones horizontales de igual altura. Por cada región se calcula la proyección vertical, y en cada proyección se halla la posición en la que comienza el pico que está más a la izquierda. Seguido se calcula la inclinación de la línea que mejor cruza por las posiciones encontradas (nótese que las posiciones son relativas al eje  $x$ ). Finalmente se asume como el ángulo de la inclinación del documento, a un ángulo perpendicular a la inclinación de la línea. La idea de este algoritmo es encontrar la línea que representa el borde izquierdo del texto, calcular su inclinación, y estimar la inclinación del documento sumando o restando  $90^0$  ya que dicha línea es perpendicular a la dirección del texto. Este algoritmo tiene varios problemas. Primeramente, es muy propenso al ruido y asume que el texto está alineado a la izquierda. Además es dependiente de la altura de las regiones horizontales en que se recorta la imagen. Encontrar el valor adecuado para dicha altura en aplicaciones reales, puede ser muy difícil.

En [1] se explica cómo utilizar la HT para estimar la inclinación. Para ello se calcula la HT de la imagen, se hallan las coordenadas  $(\rho, \theta)$  del máximo valor en la transformada, y se asume como la inclinación a un ángulo perpendicular a  $\theta$ . El principio que sustenta este algoritmo es que existen grandes probabilidades de que la mayor acumulación de puntos de objeto ocurra a lo largo de una línea de texto, y generalmente la inclinación del documento es igual a la inclinación de cualquiera de sus líneas de texto. Por tanto, el problema se reduce a encontrar la inclinación de la línea de texto más visible. Para esto, la HT es la herramienta idónea. Las ventajas de la estimación basada en la HT son que normalmente la exactitud de la estimación es elevada y que no enfrenta muchas dificultades ante texto manuscrito. Respecto a las desventajas, son varias. La principal es que calcular la HT es una operación costosa computacionalmente. La segunda es que debe conocerse a priori el conjunto de posibles ángulos de la inclinación. Otra es que se pueden obtener errores ante la existencia de objetos como figuras y bordes laterales. También puede haber fallos ante la existencia de texto dispuesto en varias columnas. Un detalle no poco común que puede generar problemas es que la mayor acumulación de puntos de objeto no sea a lo largo de una misma línea de texto, sino a lo largo de una recta imaginaria que contenga caracteres en distintas líneas de texto. Lo anterior es muy común en las regiones de texto que son más altas que anchas, y que poseen el texto alineado en cualquiera de los extremos.

De acuerdo con lo plasmado en [1], los algoritmos basados en el vecino más cercano encuentran las componentes conectadas presentes en la imagen, calculan el centroide de cada una, y por cada centroide hallan la inclinación del segmento que lo une con el centroide más cercano. La inclinación de cada segmento es registrada en un arreglo de votos, y el ángulo es estimado seleccionando la inclinación con más votos en el arreglo. Lo anterior es similar a lo planteado en [2], pero bajo el nombre “algoritmos basados en el análisis de las componentes conectadas”. En consecuencia, sus fortalezas y debilidades

son similares. La única diferencia es que en [1] se utiliza la moda y en [2] se utiliza el promedio.

La fuente [3] es una excelente revisión sobre los algoritmos que estiman la inclinación en imágenes de documentos. De acuerdo a la misma existen cuatro tipos de algoritmos para estimar la inclinación: los algoritmos basados en el análisis de proyecciones, los algoritmos basados en rasgos característicos, los algoritmos basados en la HT, y los algoritmos basados en máscaras. La suposición elemental de todos ellos es que las líneas de texto están espaciadas uniformemente y que dentro de ellas los objetos (símbolos) están alineados.

El principio básico de los algoritmos basados en proyecciones es que la imagen de un texto sin inclinación genera una proyección horizontal (PH) que posee picos y valles espaciados uniformemente. Los picos se obtienen cuando la proyección cruza por las líneas de texto y los valles cuando cruza por el espacio entre líneas. Según [3] estos algoritmos proyectan la imagen en varios ángulos, analizan cada proyección buscando rasgos que indiquen similitud con la PH de una imagen sin inclinación, y combinan dichos rasgos en una función de calidad. Seguido se selecciona la proyección en la cual se maximiza la función de calidad, y el ángulo asociado a la proyección seleccionada es tomado como el valor de la inclinación. Esto es equivalente a lo planteado en [1] y en [2] para los algoritmos basados en proyecciones. Por tanto las fortalezas y debilidades son las mismas. La diferencia es que el enfoque es más general, dado el uso de una "función de calidad".

Los algoritmos basados en rasgos característicos obtienen rasgos de la imagen, útiles de acuerdo al dominio del problema. La meta es hacer la estimación procesando dichos rasgos y no los píxeles de la imagen directamente. Las componentes conectadas son uno de los rasgos más utilizados. Una tendencia muy común dentro de estos algoritmos es detectar las componentes conectadas presentes en la imagen; calcular el centroide de cada una; y calcular la inclinación utilizando un algoritmo basado en proyecciones, pero tomando como datos de entrada la nube de centroides. Una ventaja de esta tendencia es que se puede lograr cierta robustez ante la presencia de objetos como figuras y decoraciones. Otra es que no requiere conocer a priori el conjunto de posibles ángulos de la inclinación. Respecto a las desventajas, una es que asume que los símbolos fueron separados del fondo como objetos únicos, lo cual es poco común en aplicaciones reales (ya se comentó sobre esto anteriormente). Otra es que pueden existir problemas ante texto manuscrito y ante texto dispuesto en varias columnas. En [3] también se presenta otra tendencia de algoritmos basados en rasgos característicos, que es idéntica a lo planteado en [1] bajo el nombre "algoritmos basados en el vecino más cercano".

Según [3] los algoritmos de estimación basados en la HT encuentran la línea más visible en la imagen y asumen como la inclinación de la imagen a la de dicha línea. Algo similar a lo anterior ya fue mencionado en esta sección. Lo interesante de [3] respecto a la estimación basada en la HT es que propone una variante para disminuir la alta complejidad temporal de este tipo de estimación. Consiste en calcular la transformada utilizando una separación angular elevada, seleccionar el ángulo THETA en el cual ocurre el máximo; y volver a calcular nuevamente la HT, con una separación angular muy pequeña, pero en la vecindad del ángulo THETA. La estimación se hace analizando esta nueva transformada. Ciertamente de ese modo se disminuye la complejidad computacional del algoritmo, sobre todo mientras más grande sea la separación angular inicial. Sin embargo, unido a los demás problemas típicos de la estimación basada en la HT, la variante padece de otros: ¿qué separación angular (ampliada) utilizar?, ¿qué separación angular (reducida) utilizar?, ¿cuál es el límite de la vecindad? Sin dudas, la fiabilidad de la estimación depende de los valores que se le asignen a dichas variables.

En [3] se abordan los algoritmos basados en máscaras. Estos algoritmos, como su nombre lo indica, utilizan máscaras que le dan preferencia a un ángulo dado. Los mismos pasan cada máscara por la imagen y cuentan las coincidencias que tuvo la máscara en los puntos de la imagen. Seguido hallan la máscara que tuvo más coincidencias. El ángulo asociado a dicha máscara es tomado como el valor de la inclinación. La principal ventaja de la estimación basada en máscaras es su sencillez y fácil implementación. Una desventaja es que deben conocerse a priori el conjunto de posibles ángulos. Además cada máscara debe favorecer notablemente el ángulo asociado, y otorgarle un peso lo menor posible a los demás ángulos. Esta relación, muy importante para obtener buenos resultados, puede ser difícil de lograr. Otro problema está ligado con el tamaño de las máscaras: a más tamaño, quizás mayor

precisión, pero mayor es también el tiempo de respuesta del algoritmo. Es oportuno plantear que la sensibilidad del algoritmo ante elementos ruidosos, texto manuscrito, y otros factores, depende del diseño de las máscaras.

En [4] se presenta un algoritmo de estimación basado en el análisis de la HT. Lo interesante de este algoritmo es que preprocesa la imagen del documento antes de calcular la HT. Se compone de dos etapas: preprocesamiento, y análisis de la HT. En la primera etapa se encuentran las componentes conectadas a partir de la imagen binaria de entrada, se halla el rectángulo mínimo (MBR: Minimum Bounding Rectangle) que encierra a cada componente, se calcula la altura promedio  $h$  de los MBRs hallados, y se descartan las componentes conectadas cuyo MBR sea de altura mayor que  $h$ . También se descartan las componentes conectadas cuya altura sea muy pequeña. El objetivo de la primera eliminación es obviar letras mayúsculas, números, y caracteres como  $b, d, f, h, p, l$ . El objetivo de la segunda eliminación es obviar los signos de puntuación, los acentos, y otros elementos ruidosos. Seguidamente el algoritmo aplica una dilatación para generar rectángulos a partir de las componentes conectadas que no fueron eliminadas, aplica un algoritmo de adelgazamiento (thinning) para reducir cada rectángulo a sus líneas centrales, y elimina todas las líneas que son más altas que anchas. La meta de esta última eliminación es generar una imagen que contenga solamente las líneas que con mayor probabilidad representan la inclinación del texto. En la segunda etapa se calcula la HT a partir de la imagen de líneas, se hallan las coordenadas  $(\rho, \theta)$  asociados al máximo valor de la transformada, y se asume como valor de la inclinación a un ángulo perpendicular a  $\theta$ .

La relevancia del algoritmo descrito en [4] consiste en el preprocesamiento que hace de la imagen, antes de analizar la HT. Ciertamente para estimar la inclinación no es necesario tener en cuenta todos los objetos. Incluso, es mejor tener en cuenta sólo aquellos que mejor puedan representar la inclinación. Desde otro punto de vista, en el preprocesamiento se aplican tres operaciones de mediana complejidad computacional las cuales añaden más tiempo de cómputo a una estimación basada en el análisis de la HT, que es reconocida como una de las más lentas. Estas tres operaciones son la detección de las componentes conectadas, la dilatación, y el adelgazamiento. Unido a lo anterior, el algoritmo intenta ser robusto ante la existencia de objetos como figuras y bordes de página, pero en verdad no lo es. La debilidad radica en que dichos objetos son normalmente mucho más grandes que los símbolos del texto, y se tienen en cuenta durante el cálculo de  $h$  (la altura promedio de los MBRs). Recuérdese que el promedio es una medida de posición central sensible a los valores extremos ("outliers"), por lo cual posiblemente bastaría una imagen medianamente grande para que el valor de  $h$  sea mayor que la altura de los símbolos del texto. En consecuencia, las componentes asociadas a los símbolos de texto no se tendrían en cuenta para generar la imagen de líneas a partir de la cual se analiza la HT, lo cual es perjudicial. Hubiese sido mejor utilizar una medida de posición central insensible a valores extremos, por ejemplo la mediana. El algoritmo también puede enfrentar problemas ante la existencia de texto dispuesto en varias columnas. Como otro elemento a destacar, el algoritmo asume imágenes binarias como datos de entrada. Sin embargo en ambientes reales, los datos de entrada son normalmente imágenes en colores o en escala de grises. Por lo demás, debido a que está basado en el análisis de la HT, comparte las mismas fortalezas y debilidades de este tipo de algoritmos. Las mismas ya han sido planteadas anteriormente.

En [5] se propone un algoritmo de corrección basado en el análisis de la RT. Primeramente se calcula la RT a partir de la imagen de entrada, se encuentran las coordenadas  $(\rho, \theta)$  asociadas al máximo valor en la transformada, y se rota la imagen  $(\theta - 90^\circ)$  en sentido contrario a las manecillas del reloj. Seguido se calcula la RT de la imagen rotada y se comprueba que el ángulo  $\theta$  asociado a las coordenadas del máximo es  $90^\circ$ . Si  $\theta$  es igual a  $90^\circ$ , lo cual implica que la inclinación de la mayor línea visible en la imagen es  $0^\circ$ , el proceso termina. En caso contrario el proceso continúa.

El algoritmo propuesto en [5] no tiene relevancia alguna. Es una clásica estimación de la inclinación, asumiendo como inclinación de la imagen a la inclinación de la mayor línea visible. Para estimar la inclinación de dicha línea se utiliza la RT. Recuérdese que para el caso de detección de líneas, la RT y la HT son equivalentes. Por otro lado no especifica el tipo de imagen que se recibe como datos de entrada. En la detección de líneas mediante la HT o la RT, es recomendable que los datos de entrada sean imágenes binarias. Por tanto puede que el algoritmo propuesto en [5] esté obviando el hito de

segmentación, el cual es de notable importancia y tiene sus complicaciones. Además el algoritmo puede enfrentar problemas ante la existencia de figuras y bordes de página en la imagen. También puede enfrentar problemas ante la existencia de texto dispuesto en varias columnas. Para no redundar, el algoritmo tiene las mismas fortalezas y debilidades que los algoritmos de estimación basados en el análisis de la HT.

El algoritmo propuesto en [27] está basado en la detección de las componentes conectadas y en el análisis de la HT. Primero se encuentran las componentes conectadas existentes en la imagen y se seleccionan aquellas cuyo MBR cumpla una condición de tamaño. El objetivo de dicha condición es evadir objetos ruidosos como signos de puntuación, bordes laterales, líneas, tablas y figuras. Seguido se calcula el centroide de cada componente, y la nube de centroides generada se utiliza como datos de entrada para una estimación basada en el análisis de la HT. El análisis de la HT se hace de un modo que ya fue visto anteriormente en este documento. Primero se calcula la HT utilizando una separación angular elevada  $S_1$ , se selecciona el ángulo  $\theta$  asociado al máximo, y luego se analiza la HT en la vecindad de  $\theta$ , pero utilizando una separación angular reducida  $S_2$  ( $S_2 \ll S_1$ ). Las ventajas y desventajas de este análisis de la HT en dos niveles ya fueron enunciadas anteriormente en este documento.

El algoritmo propuesto en [27] es interesante en cuanto utiliza centroides y no píxeles como datos de entrada para calcular la HT. Esto disminuye el tiempo de respuesta del algoritmo ya que el tamaño de los datos de entrada es mucho menor. El análisis de la HT en dos niveles también es llamativo, pero recuérdense los problemas implícitos en ello: ¿cuál valor utilizar para la separación angular ampliada?, ¿cuál valor utilizar para la separación angular reducida?, ¿cómo delimitar la vecindad del ángulo  $\theta$  hallado inicialmente? Es muy interesante también que el algoritmo intente eliminar la mayor cantidad posible de objetos ruidosos, antes de estimar la inclinación mediante el análisis de la HT. Este preprocesamiento de los datos de entrada es algo que ha estado ausente prácticamente en todos los algoritmos vistos anteriormente, y es vital para obtener buenos resultados. Lo preocupante es que dicho preprocesamiento se hace mediante una simple condición que verifica el tamaño. La misma puede funcionar en algunos casos. En otros, donde la calidad de la imagen no sea buena, lo más probable es que no funcione. En dichos casos se requerirá de un método de preprocesamiento más robusto.

En [6] se presenta un algoritmo muy bueno que está basado en el análisis de las componentes conectadas. Primero preprocesa la imagen binaria de entrada para eliminar objetos (ruidosos) pequeños como tildes y signos de puntuación. Seguido garantiza que el fondo sea de color negro y los símbolos sean de color blanco. El tercer paso consiste en dilatar la imagen con una estructura en forma de línea, orientada horizontalmente y de longitud  $l$ . El valor de  $l$  es calculado empíricamente de forma tal que se unan las palabras que estén en una misma línea de texto. Además se llenan los huecos existentes en la imagen dilatada (un hueco es conjunto de puntos negros encerrado totalmente por puntos blancos). Finalmente se encuentran las componentes conectadas, se calcula la inclinación de cada una, y se asume como inclinación de la imagen al promedio del conjunto que contiene la inclinación de cada componente. La inclinación de cada componente es calculada inscribiéndola en una elipse y hallando el ángulo que forma el eje mayor de la elipse respecto a la horizontal.

El algoritmo descrito en [6] es rápido y no necesita conocer a priori el conjunto de posibles ángulos de la inclinación. Puede ser eficaz ante texto manuscrito. Respecto a sus debilidades intenta ser robusto ante imágenes que contengan figuras, pero no lo es del todo. Nótese que al calcular el promedio se tiene en cuenta la inclinación de todas las componentes conectadas. Sin embargo, en el caso de las figuras es posible que la orientación del eje mayor de la elipse no se corresponda con la inclinación del texto. En consecuencia se estarían introduciendo valores que desvían al promedio de la inclinación real. La idea de calcular la inclinación local inscribiendo la componente en una elipse y hallando el ángulo que forma el eje mayor respecto a la horizontal, es teóricamente excelente. El asunto es que exige una alta precisión al hallar la elipse: el eje mayor de la misma debe ser paralelo al texto. Lograr esta precisión puede ser un reto en el campo del procesamiento de imágenes digitales. Otro problema del algoritmo radica en la longitud de la línea que se emplea en la dilatación. Debido a que su valor es establecido manualmente, se está asumiendo que las imágenes de entrada serán similares en aspectos como el tamaño de fuente. Puede que la homogeneidad anterior no sea posible en algunos casos.

En [9] se presenta un algoritmo de estimación basado en el análisis del gradiente. El principio de este algoritmo es que en la imagen de un documento, la dirección del vector gradiente en la mayoría de los puntos deberá ser perpendicular a las líneas de texto. En el algoritmo se calcula la dirección del vector gradiente en cada punto de la imagen y se registra en un histograma  $h$ . Las derivadas parciales involucradas en el cálculo de dicha dirección son obtenidas mediante el operador de Sobel. Seguido se suaviza a  $h$  mediante un filtrado basado en la mediana, y se selecciona la dirección que experimentó mayor acumulación en  $h$ . Finalmente se toma un ángulo perpendicular a la dirección seleccionada, y se refina mediante el ajuste de una función cúbica  $y = ax^3 + bx^2 + cx + d$ . El valor refinado es asumido como el valor de la inclinación.

El algoritmo propuesto en [9] es rápido y fácil de implementar. Otra de sus ventajas es que no necesita conocer a priori el conjunto de posibles ángulos de la inclinación. Su principal desventaja es que puede ser poco robusto ante imágenes ruidosas, en las cuales no se cumpla que en la mayoría de los puntos la dirección del vector gradiente sea perpendicular a la dirección del texto. También puede ser poco robusto ante imágenes que contengan figuras, porque los píxeles de las figuras se tienen en cuenta al generar el histograma  $h$ . Para lograr robustez no debería ser así. Sólo deberían tenerse en cuenta los píxeles que conforman los símbolos del texto. Incluso sería mejor no tener en cuenta todos los símbolos, sino los más representativos de acuerdo al dominio del problema. Otro problema está relacionado con el refinamiento que se hace al final del algoritmo, mediante el ajuste de una función cúbica. Puede ser engorroso encontrar las constantes  $a$ ,  $b$ ,  $c$  y  $d$  que permitan obtener los mejores resultados.

En [28] se presenta un algoritmo de estimación extremadamente sencillo. Consiste en detectar las coordenadas de los puntos de objetos más alejados en la imagen. Estos puntos extremos son: superior-izquierdo, superior-derecho, inferior-izquierdo, e inferior-derecho. Seguido se calcula el centroide de dichos puntos, y se traza un segmento desde el origen del sistema hasta el centroide. La inclinación del segmento es asumida como la inclinación del documento. Este algoritmo es sencillo y fácil de implementar. Además no requiere conocer a priori el conjunto de posibles ángulos de la inclinación. Su gran desventaja radica en que es extremadamente propenso al ruido. Basta un punto ruidoso que desvíe el valor de los puntos extremos, para que el algoritmo genere resultados incorrectos. Téngase en cuenta que en la práctica las imágenes son normalmente ruidosas. Además la idea de hacer la estimación hallando la inclinación del segmento que une el origen de coordenadas con el centroide, es dudosa. ¿Qué hacer en el caso de un texto sin inclinación? En dicha situación el segmento estará inclinado, pero la inclinación de la imagen es  $0^0$ . Quizás el autor omitió en el artículo detalles importantes.

El algoritmo descrito en [7] es también extremadamente sencillo. Consiste en encontrar el primer punto de objeto en cada columna de la imagen. Luego se emplea una técnica de regresión para estimar la línea que mejor se ajusta a los puntos encontrados. La inclinación de dicha línea es asumida como la inclinación de la imagen del documento.

El fundamento de lo descrito en [7] es que la inclinación del contenido que está en la parte superior de la imagen, generalmente debe corresponderse con la inclinación global. Esto es cierto y obvio para un observador humano, pero cuando se trabaja a nivel de pixel surgen las complicaciones típicas del procesamiento de imágenes digitales: ¿estarán alineados los píxeles superiores de los símbolos existentes a lo largo de una línea de texto?, ¿cómo manejar la separación entre los símbolos del texto? Esto último es muy interesante ya que quizás en la columna  $i$  el primer punto de objeto sea el pixel superior de un símbolo en la línea de texto  $n$ , pero debido a un espacio entre palabras, perfectamente puede ocurrir que el primer punto de objeto en la columna  $(i + 1)$  sea el pixel superior de un símbolo en la línea de texto  $(n + 1)$ <sup>2</sup>. Esto provocaría desviaciones al estimar la línea. Súmese a los problemas anteriores, que el algoritmo es muy sensible al ruido. Además asume como entrada imágenes binarias. En la práctica lo más común es encontrar imágenes en escala de grises o en colores, por lo cual el algoritmo está obviando un paso crítico (la segmentación) que no es trivial. Como ventajas, el algoritmo es sencillo y fácil de implementar. Además no requiere conocer a priori el conjunto de posibles ángulos de la inclinación.

---

<sup>2</sup> Los valores de  $i$  están ordenados ascendentemente de izquierda a derecha. Los de  $n$  están ordenados ascendentemente de arriba hacia abajo.

El algoritmo descrito en [8] recibe como entrada imágenes binarias. Se compone de tres hitos: la detección de las componentes textuales, la estimación de la inclinación en las componentes textuales, y la estimación de la inclinación global.

En el primer hito se utiliza una técnica de análisis de segmentos (“run-length”) para generar una imagen en escala de grises a partir de la imagen binaria. Dicha imagen en escala de grises es segmentada en dos clases. Posteriormente se encuentran las componentes conectadas existentes en la imagen resultante y se seleccionan las que cumplan ciertos criterios respecto a la altura, la longitud, y la densidad. Dichas componentes seleccionadas son asumidas como las textuales.

En el segundo hito se divide cada componente textual en  $n$  segmentos verticales de igual tamaño. En cada segmento se seleccionan las coordenadas del punto de objeto que está más arriba (*superior*), las del punto de objeto que está más abajo (*inferior*), y las del punto comprendido en el medio de los dos puntos hallados anteriormente (*medio*). Seguido se agrupan los puntos *superiores* en un conjunto, los *inferiores* en un conjunto, y los *medios* en otro conjunto. En cada conjunto se utiliza el método de los mínimos cuadrados para hallar la recta que mejor representa los puntos contenidos en el mismo. Lo anterior arroja tres rectas por componente textual. La recta que obtuvo el valor mínimo en la suma de los residuales es elegida como la representativa de la componente textual. Por tanto la inclinación de dicha recta es asumida como la inclinación de la componente textual.

En el tercer hito se calcula la inclinación global haciendo un promedio (basado en pesos) de la inclinación local en cada componente textual. La longitud de la componente textual es el criterio empleado para otorgar el peso. Mientras mayor sea la longitud de la componente textual, mayor es el peso asociado a su inclinación local.

El algoritmo propuesto en [8] es muy bueno. Su relevancia es que separa las componentes textuales de las no textuales, y tiene en cuenta sólo las textuales para hacer la estimación. En consecuencia ofrece cierta robustez ante texto mezclado con figuras. Además puede manejar exitosamente texto escrito a máquina y manuscrito. Respecto a las desventajas la primera es que asume imágenes binarias como entrada, y en la práctica generalmente las imágenes son en colores o en escala de grises. Otra es que el algoritmo que se emplea para estimar la inclinación en las componentes textuales puede ser propenso al ruido, aunque no tanto como otros vistos anteriormente en este documento que también usan aproximación de funciones.

Teniendo en cuenta todo lo revisado anteriormente puede plantearse que existen seis categorías principales de algoritmos para corregir la inclinación en imágenes de documentos: algoritmos basados en componentes conectadas, algoritmos basados en proyecciones, algoritmos basados en la HT (o la RT), algoritmos basados en la aproximación de funciones, algoritmos basados en máscaras, y algoritmos basados en el gradiente. De las seis categorías, las primeras cuatro son las más populares.

El fundamento de los algoritmos basados en componentes conectadas es que las mismas son equivalentes a los símbolos del texto. Estos algoritmos hallan las componentes conectadas, seleccionan el punto representativo de cada una, y estiman la inclinación local del segmento que une cada punto con su vecino más cercano (otro punto). Finalmente la inclinación global de la imagen es calculada a partir del conjunto de inclinaciones locales. Lo más común es utilizar como punto representativo al centroide, o algún punto en las esquinas del MBR asociado a la componente conectada. Respecto al cálculo de la inclinación global a partir de las inclinaciones locales, generalmente se emplean el promedio, la moda o la mediana. La ventaja de estos algoritmos es que no necesitan conocer a priori el conjunto de posibles ángulos de la inclinación. Una desventaja es que no siempre cada símbolo del texto es segmentado como un objeto único, lo cual puede provocar errores en la estimación. Otra es que pueden existir problemas en el caso de texto manuscrito ya que puede que el punto representativo más cercano sea el de una componente en otra línea de texto. Lo anterior introduce desviaciones. Además pueden existir problemas ante imágenes con ruido ya que los elementos ruidosos también son procesados como componentes conectadas. Por tanto aportan datos que desvían a la inclinación de su valor real.

El fundamento de los algoritmos basados en proyecciones es que la proyección horizontal de un texto sin inclinación se caracteriza por tener picos y valles espaciados uniformemente. Los picos se obtienen



cuando la proyección cruza por las líneas de texto y los valles se obtienen cuando cruza por el espacio entre líneas (o viceversa en dependencia del color del fondo). Estos algoritmos proyectan la imagen en varios ángulos, lo cual genera una proyección por cada ángulo. Seguido seleccionan la proyección con mayor similitud a la proyección horizontal de un texto sin inclinación. El ángulo asociado a la proyección seleccionada es tomado como el valor de la inclinación. Estos algoritmos pueden ofrecer buenos resultados ante texto manuscrito. Uno de sus problemas es que requieren conocer a priori el conjunto de posibles ángulos de la inclinación. Otro es que implican una aproximación en los ángulos distintos a  $(0^\circ + 90^\circ K)$ , con  $K$  perteneciente a los enteros, que puede generar problemas en la estimación. Además pueden existir problemas en el caso de texto dispuesto en varias columnas y texto mezclado con figuras. También puede haber problemas ante bordes laterales.

Los algoritmos basados en la HT (o la RT) se basan en que las líneas de texto generan acumulaciones elevadas en las celdas correspondientes de la transformada. Además en que la inclinación de la imagen es igual a la inclinación de cualquiera de sus líneas de texto. Por tanto el problema se reduce a encontrar la inclinación de una línea de texto, y asumir dicho valor como la inclinación de la imagen del documento. Estos algoritmos calculan la transformada y encuentran las coordenadas  $(\rho, \theta)$  del máximo valor en la misma. Supuestamente dicho máximo debe estar asociado a la línea de texto más visible. El valor que se toma como la inclinación de la imagen es un ángulo perpendicular a  $\theta$ . Una ventaja de estos algoritmos es que ofrecen alta exactitud en la estimación. Además ofrecen buenos resultados ante texto manuscrito. Sus desventajas son varias. Una es que requieren conocer a priori el conjunto de posibles ángulos de la inclinación. Otra es que calcular la transformada es una operación costosa computacionalmente. Además pueden existir problemas ante imágenes que tengan figuras, y ante texto dispuesto en varias columnas. También puede haber problemas ante imágenes con bordes laterales.

Los algoritmos basados en aproximación de funciones localizan un conjunto de puntos de la imagen, siguiendo criterios determinados. Seguido hallan la recta que mejor se ajusta a dichos puntos y su inclinación es tomada como la inclinación de la imagen. Estos algoritmos son sencillos y fáciles de implementar. Además no requieren conocer a priori el conjunto de posibles ángulos de la inclinación. Su principal desventaja es que generalmente son muy sensibles al ruido (en parte depende del algoritmo de aproximación que se utilice). Su robustez ante imágenes de documentos complejos, depende de los criterios que se utilicen para determinar los puntos iniciales.

Los algoritmos basados en máscaras, como su nombre lo indica, emplean máscaras que le dan preferencia a un ángulo dado. Los mismos pasan cada máscara por la imagen y cuentan las coincidencias que tuvo la máscara en los puntos de la imagen. El ángulo asociado a la máscara que tuvo más coincidencias es tomado como el valor de la inclinación. Estos algoritmos son sencillos y fáciles de implementar. Una de sus desventajas es que requieren conocer a priori el conjunto de posibles ángulos de la inclinación, para diseñar las máscaras correspondientes. Además para obtener buenos resultados cada máscara debe favorecer notablemente a su ángulo asociado, y otorgarle un peso lo menor posible a los demás ángulos. Téngase en cuenta que esta relación puede ser difícil de lograr.

Los algoritmos basados en el gradiente emplean la dirección del mismo para hallar la inclinación de la imagen. Generalmente calculan la dirección del gradiente en cada punto de la imagen. Luego utilizan medidas como el promedio o la moda para calcular una dirección global y toman un ángulo perpendicular a dicha dirección como el valor de la inclinación. Estos algoritmos son sencillos y fáciles de implementar. No requieren conocer a priori el conjunto de posibles ángulos de la inclinación. Su principal desventaja es que son muy propensos al ruido. Además son poco robustos ante imágenes que tengan figuras o bordes ruidosos.

En el estudio realizado se evidenció que son muy frecuentes los algoritmos híbridos. Estos algoritmos hacen la estimación combinando características típicas de las categorías ya mencionadas. Algo muy común es combinar el análisis de las componentes conectadas con el análisis de la HT o la RT. También es común combinar en análisis de las componentes conectadas con el análisis de proyecciones.

En la revisión se identificaron problemas comunes en los algoritmos existentes. La mayoría de los algoritmos asumen imágenes binarias como entrada. Sin embargo en la práctica las imágenes son en

escala de grises o en colores, generalmente. En consecuencia los algoritmos están pasando por alto el hito de segmentación. Este hito es crucial para obtener buenos resultados, y no es trivial. Por otro lado, los algoritmos existentes no aplican técnicas de mejoramiento y restauración sobre las imágenes de entrada. Seguramente serían más robustos y precisos si aplicaran dichas técnicas.

Junto a lo anterior existe un asunto muy importante al cual no se le dedica la atención requerida. Consiste en eliminar de la imagen segmentada todos los objetos no significativos para estimar la inclinación. Algunos de ellos son las figuras, los bordes laterales, los signos de puntuación, y las tildes. Esta eliminación de objetos ruidosos debe hacerse antes de estimar la inclinación. Al así hacerlo se estaría aumentando la robustez de la corrección, sobre todo ante las figuras y los objetos ruidosos.

En la actualidad el principal problema al estimar la inclinación no está en la selección de la técnica. Está en la selección y el mejoramiento de los datos sobre los que se ejecuta una técnica determinada. Las técnicas, con sus ventajas y desventajas, están bien fundamentadas y establecidas. Al respecto un problema de la mayor parte de los algoritmos revisados es que para estimar la inclinación procesan la imagen completa. Esto no es necesario. Basta con procesar las regiones de texto que mejor representen la inclinación del contenido. Normalmente todo el texto en la imagen de un documento, tiene la misma inclinación. Esta última idea no es aprovechada a cabalidad en los algoritmos existentes. De serlo se pudiesen obtener mejores tiempos de respuesta. Incluso se pudiesen obtener mejores tasas de acierto ya que los objetos ruidosos serían descartados.

Todos los problemas referidos anteriormente provocan que los algoritmos desarrollados hasta el momento para corregir la inclinación, no sean capaces de generar resultados satisfactorios para el caso de las imágenes de documentos complejos. En el Anexo 2, "Resultados retornados por otros algoritmos", se muestra la evidencia experimental de lo afirmado anteriormente.

Además en la revisión se evidenció que no existe un algoritmo que corrija la inclinación eliminando de la imagen del documento las regiones con mayor acumulación de puntos de objeto y haciendo la estimación mediante el análisis local de la RT de las regiones más significativas. Los algoritmos más cercanos a lo anterior fueron el propuesto en [4] y el propuesto en [6]. El algoritmo propuesto en [4] no elimina las regiones con mayor acumulación de puntos, sino las que cumplen ciertos criterios en cuanto a la altura y al ancho. El problema de emplear la altura y el ancho para hacer la eliminación, es que ambos descriptores son dependientes de la rotación. Por otra parte, el algoritmo propuesto en [6] no hace la estimación mediante el análisis de la RT.

Finalmente, desde el punto de vista metodológico en la corrección de la inclinación existe un gran problema. Radica en que no existe una metodología que especifique los hitos involucrados en la corrección de la inclinación. Dicha metodología deberá tener un enfoque integral, abarcando desde la digitalización de la imagen del documento hasta la obtención de la imagen digital libre de inclinación.

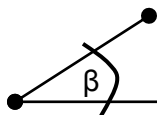
## **CONCLUSIONES PARCIALES**

En este capítulo se concluye que:

1. Existen seis categorías principales de algoritmos para estimar la inclinación: algoritmos basados en componentes conectadas, algoritmos basados en proyecciones, algoritmos basados en la HT (o la RT), algoritmos basados en la aproximación de funciones, algoritmos basados en máscaras, y algoritmos basados en el gradiente. Las cuatro primeras categorías son las más populares y son frecuentes los métodos híbridos.
2. El diseño de los algoritmos existentes no permite obtener resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos.
3. No existe un algoritmo que corrija la inclinación eliminando de la imagen del documento las regiones con mayor acumulación de puntos de objeto y haciendo la estimación mediante el análisis local de la RT de las regiones más significativas.
4. Teniendo en cuenta su eficacia, se empleará la estimación basada en la Transformada Radon

como base para elaborar un nuevo algoritmo que permita obtener resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos. En este nuevo algoritmo se introducirán elementos que contrarresten las desventajas de la estimación basada en dicha transformada.

5. No existe una metodología que especifique los hitos involucrados en la corrección de la inclinación y que tenga un enfoque integral, abarcando desde la digitalización de la imagen del documento hasta la obtención de la imagen digital libre de inclinación.



## **CAPÍTULO 2. METODOLOGÍA Y ALGORITMO PROPUESTOS**

En este capítulo se describe una metodología que especifica los hitos involucrados en la corrección de la inclinación. Dicha metodología tiene un enfoque integral, abarcando desde la digitalización de la imagen del documento hasta la obtención de la imagen libre de inclinación. En el capítulo también se describe un nuevo algoritmo para corregir la inclinación en imágenes de documentos. Este algoritmo implementa cada paso de la metodología propuesta y está diseñado de modo tal que genere resultados satisfactorios al procesar imágenes de documentos complejos.

### **2.1. Metodología para corregir la inclinación**

La metodología se compone de seis operaciones y abarca desde la digitalización de la imagen del documento hasta la obtención de la imagen libre de inclinación. Dichas operaciones son:

- **Adquirir:** Recibe como entrada una imagen continua y produce una imagen digital. La nueva imagen digital puede ser binaria, en escala de grises, en colores, o indexada.
- **Mejorar:** Recibe como entrada la imagen digital. El formato de dicha imagen puede ser binario, en escala de grises, o en colores. Mejora la calidad visual de la imagen de entrada y produce otra imagen en el mismo formato (“la imagen mejorada”).
- **Segmentar:** Recibe como entrada la imagen mejorada. El formato de dicha imagen puede ser en colores o en escala de grises. Separa a partir de la imagen mejorada los puntos de fondo de los puntos de objeto. Normaliza el color del fondo. Produce una imagen binaria (“la imagen segmentada”) donde una clase está asociada a los puntos de objeto y la otra está asociada a los puntos de fondo.
- **Preprocesar:** Recibe como entrada la imagen segmentada y elimina de la misma los objetos que no son significativos para estimar la inclinación. Algunos de estos objetos son los bordes laterales y las figuras. Produce una imagen binaria (“la imagen preprocesada”).
- **Estimar:** Recibe como entrada la imagen preprocesada y calcula la inclinación predominante de los objetos presentes en la misma. Produce un escalar (“el ángulo de la inclinación”).
- **Rotar:** Recibe como entrada la imagen original y el ángulo de la inclinación. Aplica una transformación geométrica de rotación en la imagen original para corregir la inclinación estimada. Produce una imagen de igual formato a la imagen original (“la imagen libre de inclinación”).

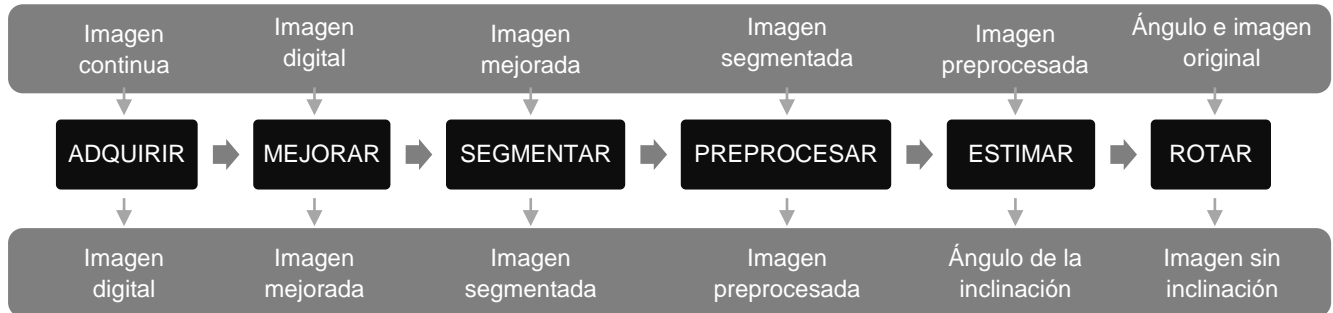
Nótese en la especificación anterior que la operación Mejorar no admite como entrada imágenes indexadas. Por tanto si en un contexto dado se están adquiriendo imágenes indexadas, deberá hacerse una conversión previa a otro formato antes de ejecutar la mejora. En esta operación no se permiten imágenes indexadas ya que muy posible que los algoritmos utilizados en la misma generen valores no existentes en la paleta de colores.

Junto a lo anterior véase que una de las responsabilidades de la operación Segmentar es normalizar el color del fondo de la imagen del documento. Por tanto luego de esta etapa todas las imágenes tendrán el mismo color de fondo, sin importar cuál era el color de fondo original. Esta normalización es necesaria para el buen funcionamiento de las etapas posteriores y es un factor crítico ante imágenes de entrada muy heterogéneas.

Existe un caso especial. Consiste en que si las imágenes adquiridas son binarias, la operación

Segmentar se reduce a normalizar el color del fondo de la imagen del documento.

Los comentarios anteriores sobre los tipos de imágenes han sido necesarios para que la metodología propuesta sea clara y precisa. Sin embargo se aclara que en la mayoría de los casos las imágenes serán en colores o en escala de grises. En la figura 2.1 se muestra la interacción entre las seis operaciones descritas anteriormente.



**Figura 2.1.** Interacción entre las operaciones que componen la metodología propuesta.

## 2.2. Nuevo algoritmo para corregir la inclinación

El nuevo algoritmo está diseñado para manipular imágenes en escala de grises. No obstante es capaz de procesar los demás tipos de imágenes haciendo conversiones previas.

Para el caso de las imágenes en colores RGB se hace una conversión hacia el espacio de color YIQ (también llamado NTSC), y se toma la matriz bidimensional de valores Y como la imagen en escala de grises (Ec 2.1). La conversión se hace de este modo ya que en el modelo de color YIQ la componente Y almacena la información sobre la escala de gris, y las demás componentes almacenan información sobre el color [21]. Además esta es la forma más aceptada para convertir imágenes en el espacio de color RGB a imágenes en escala de grises. Es incluso la vía utilizada por la función `rgb2gray` de MATLAB para hacer dicha conversión.

$$Y = R * 0,2989 + G * 0,5870 + B * 0,1140 \quad (2.1)$$

donde  $R$  es el valor de rojo,  $G$  es el valor de verde, y  $B$  es el valor de azul.

En el caso de las imágenes binarias se genera una imagen en escala de grises asociando el valor TRUE al máximo número permitido de acuerdo a la profundidad de bits de la imagen, y el valor FALSE al mínimo número permitido. Las imágenes indexadas son convertidas escribiendo directamente en la imagen los valores existentes en la paleta de colores, y haciendo una conversión desde el modelo de color RGB a escala de grises en caso de ser necesaria. Teniendo en cuenta todo lo planteado, la implementación de la operación Adquirir en el algoritmo producirá siempre imágenes en escala de grises.

La operación Mejorar fue implementada mediante un ajuste del contraste. Este ajuste se hizo saturando el 1% de los datos en las bajas y altas intensidades. El efecto de lo anterior es el incremento de la diferencia entre los píxeles de la imagen, lo cual es favorable para la segmentación (siguiente operación). Se escogió esta técnica de mejoramiento de imágenes digitales ya que está bien establecida, da buenos resultados, y sobre todo es muy veloz. Además como se dijo anteriormente aumenta la diferencia entre los píxeles de la imagen, que es el resultado deseado.

En la operación Segmentar se utilizó una técnica de segmentación basada en umbral, que divide los puntos de la imagen en dos clases: puntos de fondo, y puntos de objeto. El umbral fue estimado utilizando el método de Otsu [19]. Se eligió este método para estimar el umbral ya que por más de 30 años se ha demostrado su efectividad en disímiles entornos prácticos, está excelentemente fundamentado, y es relativamente rápido.

Respecto a los problemas reconocidos del método de Otsu, son básicamente dos. El primero es que

asume que el histograma de la imagen es bimodal<sup>3</sup>. El segundo es que asume que la iluminación es uniforme. Ambas suposiciones se cumplen en las imágenes de documentos. El histograma de la imagen de un documento se caracteriza por tener dos grandes picos (uno asociado al color del fondo y otro asociado al color de los objetos), separados por un gran valle. La elevación de dicho valle es por lo general prácticamente cero. Puede ser mayor ante la presencia de figuras en la imagen, pero nunca similar a la elevación de los picos. Respecto a la segunda suposición (iluminación uniforme) se cumple ya que las imágenes de documentos son capturadas en entornos totalmente controlados mediante un escáner. Además los escáneres están diseñados de manera que la hoja queda totalmente tapada durante el proceso de digitalización. Por todos los factores anteriores, el problema de la iluminación variable no es preocupante.

En el diseño de la operación Preprocesar se tuvo en cuenta el principio de que generalmente los objetos de la imagen segmentada que tienen mayor área, se corresponden con elementos como figuras, dibujos, y bordes laterales. Dichos elementos no son necesarios para hacer la estimación. Por el contrario, aportan ruido. Teniendo en cuenta lo anterior se encontraron las componentes conectadas en la imagen segmentada y se eliminó el 1% de las componentes encontradas, comenzado por las que tienen mayor área.

El algoritmo utilizado para implementar la operación Estimar explota un principio elemental, pero poco aprovechado en los algoritmos existentes. Consiste en que generalmente todas las regiones en la imagen de un documento tienen la misma inclinación. Por tanto no es necesario procesar toda la imagen. Basta con sólo procesar las regiones más significativas.

El algoritmo de estimación se compone de tres hitos: la selección de las regiones significativas, el cálculo de la inclinación local en cada región significativa, y el cálculo de la inclinación global a partir del conjunto de inclinaciones locales.

En el primer hito se dilató la imagen preprocesada utilizando una estructura en forma de disco, se hallaron las componentes conectadas en la imagen dilatada; y se seleccionó el 25% de las componentes encontradas, comenzando por las que tienen mayor área. Dichas componentes seleccionadas fueron asumidas como las “regiones significativas”.

La selección de las regiones significativas se hizo de este modo ya que las componentes conectadas con mayor área, se corresponden generalmente con las regiones en las que luego de la dilatación se experimentó unión entre símbolos de una misma palabra, palabras a lo largo de una misma línea de texto, e incluso distintas líneas de texto. Dichas regiones son las recomendadas para hacer la estimación porque en ellas se aprecia muy bien la inclinación de la imagen del documento. Se utilizó el área ya que es un descriptor invariante a la rotación. Esto último (independencia de la rotación) es un aspecto muy importante. Por ejemplo si se hubiese utilizado un descriptor dependiente de la rotación, como el ancho de la región, se hubiesen enfrentado problemas ya que el ancho de una región sin inclinación se transforma en el alto de la región al rotarla 90 grados.

Un aspecto crítico y novedoso en este primer hito es el cálculo automático del radio del disco que se emplea en la dilatación. Para calcularlo se hallaron las componentes conectadas en la imagen preprocesada, por cada una se calculó el diámetro del círculo con igual área a la de la componente, y se seleccionó la moda del conjunto compuesto por los diámetros calculados. El cálculo se hizo de este modo para obtener un valor de radio dependiente del tamaño de los símbolos, lo cual es necesario para que el algoritmo en general se ajuste automáticamente a cambios en el tamaño de los símbolos. Se seleccionó la moda debido a que comúnmente existe gran similitud entre el área de los símbolos presentes en la imagen de un documento.

Para calcular la inclinación local (segundo hito) se aplicó una reconstrucción morfológica a la región de entrada, con el objetivo de rellenar los huecos existentes [29]. A la región resultante de la reconstrucción morfológica se le aplicó un algoritmo morfológico de adelgazamiento para reducirla a una línea [30]. Finalmente se calculó la RT de la región resultante del adelgazamiento. El espectro de ángulos utilizados

---

<sup>3</sup> Histograma que posee dos picos, separados por un valle claramente definido.

en el cálculo de la RT fueron todos los números enteros en el intervalo  $[-179, 0]$ . La inclinación de la región fue calculada hallando las coordenadas  $(\rho, \theta)$  del máximo valor en la RT, y sumando 90 grados al valor de  $\theta$ .

La esencia de este segundo hito es el preprocesamiento adicional que se realiza en la región significativa, antes de calcular y analizar la RT. El objetivo de dicho preprocesamiento es reducir la región a una línea que represente la inclinación de la región. Si no se hiciese esta reducción podría haber pequeños errores de exactitud al hacer la estimación.

La estimación de la inclinación se hizo mediante el análisis de la RT ya que dicha tendencia ofrece una alta exactitud en la estimación y brinda buenos resultados ante texto manuscrito.

Respecto a las desventajas de esta tendencia, ya fueron mencionadas en este documento. Una es que requiere conocer a priori el conjunto de posibles ángulos de la inclinación. Otra es que calcular la RT es una operación costosa computacionalmente. Además súmese que pueden existir problemas ante la existencia de figuras, bordes laterales, y texto dispuesto en varias columnas.

De todas las desventajas planteadas, la primera es la única que no se ataca en el algoritmo propuesto. Se plantea esto ya que en el nuevo algoritmo el cálculo de la RT se está haciendo solamente sobre el 25% de las componentes conectadas existentes, y no sobre toda la imagen. Normalmente dichas componentes conectadas son pequeñas, por lo cual el cálculo es muy rápido. Respecto a la robustez ante la existencia de figuras y bordes laterales, es algo que fue manejado en la implementación de la operación Preprocesar. En consecuencia la imagen que constituye la entrada del método que implementa la operación Estimar, no tendrá bordes laterales ni figuras.

En relación a la robustez del nuevo algoritmo ante imágenes con texto dispuesto en varias columnas, recuérdese que el análisis de la RT se hace de manera local en las regiones significativas seleccionadas (no en toda la imagen). Además debido a que el algoritmo que selecciona las regiones significativas se ajusta automáticamente a cambios en el tamaño de los símbolos, es muy poco probable que en una región significativa existan símbolos pertenecientes a distintas columnas.

En el tercer hito del algoritmo que implementa la operación Estimar, la inclinación global fue calculada hallando la moda del conjunto que contiene las inclinaciones locales. Se eligió esta medida de posición central (la moda) ya que normalmente todas las regiones de la imagen de un documento tienen la misma inclinación. Por lo tanto basta con hallar la inclinación más frecuente en las regiones significativas procesadas. Lo anterior es precisamente lo que la moda posibilita.

En el nuevo algoritmo se utilizó una transformación geométrica de rotación para implementar la operación Rotar. El tipo de interpolación que se utilizó en la rotación fue por el vecino más cercano. Se empleó este tipo de interpolación principalmente porque es el más rápido de todos.

En el Anexo 3, “Evolución del algoritmo propuesto”, se demuestra cómo evoluciona el algoritmo descrito en este capítulo, para corregir la inclinación en una imagen determinada.

### **2.3. Valoración preliminar**

La metodología propuesta es breve pero describe claramente las operaciones que deben realizarse en cualquier algoritmo que corrija la inclinación en imágenes de documentos. Por tanto sirve como guía a la hora de diseñar un nuevo algoritmo, como ha ocurrido con el propuesto en esta investigación.

El enfoque integral que propone la metodología está ausente en los trabajos previos. Normalmente los autores se concentraban en mejorar las técnicas de estimación, prestando poca atención al mejoramiento de los datos que sirven como entrada a una técnica de estimación dada. Los escasos trabajos que hacían mejoramiento de los datos, lo hacían a partir de imágenes binarias. Se obviaba el hecho de que en los entornos prácticos las imágenes son generalmente en colores o en escala de grises. Una consecuencia de lo anterior es que se descartaba completamente la aplicación de técnicas de mejoramiento y restauración de imágenes digitales, sobre las posibles imágenes en escala de grises o en colores de entrada.

Respecto al algoritmo propuesto, además de estar adherido a la metodología presentada, es novedoso en cuanto al sencillo pero eficaz método de preprocesamiento que utiliza. Dicho método de preprocesamiento es extremadamente rápido. El algoritmo en general también es novedoso en cuanto al método que se utiliza para seleccionar las regiones significativas que sirven como entrada a la estimación basada en el análisis de la RT. En dicha selección se destaca el uso del área como un descriptor invariante a la rotación. Además sobresale el criterio que se emplea para calcular automáticamente el radio del disco que se emplea en la dilatación. En la literatura normalmente el tamaño de las estructuras empleadas en las operaciones morfológicas, es establecido manualmente. Incluso en algunos casos ni se menciona este asunto, pero es una sencillez en teoría que puede generar muchísimos problemas en la práctica.

Otro elemento novedoso es el preprocesamiento adicional que se realiza sobre cada región significativa, antes de calcular y analizar la RT. Ciertamente en otros trabajos se reducían los objetos de la imagen a líneas, pero no del modo realizado en el algoritmo propuesto mediante la aplicación de las operaciones morfológicas descritas.

El algoritmo que se propone es totalmente automático. Puede ser aplicado en distintos bancos de imágenes, y generar resultados satisfactorios. No es necesario que las imágenes en los bancos sean similares. Nótese que variando el por ciento que indica la cantidad de regiones significativas a procesar (25% por defecto), se puede controlar la velocidad del algoritmo que se propone en esta tesis. Al respecto lo recomendable es que el valor de este por ciento no sea elevado para así evadir las regiones pequeñas. Tales regiones pueden generar problemas al hacer la estimación. Además se repite que mientras más elevado sea dicho por ciento, menos rápido será el algoritmo en general.

Nótese también que el algoritmo está diseñado de modo que pueda lidiar eficazmente con imágenes de documentos que tengan bordes laterales, bordes de página, figuras, gráficos, texto dispuesto en varias columnas, y texto manuscrito.

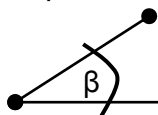
## **CONCLUSIONES PARCIALES**

En este capítulo se concluye que:

1. La metodología propuesta, que describe las operaciones a realizar en cualquier algoritmo de corrección de la inclinación en imágenes de documentos, constituye un aporte de esta investigación.
2. El algoritmo propuesto elimina de la imagen del documento las regiones con mayor acumulación de puntos de objeto, y estima la inclinación analizando localmente la Transformada Radon de las regiones más significativas
3. El algoritmo propuesto es novedoso en cuanto al método de preprocesamiento que se utiliza, en cuanto al método que se utiliza para seleccionar las regiones significativas, y en cuanto al preprocesamiento adicional basado en operaciones morfológicas que se aplica a cada región significativa antes de calcular y analizar la RT.
4. El algoritmo propuesto es totalmente automático, y está diseñado de modo que retorne resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos.



*¿Por qué será que para decir que un aporte científico está bueno, obligatoriamente tenemos que decir que otros están malos?*



**El autor**

### **CAPÍTULO 3. RESULTADOS Y DISCUSIÓN**

En este capítulo se describen los archivos de código fuente generados al implementar el algoritmo propuesto. También se describe la composición del banco de imágenes utilizado en las pruebas, y se reflejan los resultados arrojados por el nuevo algoritmo. En el capítulo se discuten las fortalezas y debilidades del algoritmo propuesto en esta tesis, y se demuestra su superioridad respecto a algoritmos precedentes.

#### **3.1. Implementación en MATLAB del algoritmo propuesto**

Se utilizó la herramienta MATLAB 2008b para implementar el algoritmo propuesto en esta tesis. El paquete (toolbox) que se utilizó fue “Image Processing”. Durante la implementación se generaron cinco archivos de código fuente. El archivo “enhance.m” contiene la implementación de la operación Mejorar, y el archivo “segment.m” contiene la implementación de la operación Segmentar. La implementación de la operación Preprocesar está en el archivo “preprocess.m” y la implementación de la operación Estimar está en el archivo “estimate.m”. El archivo “main.m” contiene la implementación de la operación Rotar. Además contiene el código que controla la interacción entre las operaciones que constituyen el algoritmo propuesto.

El código expuesto en cada uno de los archivos mencionados anteriormente puede ser consultado en el Anexo 4, “Codificación en MATLAB del algoritmo propuesto”.

#### **3.2. Composición del banco de imágenes utilizado en las pruebas**

El banco de imágenes de prueba fue elaborado por el autor, ya que no se tenían los recursos necesarios para obtener uno previamente elaborado. Este banco se conforma de imágenes en dos grandes categorías: imágenes con texto escrito a máquina, e imágenes con texto manuscrito.

Para obtener las imágenes con texto escrito a máquina se seleccionaron de libros, artículos de revista, y otros materiales afines, 50 imágenes sin inclinación por cada una de las siguientes subcategorías:

- **Subcategoría 1:** Imágenes con texto dispuesto en varias columnas, y no mezclado con objetos ruidosos como figuras y decoraciones.
- **Subcategoría 2:** Imágenes con texto dispuesto en varias columnas, mezclado con objetos ruidosos.
- **Subcategoría 3:** Imágenes con texto dispuesto en una columna, y no mezclado con objetos ruidosos.
- **Subcategoría 4:** Imágenes con texto dispuesto en una columna, mezclado con objetos ruidosos.

Lo anterior permitió disponer de 200 imágenes con texto escrito a máquina, y sin inclinación. Se destaca que en la mayoría de dichas imágenes existían bordes laterales. Además en el caso de las imágenes con figuras, estas ocupaban diversas posiciones en la imagen, tenían varios tamaños, y exhibían distintos colores y formas. Se resalta también que existían variaciones en el tipo de letra, el tamaño de letra, el interlineado, el estilo (e.g. negrita, cursiva), la sangría, y los márgenes del texto. En el caso de las imágenes con texto dispuesto en varias columnas, la separación entre dichas columnas era variable. Respecto a la calidad de las imágenes, era baja.

Cada una de las 200 imágenes mencionadas anteriormente fue rotada respecto a los ángulos -5, -10, -15, -20, -25, -30, -36, -42, -48, -60, -72, -75, -80, -85, 5, 10, 15, 20, 25, 30, 36, 42, 48, 60, 72, 70, 80, y 85 grados. En total fueron 28 ángulos diferentes. Esto permitió disponer de 1450 imágenes por cada subcategoría, lo que es igual a 5800 imágenes en total con texto escrito a máquina. Se hizo esta rotación para aumentar la cantidad de imágenes a procesar en las pruebas, y sobre todo para verificar la efectividad de los algoritmos ante distintas inclinaciones.

Las imágenes con texto manuscrito fueron generadas utilizando un procesador de texto y empleando tipos de letra con estilo manuscrito. Se tuvo que hacer de este modo ya que aunque se tenían hojas con texto manuscrito, no se disponía de un medio de digitalización.

La generación de las imágenes con texto manuscrito se hizo de modo similar a lo realizado para obtener las imágenes con texto escrito a máquina. Para ello se generaron 50 imágenes sin inclinación por cada una de las cuatro subcategorías ya mencionadas. En todos los casos se trataba de páginas tipo Carta (21,59 cm x 27, 94 cm) con márgenes de 2 cm en todos los lados, e interlineado sencillo con un espacio de 5 puntos entre párrafos. Se destaca que cada grupo de 50 imágenes estaba dividido en dos subgrupos de 25 imágenes. En las imágenes del primer grupo el texto se escribió utilizando el tipo de letra *Brush Script MT*, con 13 puntos como tamaño de letra. En las imágenes del segundo subgrupo se utilizó el tipo de letra *Lucida Handwritten*, con 12 puntos como tamaño.

Lo descrito en el párrafo anterior permitió disponer de 200 imágenes con texto manuscrito, y sin inclinación. En la mayoría de dichas imágenes existían bordes laterales y en todas existían bordes de página. En el caso de las imágenes con figuras, estas ocupaban diversas posiciones en la imagen, tenían varios tamaños, y exhibían distintos colores y formas. Al igual que las imágenes con texto escrito a máquina, la separación entre las columnas del texto era variable y la calidad de las imágenes era baja.

Para generar más imágenes con texto manuscrito, se rotaron las 200 imágenes (sin inclinación) respecto a los 28 ángulos ya mencionados en esta sección. Esto permitió disponer de 5800 imágenes con texto manuscrito para hacer las pruebas. Se hizo esta rotación para comprobar la efectividad de los algoritmos ante imágenes con texto manuscrito, y con varias inclinaciones.

Resumiendo, el banco de imágenes de prueba se compone de 11600 imágenes en total. En la Tabla 3.1 se muestra detalladamente la composición del mismo.

**Tabla 3.1.** Composición del banco de prueba.

Ángulo	Texto escrito a máquina				Texto manuscrito				Subtotal
	SCat1	SCat2	SCat3	SCat4	SCat1	SCat2	SCat3	SCat4	
-5	50	50	50	50	50	50	50	50	400
-10	50	50	50	50	50	50	50	50	400
-15	50	50	50	50	50	50	50	50	400
-20	50	50	50	50	50	50	50	50	400
-25	50	50	50	50	50	50	50	50	400
-30	50	50	50	50	50	50	50	50	400
-36	50	50	50	50	50	50	50	50	400
-42	50	50	50	50	50	50	50	50	400
-48	50	50	50	50	50	50	50	50	400
-60	50	50	50	50	50	50	50	50	400
-72	50	50	50	50	50	50	50	50	400

-75	50	50	50	50	50	50	50	50	400
-80	50	50	50	50	50	50	50	50	400
-85	50	50	50	50	50	50	50	50	400
0	50	50	50	50	50	50	50	50	400
5	50	50	50	50	50	50	50	50	400
10	50	50	50	50	50	50	50	50	400
15	50	50	50	50	50	50	50	50	400
20	50	50	50	50	50	50	50	50	400
25	50	50	50	50	50	50	50	50	400
30	50	50	50	50	50	50	50	50	400
36	50	50	50	50	50	50	50	50	400
42	50	50	50	50	50	50	50	50	400
48	50	50	50	50	50	50	50	50	400
60	50	50	50	50	50	50	50	50	400
72	50	50	50	50	50	50	50	50	400
75	50	50	50	50	50	50	50	50	400
80	50	50	50	50	50	50	50	50	400
85	50	50	50	50	50	50	50	50	400
<b>Subtotal</b>	1450	1450	1450	1450	1450	1450	1450	1450	
<b>Total</b>	5800				5800				
<b>Total general</b>	11600								11600

### 3.3. Resultados obtenidos mediante el algoritmo propuesto

Se probó el algoritmo en las 11600 imágenes que constituyen el banco de prueba. Las métricas seleccionadas para evaluar el comportamiento de dicho algoritmo (y el de los precedentes) son detalladas a continuación:

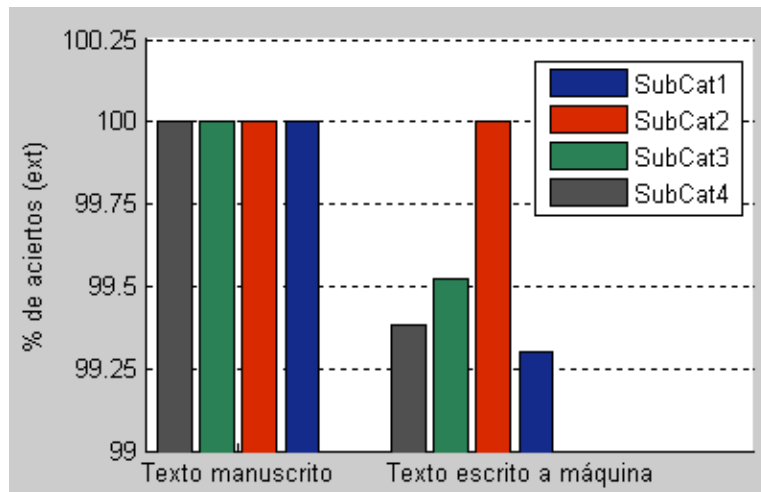
- **Total de fallos:** Total de casos en los que la inclinación estimada fue distinta a la inclinación real. Dicho total se divide en dos subtotales. El primer subtotal indica la cantidad de casos en los que la diferencia absoluta entre la inclinación real y la inclinación estimada fue menor o igual a un grado. El segundo subtotal indica la cantidad de casos en los que se superó dicha diferencia.
- **Por ciento de aciertos:** Por ciento de imágenes en las que la inclinación estimada fue igual a la inclinación real. Se utilizó también un por ciento extendido. En el cálculo del mismo se aceptaron como aciertos los casos en los que la diferencia absoluta entre la inclinación real y la inclinación estimada fue menor o igual a un grado. Se consideró útil utilizar el por ciento extendido porque el pequeño error de estimación que acepta ( $\mp 1^{\circ}$ ), es prácticamente insignificante para propósitos prácticos.
- **Error medio absoluto (MAE):** Promedio de las diferencias absolutas entre la inclinación estimada y la inclinación real, obtenidas al procesar cada imagen en la muestra.

- **Error mediano absoluto (MdAE):** Mediana de las diferencias absolutas entre la inclinación estimada y la inclinación real, obtenidas al procesar cada imagen en la muestra.

Préstese especial atención al uso de las métricas MAE y MdAE [31]. Se seleccionaron ya que permiten evaluar el tamaño del fallo. Un fallo con un error de estimación de un grado no tiene la misma relevancia que un fallo con un error de estimación de 45 grados. Obviamente el segundo error es más severo. Ambas métricas son similares, pero no iguales. La diferencia es que el MAE no es robusto ante errores de estimación extremos y ruidosos (outliers) mientras que el MdAE, al estar basado en la mediana, sí lo es.

Otras métricas como la precisión y el *recall*, a pesar de ser muy utilizadas en otros campos, no son las mejores para evaluar un algoritmo que corrija la inclinación en imágenes de documentos. Se plantea esto ya que dichas métricas no miden el tamaño del fallo. Lo que ellas hacen es medir los errores de inserción, sustitución y eliminación que arroja un algoritmo, tomando como base una referencia o *ground-truth* [32].

En el Gráfico 3.1 se resumen los resultados arrojados por el algoritmo propuesto en esta tesis, al probarlo en las 11600 imágenes que constituyen el banco de prueba. Dichos resultados se muestran detalladamente en las Tablas A1.1 y A1.2 existentes en el Anexo 1, “Resultados retornados por el algoritmo propuesto”.



**Gráfico 3.1.** Resultados generados por el algoritmo propuesto en esta tesis.

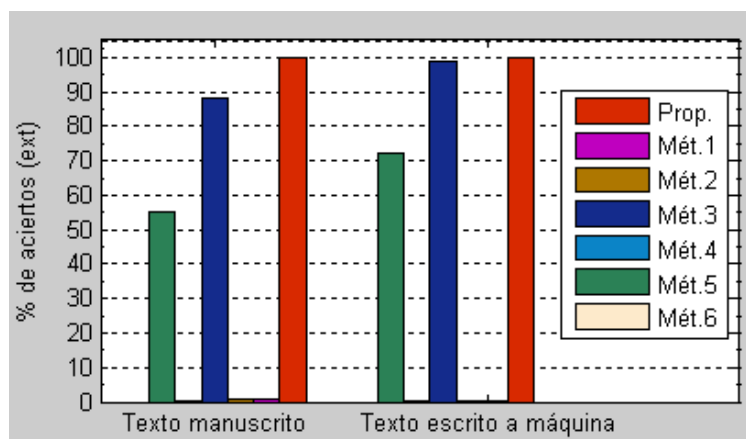
### 3.4. Discusión

Los resultados generados por el algoritmo propuesto en esta tesis son excelentes.

Se implementaron seis algoritmos precedentes para comparar los resultados arrojados por los mismos al procesar las imágenes en el banco de prueba, con los resultados arrojados por el algoritmo propuesto. El primero está basado en el análisis clásico de la RT (Método 1). El segundo está basado en el análisis de proyecciones (Método 2). El tercero combina el análisis de la RT con el análisis de las componentes conectadas (Método 3). El cuarto se basa en el análisis multiresolución de la RT (Método 4). En el quinto se preprocesa la imagen para eliminar algunos objetos y luego se analiza la RT (Método 5). En el sexto también se preprocesa la imagen, pero la estimación se hace analizando las componentes conectadas (Método 6). Todos fueron implementados utilizando la herramienta MATLAB 2008b. Los resultados arrojados por estos seis algoritmos se presentan en el Anexo 2, “Resultados arrojados por otros algoritmos”.

El algoritmo propuesto brindó resultados superiores a los otros seis algoritmos (Gráf 3.2). Nótese que estos seis algoritmos representan las principales categorías de algoritmos para corregir la inclinación. Se aclara que no se implementaron algoritmos basados en aproximación de funciones, ya que los mismos son extremadamente sensibles ante bordes laterales en las imágenes de entrada, y prácticamente todas

las imágenes en el banco de prueba muestran bordes laterales. En consecuencia se sabía a priori que los resultados serían malos. Igualmente no se implementaron métodos basados en el gradiente ni en máscaras ya que son poco usados, y sobre todo, son muy sensibles ante la existencia de bordes laterales, bordes de página, figuras, y puntos ruidosos.



**Gráfico 3.2.** Por cientos de aciertos (extendidos) generados por el algoritmo propuesto y por los seis precedentes.

Nótese en el Gráfico 3.2 que los algoritmos con resultados más cercanos a los retornados por el algoritmo propuesto fueron el que combina el análisis de las componentes conectadas con el análisis de la RT (Método 3), y el que preprocesa la imagen y luego analiza la RT (Método 5). Los demás algoritmos brindaron resultados muy malos, siendo en ellos el por ciento extendido de aciertos cero o prácticamente cero (véase Anexo 2).

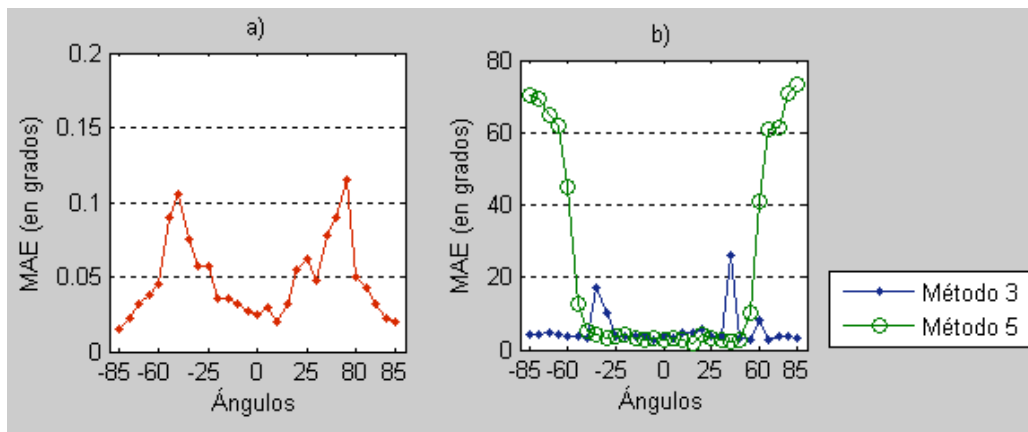
El Método 3 arrojó buenos resultados ya que al utilizar la nube de centroides como datos de entrada para calcular la RT, pudo lograr cierta robustez ante imágenes con bordes laterales, bordes de página, y figuras. Sépase que en el banco de imágenes había muchísimas imágenes con las características anteriores. Factores ante los que el Método 3 no es robusto, son la existencia de varias columnas de texto y de texto manuscrito. Esta incapacidad del Método 3 fue la que provocó la diferencia con los excelentes resultados retornados por el algoritmo propuesto.

Los resultados arrojados por el Método 5 fueron regulares. El preprocesamiento que se hace en el mismo intenta dotarlo de robustez ante la existencia de figuras, bordes laterales y bordes de página, pero no lo logra. La debilidad radica en el cálculo de la altura promedio y en la eliminación de las componentes conectadas que son más altas que anchas. Esta debilidad ya fue detallada en la revisión crítica al algoritmo propuesto en [4], realizada en la Sección 1.2 de esta tesis. La misma fue la que provocó los resultados regulares arrojados por este algoritmo.

Los malos resultados arrojados por los Métodos 1, 2 y 4 son consecuencia de que dichos algoritmos no preprocesan la imagen con el objetivo de eliminar los objetos no significativos para estimar la inclinación. Por tanto son muy sensibles ante la existencia de bordes de página, bordes laterales, figuras y decoraciones en las imágenes de entrada. Se recuerda que en el banco de prueba se incluyeron muchísimas imágenes con estas características ya que son muy comunes en la práctica cotidiana.

El Método 6 sí preprocesa la imagen, pero hace la estimación utilizando la inclinación del eje mayor de las elipses obtenidas al encerrar cada componente conectada en una elipse. Los inconvenientes de este tipo de estimación fueron planteadas en la revisión crítica al algoritmo propuesto en [6], realizada en la Sección 1.2 de esta tesis. El por ciento extendido de aciertos (con valor cero) arrojado por el Método 6 demuestra la veracidad de las críticas realizadas.

En el Gráfico 3.3 se comparan el algoritmo propuesto, el Método 3, y el Método 5, tomando como criterio el MAE. En este gráfico el eje horizontal se corresponde con los 29 ángulos de inclinación representados en las imágenes del banco de prueba. El eje vertical indica el MAE obtenido al procesar las imágenes en el banco de prueba que tienen como inclinación un ángulo dado. Nótese el pequeño MAE arrojado por el algoritmo propuesto. Nunca es superior a los 0,115 grados. Respecto a los Métodos 3 y 5 se demuestra que tomando como criterio el MAE, el Método 3 sigue siendo el mejor. No se incluye el MAE de los algoritmos restantes por una cuestión de claridad. En todos los casos, al igual que con el por ciento extendido de aciertos, los valores de MAE generados fueron muy malos.



**Gráfico 3.3.** Errores medios absolutos (MAE) obtenidos. a) MAE generado por el algoritmo propuesto. b) MAEs generados por los Métodos 3 y 5.

De los algoritmos precedentes, los más similares al algoritmo propuesto en esta tesis son el propuesto en [4] y el propuesto en [6]. Se recuerda que el algoritmo propuesto en [4] preprocesa la imagen y luego analiza la HT. El algoritmo propuesto en [6] también preprocesa la imagen, pero hace la estimación utilizando el ángulo del eje mayor de las elipses. Ambos algoritmos son buenos pero tienen debilidades (ya criticadas) que los incapacitan de retornar resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos.

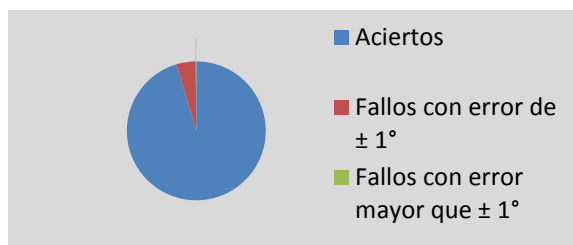
El algoritmo propuesto en esta tesis se diferencia del propuesto en [4] y del propuesto en [6], primeramente porque está adherido a la metodología que se propone en esta tesis para corregir la inclinación en imágenes de documentos. En consecuencia mejora la imagen y tiene en cuenta el hito de segmentación. Ambas tareas no son realizadas por los dos algoritmos precedentes bajo comparación. Además el algoritmo propuesto hace un preprocesamiento de manera sencilla, rápida y efectiva, que garantiza la eliminación de bordes de página, bordes laterales, figuras y decoraciones, entre otros objetos ruidosos. En este preprocesamiento es clave la utilización del área como un descriptor invariante a la rotación. Recuérdese que un problema del preprocesamiento que se hace en [4] es que se utilizan dos descriptores dependientes de la rotación: el alto y el ancho. Un problema del preprocesamiento que se hace en [6] consiste en que la longitud de la línea empleada en la dilatación es establecida manualmente. Por tanto se asume que las imágenes de entrada serán similares en aspectos como el tamaño de fuente. La homogeneidad anterior puede ser no posible en algunos casos.

El algoritmo propuesto también se diferencia de los precedentes en cuanto al método que utiliza para generar las regiones significativas, y en cuanto al preprocesamiento adicional que se hace sobre cada región significativa antes de calcular y analizar la RT. El modo en que se implementó la operación Estimar en el algoritmo permitió aprovechar las ventajas de la estimación basada en el análisis de la RT (e.g. precisión, robustez ante texto manuscrito) y disminuir el alto tiempo de cómputo de este tipo de estimación. Se plantea lo último ya que la RT es calculada solamente en una reducida parte de la imagen (véase Anexo 3).

Los excelentes resultados generados por el algoritmo propuesto al probarlo en las 11600 imágenes existentes en el banco de prueba, demuestran que es capaz de brindar resultados satisfactorios ante imágenes con texto manuscrito o escrito a máquina, dispuesto en una o en varias columnas. También es capaz de brindar resultados satisfactorios ante imágenes que tengan bordes laterales, bordes de página, figuras, y decoraciones. En otras palabras: los resultados obtenidos demuestran que el algoritmo propuesto brinda resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos.

Los resultados arrojados por el algoritmo propuesto son evidencia clara de sus fortalezas. El algoritmo propuesto mejora la imagen de entrada, lo cual incrementa su robustez ante datos de entrada ruidosos. Además aplica un método de segmentación probado en disímiles entornos, que le permite procesar exitosamente imágenes de documentos envejecidos (el efecto más común del envejecimiento de un documento de papel es que se torna amarillo debido a la exposición de la lignina al aire y a la luz). Debido al método de preprocesamiento que se utiliza, el algoritmo propuesto es robusto ante imágenes de documentos con bordes de página, bordes laterales, figuras, gráficos y decoraciones, entre otros objetos. Además debido a la manera en que se estima la inclinación, es robusto ante texto dispuesto en varias columnas y ante texto manuscrito. También es robusto ante cambios en el tipo de letra, el tamaño de letra, y el interlineado.

Toda obra resultante del intelecto humano tiene defectos y por tanto se puede mejorar. El mayor defecto del algoritmo propuesto radica en el pequeño error de  $\mp 1^{\circ}$  respecto a la inclinación real que se produce en algunas imágenes (Gráf 3.4) Dichos errores son pequeños en magnitud, son pocos comparados con el total de imágenes procesadas, pero como su nombre lo indica: son errores.



**Gráfico 3.4.** Total de aciertos y fallos generados por el algoritmo propuesto. De un total de 11600 imágenes, se obtuvieron 11069 aciertos (95,42%), 514 fallos en los que el error fue de  $\mp 1^{\circ}$  (4,43%), y 17 fallos en los que el error fue superior a  $\mp 1^{\circ}$  (0,15%).

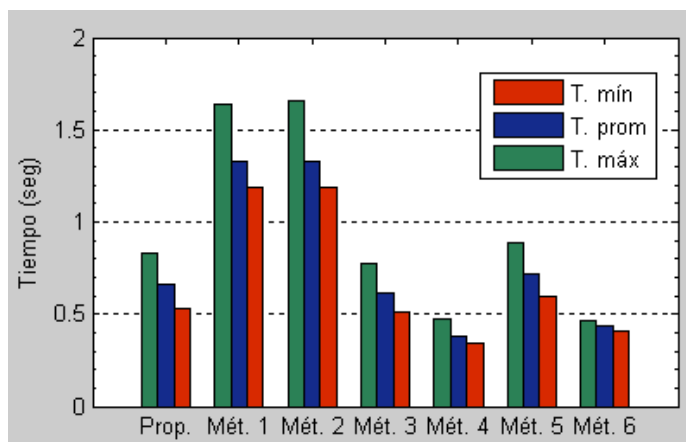
El algoritmo propuesto en esta investigación no es una panacea. Tiene varias debilidades. Puede haber problemas si la imagen de entrada contiene más figuras, bordes laterales, bordes de página, y decoraciones, que texto. Ante dicha situación es posible que luego del preprocesamiento todavía existan en la imagen elementos ruidosos combinados con el texto, lo cual afectaría notablemente el resultado de la estimación. Otra situación es cuando la imagen sólo contiene texto, pero en una cantidad muy limitada (e.g. una palabra, una línea de texto). El problema derivado de lo anterior es que el método de preprocesamiento eliminará texto de una imagen que ya tiene poco texto, pudiendo incluso borrarlo completamente. Para erradicar este problema debe introducirse una condición en el método de preprocesamiento. Dicha condición debe garantizar que al menos una componente conectada sobreviva al proceso de eliminación.

A las dos situaciones anteriores se suma otra: si los símbolos del texto son extremadamente gruesos, y la separación entre las líneas de texto es muy pequeña, es posible que la dilatación aplicada durante la estimación genere un objeto compacto. Por tanto es probable que el resultado de la estimación sea incorrecto. La causa de este problema es que el radio del disco que se emplea en la dilatación depende del área de los objetos. Esta cualidad es la que permite que el algoritmo propuesto se ajuste automáticamente a cambios en el tamaño de los caracteres, pero se puede apreciar que no es un remedio universal ya que provoca el problema abordado en este párrafo.

Por fortuna, las situaciones asociadas a las debilidades del algoritmo propuesto son muy raras en la práctica cotidiana. Generalmente en las imágenes de documentos predomina el texto por encima de los demás elementos, hay muchas líneas de texto, y la separación entre las mismas es adecuada.

Se eligieron 50 imágenes para hacer una prueba del tiempo de cómputo del algoritmo propuesto y de los de los otros seis. La prueba consistió en registrar el tiempo mínimo, el tiempo promedio, y el tiempo máximo obtenidos al procesar con un algoritmo dado cada una de las 50 imágenes, una vez. Se hizo en una computadora con dos núcleos (2,20 GHz cada uno) y de 1 GB de RAM. El sistema operativo utilizado fue Microsoft Windows XP SP 3. La herramienta utilizada fue MATLAB 2008b.

A pesar de que el algoritmo propuesto realiza más tareas que los demás (e.g. mejora de la imagen), su tiempo de cómputo fue similar al de los algoritmos que ofrecieron mejor tiempo de cómputo (Gráf 3.5). Por otro lado, en la gráfica se demuestra de manera empírica el alto tiempo de cómputo requerido por los algoritmos que calculan la RT de toda la imagen (Método 1). También se evidencia el impacto positivo que se obtiene en el tiempo de cómputo al hacer un cálculo multiresolución de la RT (Método 4).



**Gráfico 3.5.** Tiempos de cómputo obtenidos al ejecutar el algoritmo propuesto y los otros seis en un conjunto de 50 imágenes.

El novedoso algoritmo propuesto en esta tesis puede ser empleado inmediatamente en las aplicaciones informáticas que lo requieran. El problema de la corrección de la inclinación está presente en cualquier aplicación que interactúe con imágenes de documentos digitalizados. Algunas de ellas son los sistemas de reconocimiento óptico de caracteres, las soluciones para el análisis automático de formularios, y las soluciones para la digitalización de grandes archivos de documentos históricos. Se recuerda que en el Anexo 4 se presenta la implementación en MATLAB del algoritmo propuesto en esta tesis. Una ventaja de implementar el algoritmo utilizando MATLAB es que el código resultante puede ser exportado a otras plataformas como C/C++, Java y .NET. Esta facilidad aumenta el espectro de posibles aplicaciones que se pueden beneficiar con el resultado de esta investigación.

### CONCLUSIONES PARCIALES

En este capítulo se concluye que:

1. El algoritmo propuesto mejora los resultados retornados por los algoritmos precedentes al corregir la inclinación en imágenes de documentos complejos.
2. El algoritmo propuesto brinda resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos.
3. El algoritmo propuesto es apto para ser utilizado en las aplicaciones informáticas que lo requieran.



## **CONCLUSIONES GENERALES**

En esta tesis se concluye que:

1. El algoritmo propuesto, que elimina de la imagen del documento las regiones con mayor acumulación de puntos de objeto y que estima la inclinación analizando localmente la Transformada Radon de las regiones más significativas, brinda resultados satisfactorios al corregir la inclinación en imágenes de documentos complejos lo cual permite plantear que la hipótesis planteada en esta investigación es verdadera.
2. El algoritmo propuesto mejora los resultados retornados por los algoritmos precedentes al corregir la inclinación en imágenes de documentos complejos.
3. El algoritmo propuesto es novedoso en cuanto al algoritmo de preprocesamiento que se utiliza, en cuanto al método que se utiliza para seleccionar las regiones significativas, y en cuanto al preprocesamiento adicional basado en operaciones morfológicas que se aplica a cada región significativa antes de calcular y analizar la RT.
4. El algoritmo propuesto es totalmente automático y es apto para ser utilizado en las aplicaciones informáticas que lo requieran.
5. La metodología propuesta, que describe las operaciones a realizar en cualquier algoritmo de corrección de la inclinación en imágenes de documentos, constituye un aporte de esta investigación.

## TRABAJO FUTURO

Las siguientes tareas requerirán esfuerzos en el futuro para darle continuidad a esta investigación:

1. Probar el algoritmo propuesto en un banco de imágenes más grande y más complejo. Hacer énfasis sobre todo en la utilización de imágenes con texto manuscrito real.
2. Plantear un nuevo método para implementar la operación Segmentar, que sea eficaz ante imágenes de documentos con fondo uniforme o no. Ejemplos de fondos no uniformes son los que muestran varios colores, transiciones entre colores, y tramas. El método de segmentación además debe ser eficaz ante imágenes en las que el texto se haya escrito utilizando varios colores.
3. Plantear un nuevo método para implementar la operación Estimar, que permita erradicar los pequeños errores de estimación de  $\mp 1^0$  padecidos por el algoritmo propuesto y que conserve o mejore la efectividad de dicho algoritmo.
4. Plantear un nuevo método para implementar la operación Estimar, que esté basado en el cálculo multiresolución de la RT y que conserve o mejore la efectividad del algoritmo propuesto. El cálculo multiresolución de la RT permitirá reducir todavía más el tiempo de cómputo requerido por el algoritmo propuesto.
5. Plantear un método para eliminar los bordes laterales existentes en la imagen resultante de la corrección de la inclinación. Dichos bordes laterales se originan durante la rotación de la imagen original. Idealmente el método debe seleccionar el contenido de interés existente en la imagen sin inclinación, y desechar los bordes laterales existentes. Por tanto genera una imagen de menor tamaño.

## REFERENCIAS

- [1] Lawrence O’Gorman and Rangachar Kasturi, *Document image analysis.*, 2009.
- [2] Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu, and Ching Y. Suen, *Character recognition systems. A guide for students and practitioners.*: Wiley Interscience, 2007.
- [3] Jonathan J. Hull, "Document image skew detection: survey and annotated bibliography," *Document analysis system*, vol. II, pp. 40-64, 1998.
- [4] Manjunath Aradhya, Hemantha Kumar, and Shivakumara P, "Skew detection technique for binary document images based on Hough Transform," *International Journal of Information Technology*, 2007.
- [5] Rajiv Kapoor, Deepak Bagai, and T.S. Kamal, "A new algorithm for skew detection and correction," *Pattern recognition letters*, pp. 1215-1229, 2004.
- [6] B. V Dhandra, V. S Malemath, Mallikarjun H, and Ravindra Hegadi, "Skew detection in binary image documents based on image dilation and region labeling approach," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006.
- [7] Mohamed El-Adawi, Hesham A. Keshk, Manal A. Ismail, and Essam M. Zaki, "Fast skew and slant correction for arabic written word or line,".
- [8] Zhixin Shi and Venu Govindaraju, "Skew detection for complex document images using fuzzy runlength," in *Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*, 2003.
- [9] Khairuddin Omar, Rahman Ramli, Ramlan Mahmod, and Nasir Sulaiman, "Skew detection and correction of jawi images using gradient direction," *Jurnal teknologi*, pp. 117-126, 2002, Universiti Teknologi Malaysia.
- [10] Rafael C. González and Richard E. Woods, *Digital image processing*, 2nd ed. New Jersey: Prentice Hall, 2002.
- [11] The Mathworks, Inc. (2010) Image Processing Toolbox - Image Types in the Toolbox. [Online]. <http://www.mathworks.com/help/toolbox/images/f14-13543.html>
- [12] Peter Toft, "The Radon Transform. Theory and implementation," 1996.
- [13] Richard O Duda and Peter E Hart, "Use of the Hough Transformation to detect lines and curves in pictures," 1972.
- [14] William K Pratt, *Digital image processing.*: John Wiley & Sons, Inc, 2001.
- [15] David A Forsyth and Jean Ponce, *Computer Vision: A Modern Approach.*: Prentice Hall, 2002.
- [16] Peter E Hart, "How the Hough Transform Was Invented," 2009.
- [17] P.V.C Hough, "Method and means for recognizing complex patterns," U.S. Patent 3 069 654, Dec. 18, 1962.
- [18] Rafael Molina, *Introducción al procesamiento y análisis de imágenes digitales.*, 1998.
- [19] Nobuyuki Otsu, "A threshold selection method from gray level histograms," vol. SMC-9, no. 1, 1979.
- [20] Mehmet Sezgin and Bulent Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," vol. 13, no. 1, 2004.
- [21] Tinku Acharya and Ajoy K. Ray, *Digital image processing. Principles and applications.* New Jersey: Wiley-Interscience, 2005.

- [22] The Mathworks, Inc. Image processing toolbox - Morphology Fundamentals: Dilation and Erosion. [Online]. <http://www.mathworks.com/help/toolbox/images/f18-12508.html>
- [23] Rafael C. González, Richard E Woods, and Steven L Eddins, *Digital image processing using MATLAB*. New Jersey: Pearson Prentice Hall, 2004.
- [24] The Mathworks, Inc. Image processing toolbox - Connected component labeling: Wrapping up. [Online]. <http://blogs.mathworks.com/steve/2007/06/13/connected-component-labeling-wrapping-up/>
- [25] Manuel Blanco Álvarez, Alfredo Guerra Hernández, and Rogelio Lau Fernández, *Matemática numérica*. La Habana, Cuba: Félix Varela, 2007.
- [26] Grupo de Investigación EDMANS, *Técnicas y algoritmos básicos de Visión artificial*. La Rioja, España: Editorial de la Universidad de La Rioja, 2006, ISBN: 84-689-9345-X.
- [27] Bin Yu and Anil K Jain, "A robust and fast skew detection algorithm for generic documents," *Pattern recognition letters*.
- [28] Atallah Mahmoud Al-Shatnawi and Khairuddin Omar, "Skew detection and correction technique for arabic document images based on centre of gravity," *Journal of computer sciences*, pp. 363-369, 2009, University Kebangsaan Malaysia, Selangor, Malaysia.
- [29] P Soille, *Morphological Image Analysis: Principles and Applications*.: Springer-Verlag, 1999.
- [30] L Lam, Lee Seong-Whan, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, p. 879, septiembre 1979.
- [31] Rob J Hyndman and Anne B Koehler, "Another look at measures of forecast accuracy," 2005.
- [32] John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel, "Performance measures for information extraction".

## BIBLIOGRAFÍA

- [1] Lawrence O’Gorman and Rangachar Kasturi, *Document image analysis.*, 2009.
- [2] Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu, and Ching Y. Suen, *Character recognition systems. A guide for students and practitioners.*: Wiley Interscience, 2007.
- [3] Jonathan J. Hull, "Document image skew detection: survey and annotated bibliography," *Document analysis system*, vol. II, pp. 40-64, 1998.
- [4] Manjunath Aradhya, Hemantha Kumar, and Shivakumara P, "Skew detection technique for binary document images based on Hough Transform," *International Journal of Information Technology*, 2007.
- [5] Rajiv Kapoor, Deepak Bagai, and T.S. Kamal, "A new algorithm for skew detection and correction," *Pattern recognition letters*, pp. 1215-1229, 2004.
- [6] B. V Dhandra, V. S Malemath, Mallikarjun H, and Ravindra Hegadi, "Skew detection in binary image documents based on image dilation and region labeling approach," in *18th International Conference on Pattern Recognition (ICPR'06)*, 2006.
- [7] Mohamed El-Adawi, Hesham A. Keshk, Manal A. Ismail, and Essam M. Zaki, "Fast skew and slant correction for arabic written word or line,".
- [8] Zhixin Shi and Venu Govindaraju, "Skew detection for complex document images using fuzzy runlength," in *Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*, 2003.
- [9] Khairuddin Omar, Rahman Ramli, Ramlan Mahmod, and Nasir Sulaiman, "Skew detection and correction of jawi images using gradient direction," *Jurnal teknologi*, pp. 117-126, 2002, Universiti Teknologi Malaysia.
- [10] Rafael C. González and Richard E. Woods, *Digital image processing*, 2nd ed. New Jersey: Prentice Hall, 2002.
- [11] The Mathworks, Inc. (2010) Image Processing Toolbox - Image Types in the Toolbox. [Online]. <http://www.mathworks.com/help/toolbox/images/f14-13543.html>
- [12] Peter Toft, "The Radon Transform. Theory and implementation," 1996.
- [13] Richard O Duda and Peter E Hart, "Use of the Hough Transformation to detect lines and curves in pictures," 1972.
- [14] William K Pratt, *Digital image processing.*: John Wiley & Sons, Inc, 2001.
- [15] David A Forsyth and Jean Ponce, *Computer Vision: A Modern Approach.*: Prentice Hall, 2002.
- [16] Peter E Hart, "How the Hough Transform Was Invented," 2009.
- [17] P.V.C Hough, "Method and means for recognizing complex patterns," U.S. Patent 3 069 654, Dec. 18, 1962.
- [18] Rafael Molina, *Introducción al procesamiento y análisis de imágenes digitales.*, 1998.
- [19] Nobuyuki Otsu, "A threshold selection method from gray level histograms," vol. SMC-9, no. 1, 1979.
- [20] Mehmet Sezgin and Bulent Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," vol. 13, no. 1, 2004.
- [21] Tinku Acharya and Ajoy K. Ray, *Digital image processing. Principles and applications.* New Jersey: Wiley-Interscience, 2005.

- [22] The Mathworks, Inc. Image processing toolbox - Morphology Fundamentals: Dilation and Erosion. [Online]. <http://www.mathworks.com/help/toolbox/images/f18-12508.html>
- [23] Rafael C. González, Richard E Woods, and Steven L Eddins, *Digital image processing using MATLAB*. New Jersey: Pearson Prentice Hall, 2004.
- [24] The Mathworks, Inc. Image processing toolbox - Connected component labeling: Wrapping up. [Online]. <http://blogs.mathworks.com/steve/2007/06/13/connected-component-labeling-wrapping-up/>
- [25] Manuel Blanco Álvarez, Alfredo Guerra Hernández, and Rogelio Lau Fernández, *Matemática numérica*. La Habana, Cuba: Félix Varela, 2007.
- [26] Grupo de Investigación EDMANS, *Técnicas y algoritmos básicos de Visión artificial*. La Rioja, España: Editorial de la Universidad de La Rioja, 2006, ISBN: 84-689-9345-X.
- [27] Bin Yu and Anil K Jain, "A robust and fast skew detection algorithm for generic documents," *Pattern recognition letters*.
- [28] Atallah Mahmoud Al-Shatnawi and Khairuddin Omar, "Skew detection and correction technique for arabic document images based on centre of gravity," *Journal of computer sciences*, pp. 363-369, 2009, University Kebangsaan Malaysia, Selangor, Malaysia.
- [29] P Soille, *Morphological Image Analysis: Principles and Applications*.: Springer-Verlag, 1999.
- [30] L Lam, Lee Seong-Whan, and Ching Y. Suen, "Thinning Methodologies-A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, p. 879, septiembre 1979.
- [31] Rob J Hyndman and Anne B Koehler, "Another look at measures of forecast accuracy," 2005.
- [32] John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel, "Performance measures for information extraction".
- [33] Rolando Alfredo Hernández León and Sayda Coello González, *El proceso de investigación científica*. La Habana, Cuba: Editorial Universitaria del Ministerio de Educación Superior, 2011.
- [34] Carlos Álvarez de Zayas, *Metodología de la investigación científica*. Santiago de Cuba, Cuba, 1995.
- [35] Cecilia Polaino de los Santos and Osvaldo Balmaseda Neyra, "50 recomendaciones para la redacción del diseño e informe de la investigación," Programa Especial de Formación Científica en Informática, Universidad de las Ciencias Informáticas, La Habana, 2011.
- [36] Pedro Febles Rodríguez, Cecilia Polaino de los Santos, and Vivian Estrada Sentí, "Algunas recomendaciones para la escritura de una tesis," Universidad de las Ciencias Informáticas, La Habana, 2011.
- [37] Evapostgrado, "Problemas frecuentes al redactar la introducción de una tesis," Universidad de las Ciencias Informáticas, La Habana, 2011.
- [38] Robert A. Day, *How to write and publish a scientific paper*, Quinta edición ed. Phoenix, Arizona, United States of America, 1998.

## ANEXOS

### Anexo 1. Resultados retornados por el algoritmo propuesto

En este anexo se muestran (detalladamente) los resultados retornados por el algoritmo propuesto, al procesar las 11600 imágenes existentes en el banco de prueba.

**Tabla A1.1.** Resultados generados por el algoritmo propuesto (Vista: Ángulos y tipos de texto).

Áng	Texto escrito a máquina						Texto manuscrito					
	Fallos		Aciertos		MAE	MdA E	Fallos		Aciertos		MAE	MdA E
	$\bar{\pi}1^0$	Otro	%	% ext			$\bar{\pi}1^0$	Otro	%	% ext		
-5	11	0	94,5	100	0,01	0	0	0	100	100	0	0
-10	13	0	93,5	100	0,07	0	0	0	100	100	0	0
-15	14	0	93	100	0,07	0	0	0	100	100	0	0
-20	14	0	93	100	0,07	0	0	0	100	100	0	0
-25	23	0	88,5	100	0,12	0	0	0	100	100	0	0
-30	21	1	89	99,5	0,12	0	0	0	100	100	0	0
-36	28	1	85,5	99,5	0,15	0	0	0	100	100	0	0
-42	33	3	82	98,5	0,21	0	1	0	99,5	100	0,01	0
-48	31	1	84	99,5	0,18	0	1	0	99,5	100	0,01	0
-60	18	0	91	100	0,09	0	0	0	100	100	0	0
-72	15	0	92,5	100	0,08	0	0	0	100	100	0	0
-75	13	0	93,5	100	0,07	0	0	0	100	100	0	0
-80	9	0	95,5	100	0,06	0	0	0	100	100	0	0
-85	6	0	97	100	0,03	0	0	0	100	100	0	0
0	10	0	95	100	0,05	0	0	0	100	100	0	0
5	8	2	95	99	0,06	0	0	0	100	100	0	0
10	8	0	96	100	0,04	0	0	0	100	100	0	0
15	13	0	93,5	100	0,07	0	0	0	100	100	0	0
20	20	1	89,5	99,5	0,11	0	0	0	100	100	0	0
25	20	2	89	99	0,13	0	0	0	100	100	0	0
30	17	1	91	99,5	0,1	0	0	0	100	100	0	0
36	29	1	85	99,5	0,16	0	0	0	100	100	0	0
42	36	0	82	100	0,18	0	0	0	100	100	0	0
48	35	4	80,5	98	0,23	0	0	0	100	100	0	0
60	20	0	90	100	0,1	0	0	0	100	100	0	0
72	17	0	91,5	100	0,09	0	0	0	100	100	0	0
75	13	0	93,5	100	0,07	0	0	0	100	100	0	0
80	9	0	95,5	100	0,05	0	0	0	100	100	0	0

85	8	0	96	100	0,04	0	0	0	100	100	0	0
<b>%</b>	90,88						99,97					
<b>% ext</b>	99,71						100					

**Tabla A1.2.** Por cientos de aciertos obtenidos mediante el algoritmo propuesto (Vista: Subcategorías).

<b>Subcategoría</b>	<b>Texto manuscrito</b>		<b>Texto escrito a máquina</b>	
	<b>% aciertos</b>	<b>% aciertos (ext)</b>	<b>% aciertos</b>	<b>% aciertos (ext)</b>
1	100	100	96,48	99,93
2	99,93	100	87,03	100
3	100	100	90,34	99,52
4	99,93	100	89,66	99,38



## Anexo 2. Resultados retornados por otros algoritmos.

En este anexo se muestran los resultados retornados por algoritmos precedentes, al procesar las imágenes en el banco de prueba. En total se muestran los resultados retornados por seis algoritmos. El primero está basado en el análisis clásico de la Transformada Radon (RT). El segundo está basado en el análisis de proyecciones. El tercero combina el análisis de la RT con el análisis de las componentes conectadas. En el cuarto se hace un análisis multiresolución de la RT. En el quinto se preprocesa la imagen para eliminar algunos objetos, y luego se analiza la RT. En el sexto también se preprocesa la imagen, pero la estimación se hace analizando las componentes conectadas. Los seis algoritmos fueron implementados utilizando la herramienta MATLAB 2008b.

No se implementaron algoritmos basados en aproximación de funciones ya que los mismos son extremadamente sensibles ante la existencia de bordes laterales en la imagen de entrada, y prácticamente todas las imágenes en el banco de prueba muestran bordes laterales. Por tanto se sabía a priori que los resultados serían malos. Igualmente no se implementaron métodos basados en el gradiente ni en máscaras ya que son poco usados, y sobre todo, son muy sensibles ante la existencia de bordes laterales, bordes de página, figuras, y puntos ruidosos.

### A2.1. Algoritmo basado en el análisis clásico de Transformada Radon

En este algoritmo se segmentan los puntos de la imagen en dos clases, se establece el color del fondo a negro, se calcula la RT, y se analiza la misma del modo clásico ya discutido en el Capítulo 1.

Este algoritmo extiende al propuesto en [1]. La extensión consiste básicamente en la segmentación de la imagen y en la normalización del color del fondo. Fue necesario hacer esta extensión para que el algoritmo propuesto en [1] fuese capaz de procesar las imágenes en escala de grises y en colores, existentes en el banco de prueba. Los resultados arrojados por el algoritmo se muestran en la Tabla A2.1.

**Tabla A2.1.** Resultados generados por el algoritmo basado en el análisis clásico de la RT.

Áng	Texto escrito a máquina						Texto manuscrito					
	Fallos		Aciertos		MAE	MdAE	Fallos		Aciertos		MAE	MdAE
	±1 <sup>0</sup>	Otro	%	% ext			±1 <sup>0</sup>	Otro	%	% ext		
-5	0	200	0	0	3,89	4	0	200	0	0	95	95
-10	0	200	0	0	7,49	8	0	200	0	0	100	100
-15	0	200	0	0	12,43	13	0	200	0	0	104	105
-20	0	200	0	0	16,71	17	0	200	0	0	108	110
-25	0	200	0	0	22,06	22	0	200	0	0	113	115
-30	0	200	0	0	26,57	27	0	200	0	0	117	120
-36	1	199	0	0,5	32,26	33	0	200	0	0	121	126
-42	0	197	1,5	1,5	38,31	39	4	190	3	5	120	132
-48	1	198	0,5	1	44,1	39	4	192	2	4	44,3	48
-60	0	200	0	0	33,78	27	0	200	0	0	59,4	60
-72	0	200	0	0	23,47	15	0	200	0	0	71,8	72

-75	0	200	0	0	21,43	13	0	200	0	0	74,8	75
-80	0	200	0	0	17,1	8	0	200	0	0	80	80
-85	0	200	0	0	16,64	4	0	200	0	0	85	85
0	1	194	2,5	3	83,29	87	0	200	0	0	90	90
5	0	200	0	0	3,89	4	0	200	0	0	85	85
10	0	200	0	0	7,5	8	0	200	0	0	80	80
15	0	200	0	0	12,43	13	0	200	0	0	74,8	75
20	0	200	0	0	16,72	17	0	200	0	0	69,8	70
25	0	200	0	0	22	22	0	200	0	0	64,58	65
30	0	200	0	0	26,58	27	0	200	0	0	59,36	60
36	0	200	0	0	32,49	33	0	200	0	0	52,19	54
42	0	198	1	1	38,75	39	4	192	2	4	44,07	48
48	1	197	1	1,5	38,84	39	4	190	3	5	44,06	48
60	0	200	0	0	26,57	27	0	200	0	0	59,16	60
72	0	200	0	0	14,82	15	0	200	0	0	71,6	72
75	0	200	0	0	12,43	13	0	200	0	0	74,58	75
80	0	200	0	0	7,48	8	0	200	0	0	80	80
85	0	200	0	0	3,89	4	0	200	0	0	85	85
%	0,22						0,34					
% ext	0,29						0,62					

## A2.2. Algoritmo basado en el análisis de proyecciones

En este algoritmo se segmentan los puntos de la imagen en dos clases, se establece el color del fondo a negro, se proyecta la imagen en varios ángulos, se calcula la varianza de cada proyección, y se selecciona la proyección que arroja la mayor varianza. El ángulo asociado a la proyección seleccionada, es tomado como el ángulo de la inclinación del documento.

Este algoritmo extiende al propuesto en [2]. La extensión consiste básicamente en la segmentación de la imagen y en la normalización del color del fondo. Fue necesario hacer esta extensión para que el algoritmo propuesto en [2] fuese capaz de procesar las imágenes en escala de grises y en colores, existentes en el banco de prueba. Los resultados arrojados por el algoritmo se muestran en la Tabla A2.2.

**Tabla A2.2.** Resultados generados por el algoritmo basado en el análisis de proyecciones.

Áng	Texto escrito a máquina						Texto manuscrito					
	Fallos		Aciertos		MAE	MdA E	Fallos		Aciertos		MAE	MdA E
	±1 <sup>0</sup>	Otro	%	% ext			±1 <sup>0</sup>	Otro	%	% ext		
-5	0	200	0	0	78,8	95	0	200	0	0	95	95
-10	0	200	0	0	11,35	10	0	200	0	0	100	100

-15	0	200	0	0	15	15	0	200	0	0	105	105
-20	0	200	0	0	20	20	0	200	0	0	110	110
-25	0	200	0	0	25	25	0	200	0	0	115	115
-30	0	200	0	0	30	30	0	200	0	0	120	120
-36	0	200	0	0	36	36	0	200	0	0	126	126
-42	0	200	0	0	43,8	42	0	200	0	0	128	132
-48	0	200	0	0	137,1	138	0	200	0	0	49,35	48
-60	0	200	0	0	150	150	0	200	0	0	60	60
-72	0	200	0	0	162	162	0	200	0	0	72	72
-75	0	200	0	0	165	165	0	200	0	0	75	75
-80	0	200	0	0	168	170	0	200	0	0	80	80
-85	0	200	0	0	99,85	85	0	200	0	0	85	85
0	0	185	7,5	7,5	83,25	90	0	143	28,5	28,5	63,36	90
5	0	200	0	0	71,8	85	0	200	0	0	85	85
10	0	200	0	0	11,05	10	0	200	0	0	80	80
15	0	200	0	0	15	15	0	200	0	0	75	75
20	0	200	0	0	20	20	0	200	0	0	70	70
25	0	200	0	0	25	25	0	200	0	0	65	65
30	0	200	0	0	30	30	0	200	0	0	60	60
36	0	200	0	0	36	36	0	200	0	0	54	54
42	0	200	0	0	42,06	42	0	200	0	0	47,91	48
48	0	200	0	0	42,12	42	0	200	0	0	47,73	48
60	0	200	0	0	30	30	0	200	0	0	60	60
72	0	200	0	0	18	18	0	200	0	0	72	72
75	0	200	0	0	15	15	0	200	0	0	75	75
80	0	200	0	0	11,05	10	0	200	0	0	80	80
85	0	200	0	0	70,6	85	0	200	0	0	85	85
%	0,23						0,98					
% ext	0,26						0,98					

### A2.3. Algoritmo basado en el análisis de las componentes conectadas y de la Transformada Radon

En este algoritmo se segmentan los puntos de la imagen en dos clases, se establece el color del fondo a negro, se hallan las componentes conectadas, y se calcula el centroide de cada una. Luego se genera una imagen binaria que tiene valor TRUE solamente en la ubicación de los centroides calculados. Esta imagen binaria es tomada como entrada para una estimación basada en el análisis clásico de la RT.

Este algoritmo extiende al propuesto en [27]. La extensión consiste básicamente en la segmentación de la imagen y en la normalización del color del fondo. Fue necesario hacer esta extensión para que el

algoritmo propuesto en [27] fuese capaz de procesar las imágenes en escala de grises y en colores, existentes en el banco de prueba. Se diferencia además en que el análisis de la RT se hace de modo clásico, y no del modo multiresolución propuesto en [27] y por otros autores. No se utilizó el modo multiresolución para evadir las desventajas del mismo. Los resultados arrojados por el algoritmo se muestran en la Tabla A2.3.

**Tabla A2.3.** Resultados generados por el algoritmo basado en el análisis de las componentes conectadas y de la RT.

Áng	Texto escrito a máquina						Texto manuscrito					
	Fallos		Aciertos		MAE	MdAE	Fallos		Aciertos		MAE	MdAE
	±1 <sup>0</sup>	Otro	%	% ext			±1 <sup>0</sup>	Otro	%	% ext		
-5	3	2	97,5	99	0,92	0	8	14	89	93	5,03	0
-10	5	1	97	99,5	0,48	0	6	18	88	91	7,19	0
-15	5	2	96,5	99	0,93	0	11	18	85,5	91	6,85	0
-20	5	3	96	98,5	1,38	0	12	16	86	92	6,26	0
-25	4	2	97	99	0,92	0	10	20	85	90	6,8	0
-30	5	3	96	98,5	1,38	0	7	42	75,5	79	19,12	0
-36	5	2	96,5	99	0,93	0	2	74	62	63	32,92	0
-42	5	2	96,5	99	0,93	0	15	14	85,5	93	5,64	0
-48	4	3	96,5	98,5	1,37	0	17	16	83,5	92	6,41	0
-60	6	3	95,5	98,5	1,38	0	12	13	87,5	93,5	5,75	0
-72	4	1	97,4	99,5	0,47	0	5	17	89	91,5	7,72	0
-75	6	2	96	99	0,93	0	5	18	88,5	91	8,47	0
-80	4	1	97,5	99,5	0,47	0	10	17	86,5	91,5	8,24	0
-85	4	1	97,5	99,5	0,47	0	9	15	88	92,5	8,01	0
0	3	0	98,5	100	0,02	0	4	20	88	90	7,7	0
5	2	1	98,5	99,5	0,46	0	11	15	87	92,5	5,59	0
10	2	3	97,5	98,5	1,36	0	7	19	87	90,5	7,7	0
15	4	2	97	99	0,92	0	8	22	85	89	8,64	0
20	6	2	96	99	0,93	0	5	26	84,5	87	10,44	0
25	5	3	96	98,5	1,38	0	10	14	88	93	5,71	0
30	5	2	96,5	99	0,93	0	13	16	85,5	92	6,4	0
36	7	3	95	98,5	1,39	0	6	114	40	43	50,59	90
42	8	3	94,5	98,5	1,39	0	14	11	87,5	94,5	4,72	0
48	2	3	97,5	98,5	1,36	0	12	12	88	94	4,23	0
60	4	1	97,5	99,5	0,47	0	7	36	78,5	82	15,68	0
72	3	1	98	99,5	0,47	0	10	13	88,5	93,5	5,34	0
75	3	3	97	98,5	1,37	0	8	16	88	92	5,87	0

80	2	1	98,5	99,5	0,46	0	9	15	88	92,5	6,58	0
85	4	1	97,5	99,5	0,47	0	11	14	87,5	93	5,73	0
%	96,86						83,81					
% ext	99,02						88,36					

#### A2.4. Algoritmo basado en el análisis multiresolución de la Transformada Radon

En este algoritmo se segmentan los puntos de la imagen en dos clases, se establece el color del fondo a negro, se calcula la RT utilizando una separación angular  $d_1$ , y se toma el ángulo  $\theta_1$  asociado al máximo valor en la RT. Luego se calcula nuevamente la RT, pero utilizando una separación angular  $d_2$  ( $d_2 \ll d_1$ ) y en la vecindad del ángulo  $\theta_1$ . Finalmente se hace un análisis clásico de la RT que se calculó utilizando la separación angular  $d_2$ .

Este algoritmo extiende al propuesto en [3]. La extensión consiste básicamente en la segmentación de la imagen y en la normalización del color del fondo. Fue necesario hacer esta extensión para que el algoritmo propuesto en [3] fuese capaz de procesar las imágenes en escala de grises y en colores, existentes en el banco de prueba. Los resultados arrojados por el algoritmo se muestran en la Tabla A2.4. En la implementación que se hizo el valor de  $d_1$  fue 10 grados y el valor de  $d_2$  fue un grado. El tamaño de vecindad que se utilizó fue 15 grados, a ambos lados de  $\theta_1$ .

Tabla A2.4. Resultados generados por el algoritmo basado en el análisis multiresolución de la RT.

Áng	Texto escrito a máquina						Texto manuscrito					
	Fallos		Aciertos		MAE	MdAE	Fallos		Aciertos		MAE	MdAE
	$\mp 1^0$	Otro	%	% ext			$\mp 1^0$	Otro	%	% ext		
-5	0	200	0	0	3,89	4	0	200	0	0	85	85
-10	0	200	0	0	7,49	8	0	200	0	0	80	80
-15	0	200	0	0	12,43	13	0	200	0	0	74,58	75
-20	0	200	0	0	16,71	17	0	200	0	0	69,36	70
-25	0	200	0	0	22,06	22	0	200	0	0	64,18	65
-30	0	200	0	0	26,67	27	0	200	0	0	59,16	60
-36	1	199	0	0,5	32,51	33	0	200	0	0	52,68	54
-42	0	199	0,5	0,5	38,47	39	1	197	1	1,5	45,95	48
-48	1	198	0,5	1	38,29	39	4	194	1	3	45,68	48
-60	0	200	0	0	26,68	27	0	200	0	0	59,79	60
-72	0	200	0	0	14,83	15	0	200	0	0	72	72
-75	0	200	0	0	12,43	13	0	200	0	0	75	75
-80	0	200	0	0	7,5	8	0	200	0	0	80	80
-85	0	200	0	0	3,89	4	0	200	0	0	85	85
0	2	188	5	6	80,75	86	0	189	5,5	5,5	83,79	90
5	0	200	0	0	3,89	4	0	200	0	0	95	95

10	0	200	0	0	7,5	8	0	200	0	0	100	100
15	0	200	0	0	12,43	13	0	200	0	0	105	105
20	0	200	0	0	16,72	17	0	200	0	0	109	110
25	0	200	0	0	22,07	22	0	200	0	0	114	115
30	0	200	0	0	26,68	27	0	200	0	0	119	120
36	0	200	0	0	32,49	33	0	200	0	0	123	126
42	0	198	1	1	38,29	39	4	194	1	3	125	132
48	1	199	0	0,5	140	141	1	197	1	1,5	45,95	48
60	0	200	0	0	153	153	0	200	0	0	59,16	60
72	0	200	0	0	165	165	0	200	0	0	71,39	72
75	0	200	0	0	167	167	0	200	0	0	74,58	75
80	0	200	0	0	171	172	0	200	0	0	80	80
85	0	200	0	0	176	176	0	200	0	0	85	85
%	0,24						0,33					
% ext	0,33						0,5					

## A2.5. Algoritmo basado en el preprocesamiento de la imagen y en el análisis de la Transformada Radon

En este algoritmo se segmentan los puntos de la imagen en dos clases, se establece el color del fondo a negro, se encuentran las componentes conectadas en la imagen, se calcula la altura promedio de las componentes encontradas, se eliminan aquellas cuya altura sea mayor o igual que la altura promedio, se aplica una dilatación utilizando una estructura rectangular de tamaño fijo, y se aplica un algoritmo morfológico de adelgazamiento. Luego se encuentran nuevamente las componentes conectadas (en la imagen resultante del adelgazamiento), se eliminan aquellas que son más altas que anchas, se calcula la RT de la imagen resultante, y se analiza dicha RT del modo clásico ya discutido en el Capítulo 1.

Este algoritmo extiende al propuesto en [4]. La extensión consiste básicamente en la segmentación de la imagen y en la normalización del color del fondo. Fue necesario hacer esta extensión para que el algoritmo propuesto en [4] fuese capaz de procesar las imágenes en escala de grises y en colores, existentes en el banco de prueba. Los resultados arrojados por el algoritmo se muestran en la Tabla A2.5. En la implementación, el rectángulo empleado en la dilatación tuvo un tamaño de 4 filas por tres columnas.

**Tabla A2.5.** Resultados generados por el algoritmo basado en el preprocesamiento de la imagen y en el análisis de la RT.

Áng	Texto escrito a máquina						Texto manuscrito					
	Fallos		Aciertos		MAE	MdA E	Fallos		Aciertos		MAE	MdA E
	$\mp 1^0$	Otro	%	% ext			$\mp 1^0$	Otro	%	% ext		
-5	23	32	72,5	84	3,74	0	21	27	76	86,5	2,23	0
-10	19	33	74	83,5	3,23	0	22	20	79	90	2,23	0

-15	20	34	73	83	4,42	0	24	24	76	88	1,94	0
-20	25	29	73	85,5	2,72	0	19	44	68,5	78	6	0
-25	24	25	75,5	87,5	2,12	0	19	40	70,5	80	5,09	0
-30	26	25	74,5	87,5	3,1	0	30	36	67	82	3,47	0
-36	27	27	73	86,5	3,07	0	22	41	68,5	79,5	5,01	0
-42	29	31	70	84,5	5,06	0	29	50	60,5	75	5,82	0
-48	25	44	65,5	78	9,21	0	31	68	50,5	66	16,03	0
-60	15	94	45,5	53	33,85	1	6	152	21	24	56,5	78,5
-72	7	119	37	40,5	47,4	49,5	0	193	3,5	3,5	76,55	90
-75	8	125	33,5	37,5	50,69	67,5	4	187	4,5	6,5	79,31	90
-80	11	126	31,5	37	57,09	87	3	193	2	3,5	81,31	90
-85	7	128	32,5	36	56,63	82	0	197	1,5	1,5	84,19	90
0	9	12	89,5	94	1,69	0	13	25	81	87,5	3,76	0
5	22	28	75	86	3,25	0	31	25	72	87,5	2,75	0
10	20	26	77	87	3,39	0	17	23	80	88,5	2,24	0
15	19	20	80,5	90	2,02	0	19	31	75	84,5	1,63	0
20	17	24	79,5	88	3,07	0	23	37	70	81,5	5,35	0
25	18	16	83	92	1,76	0	24	36	70	82	4,3	0
30	21	18	80,5	91	1,72	0	29	40	65,5	80	3,3	0
36	14	20	83	90	1,14	0	35	36	64,5	82	3,25	0
42	19	13	84	93,5	1,41	0	49	43	54	78,5	3,95	0
48	20	32	74	84	5,1	0	27	90	41,5	55	15,49	1
60	7	70	61,5	65	22,1	0	11	172	8,5	14	59,75	89,5
72	10	110	40	45	40,06	27,5	2	190	4	5	81,31	90
75	14	108	39	46	40,05	24,5	1	193	3	3,5	82,63	90
80	7	122	35,5	39	49,55	61	2	198	0	1	91,66	90
85	4	128	34	36	54,53	78,5	1	197	1	1,5	92,05	90
%	63,69						46,17					
% ext	72,09						55,03					

## A2.6. Algoritmo basado en el preprocesamiento de la imagen y en el análisis de las componentes conectadas

En este algoritmo se segmentan los puntos de la imagen en dos clases, se establece el color del fondo a negro, se encuentran las componentes conectadas y se eliminan las más pequeñas, se aplica una dilatación morfológica utilizando una línea horizontal de longitud fija, se llenan los huecos, se encuentran nuevamente las componentes conectadas, y se estima la inclinación local de cada de ellas. La inclinación local de una componente dada es asumida como el ángulo del eje mayor de la elipse que la encierra. La inclinación global es calculada promediando el conjunto de inclinaciones locales.

Este algoritmo extiende al propuesto en [6]. La extensión consiste básicamente en la segmentación de la imagen y en la normalización del color del fondo. Fue necesario hacer esta extensión para que el algoritmo propuesto en [6] fuese capaz de procesar las imágenes en escala de grises y en colores, existentes en el banco de prueba. Los resultados arrojados por el algoritmo se muestran en la Tabla A2.6. En la implementación, se asumieron como componentes “pequeñas” a aquellas cuya área fuese menor que un umbral. Para hallar dicho umbral se calculó la mediana del conjunto dado por el área de cada componente en la imagen, y se dividió la mediana entre dos. La longitud de la línea fue de cinco puntos.

**Tabla A2.6.** Resultados generados por el algoritmo basado en el preprocesamiento de la imagen y en el análisis de las componentes conectadas.

Áng	Texto escrito a máquina						Texto manuscrito					
	Fallos		Aciertos		MAE	MdA E	Fallos		Aciertos		MAE	MdAE
	$\mp 1^0$	Otro	%	% ext			$\mp 1^0$	Otro	%	% ext		
-5	0	200	0	0	5	5	0	200	0	0	93,24	95
-10	0	200	0	0	10	10	0	200	0	0	98,15	100
-15	0	200	0	0	15	15	0	200	0	0	29,94	9,96
-20	0	200	0	0	19,37	20	0	200	0	0	31,67	11,51
-25	0	200	0	0	24,92	25	0	200	0	0	115	115
-30	0	200	0	0	29,12	30	0	200	0	0	117	120
-36	0	200	0	0	36	36	0	200	0	0	125	126
-42	0	200	0	0	41,85	42	0	200	0	0	39,21	20,97
-48	0	200	0	0	130	138	0	200	0	0	48	48
-60	0	200	0	0	149	150	0	200	0	0	60,09	60
-72	0	200	0	0	161	162	0	200	0	0	72,01	72
-75	0	200	0	0	165	165	0	200	0	0	53,2	41,48
-80	0	200	0	0	169	170	0	200	0	0	57,46	47,03
-85	0	200	0	0	174	175	0	200	0	0	84,3	85
0	0	83	0	0	0,97	0,76	0	200	0	0	89,69	89,99
5	0	200	0	0	4,99	5	0	200	0	0	83,49	85
10	0	200	0	0	10	10	0	200	0	0	78,65	80
15	0	200	0	0	15	15	0	200	0	0	29,13	9,89
20	0	200	0	0	19,5	20	0	200	0	0	30,82	12,51
25	0	200	0	0	24,92	25	0	200	0	0	65	65
30	0	200	0	0	29,26	30	0	200	0	0	58,89	60
36	0	200	0	0	36	36	0	200	0	0	54,04	54
42	0	200	0	0	41,87	42	0	200	0	0	36,9	21,89
48	0	200	0	0	41,4	42	0	200	0	0	48	48
60	0	200	0	0	29,98	30	0	200	0	0	60,09	60



72	0	200	0	0	18,26	18	0	200	0	0	72,01	72
75	0	200	0	0	15	15	0	200	0	0	54,48	49,91
80	0	200	0	0	10,35	10	0	200	0	0	73,52	74,92
85	0	200	0	0	5,33	5	0	200	0	0	84,32	85
<b>%</b>	0						0					
<b>% ext</b>	0						0					

### Anexo 3. Evolución del algoritmo propuesto

En este anexo se demuestra la evolución del algoritmo propuesto, para corregir la inclinación en una imagen determinada. En la figura A3.1 se muestra la imagen original, inclinada 85 grados. En la figura A3.2 se muestra la imagen mejorada. En la figura A3.3 se muestra la imagen retornada por la implementación de la operación Segmentar en el algoritmo. La figura A3.4 muestra la imagen preprocesada. Las figuras A3.5 y A3.6 están relacionadas con la operación Estimar. En la primera se encierran en rectángulos rojos las regiones significativas seleccionadas. En la segunda se muestra el resultado del preprocesamiento adicional en cada región significativa seleccionada, y la inclinación local de cada una de ellas. En la figura A3.7 se muestra la imagen resultante del algoritmo propuesto en esta tesis.

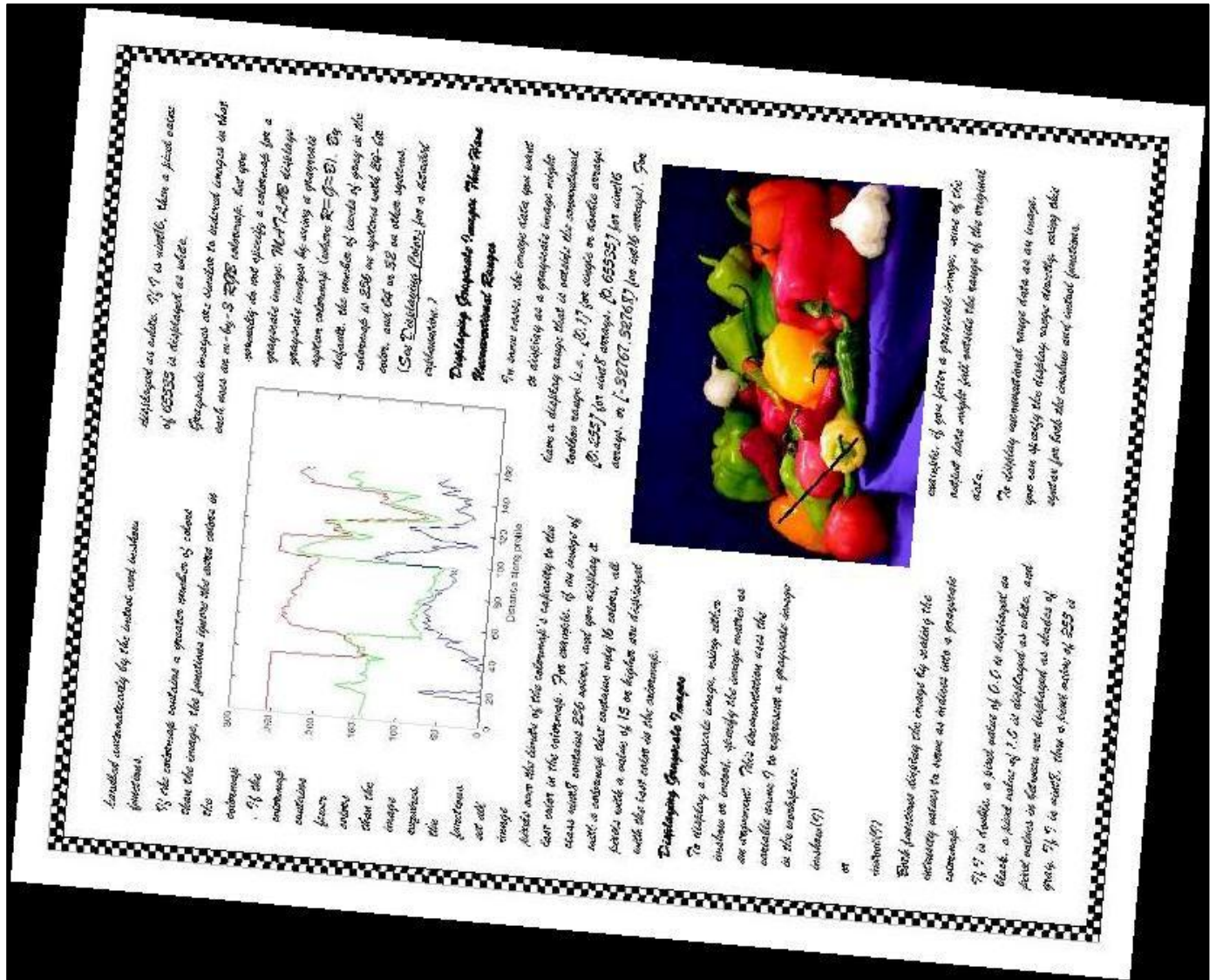
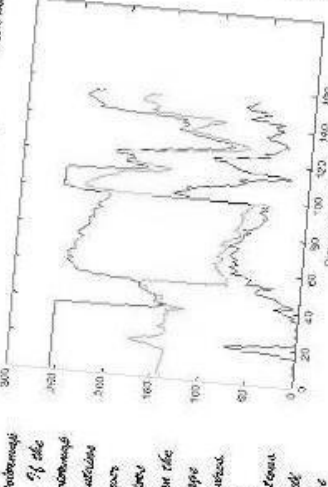


Figura A3.1. Imagen de un documento, inclinada 85 grados. La imagen muestra texto manuscrito dispuesto en dos columnas de manera irregular. Además contiene figuras, gráficos, bordes de página, y bordes laterales.

contaminated by the ambient and ambient functions.

The contour contains a greater number of values than the image, the functions ignore the areas where the contours are.



displayed as white. If  $T$  is white, then a fixed value of  $0.5535$  is displayed as white. Grayscale images are similar to binary images in that each area is either  $S$  (255) or  $B$  (0). The gray levels are not strictly a continuum for a grayscale image. MATLAB displays grayscale images by using a grayscale lookup table (where  $R=G=B$ ). By default, the number of levels is gray in the colormap is 256 or systems with 24-bit color, and 64 or 32 on other systems. (See [Displaying Colors](#) for a detailed explanation.)

### Displaying Grayscale Images Their True Numerical Range

In some cases, the image data you want to display as a grayscale image might have a display range that is not the conventional range [0, 255] for each or double range [0, 65535] for each image. For example, in [1, 5127, 5127] for each image.



For example, if you have a grayscale image, some of the pixel data might fall outside the range of the original image. To display numerical range data as an image, you can specify the display range directly using the syntax for both the `imshow` and `imwrite` functions.

Areas near the limits of the colormap's capacity to the color in the image. For example, if an image of each pixel contains 255 values, and you display it with a colormap that contains only 16 colors, all with the last color as the dominant.

### Displaying Grayscale Images

To display a grayscale image using either `imshow` or `imwrite`, specify the image matrix as an argument. This documentation uses the variable name  $I$  to represent a grayscale image of the workspace.

`imshow(I)`  
or  
`imwrite(I)`

Both functions display the image by setting the display range to have an interval into a grayscale colormap.

If  $I$  is double, a fixed value of 0 is displayed as black, a fixed value of 1 is displayed as white, and pixel values in between are displayed as shades of gray. If  $I$  is uint8, then a fixed value of 255 is

Figura A3.2. Resultado obtenido al mejorar la imagen de entrada.



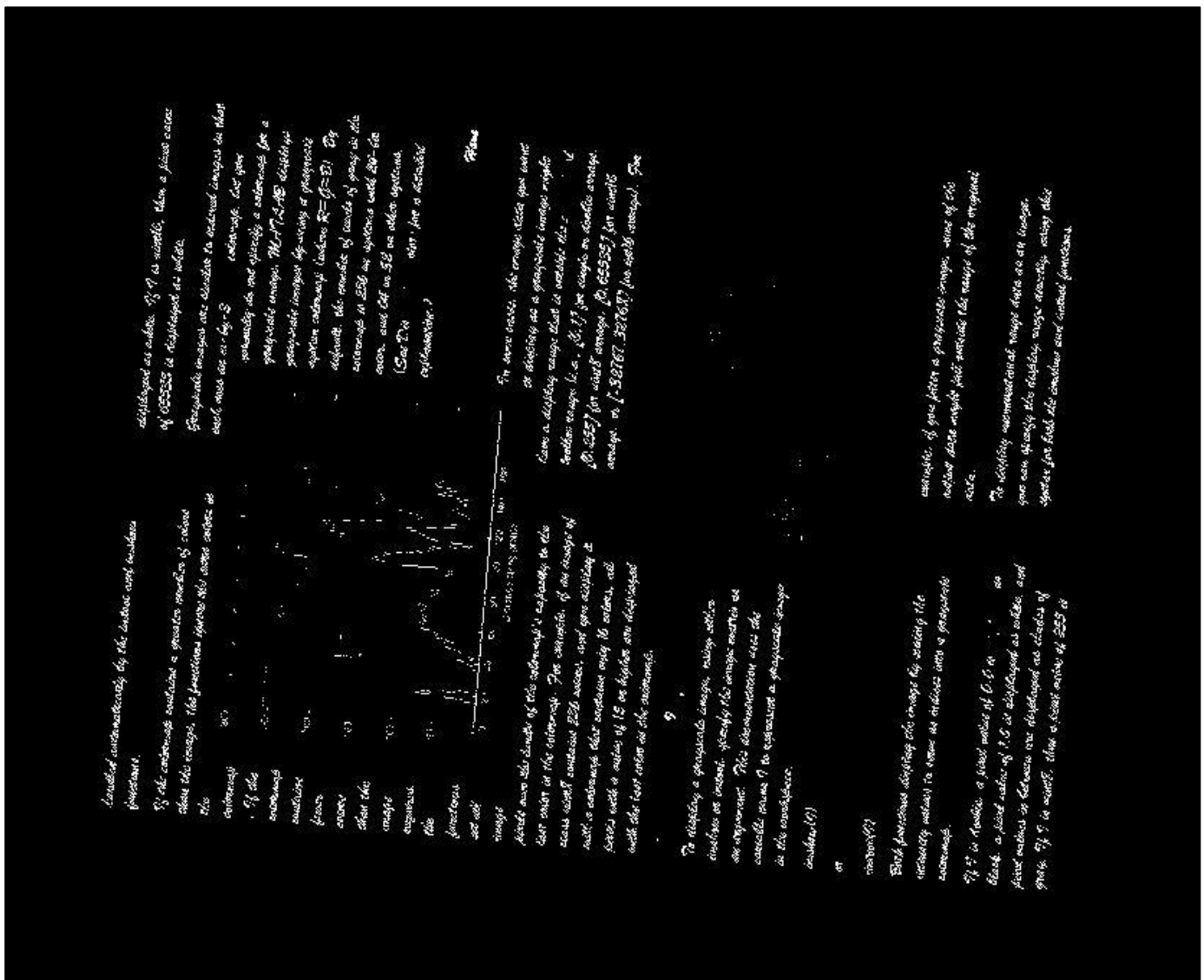
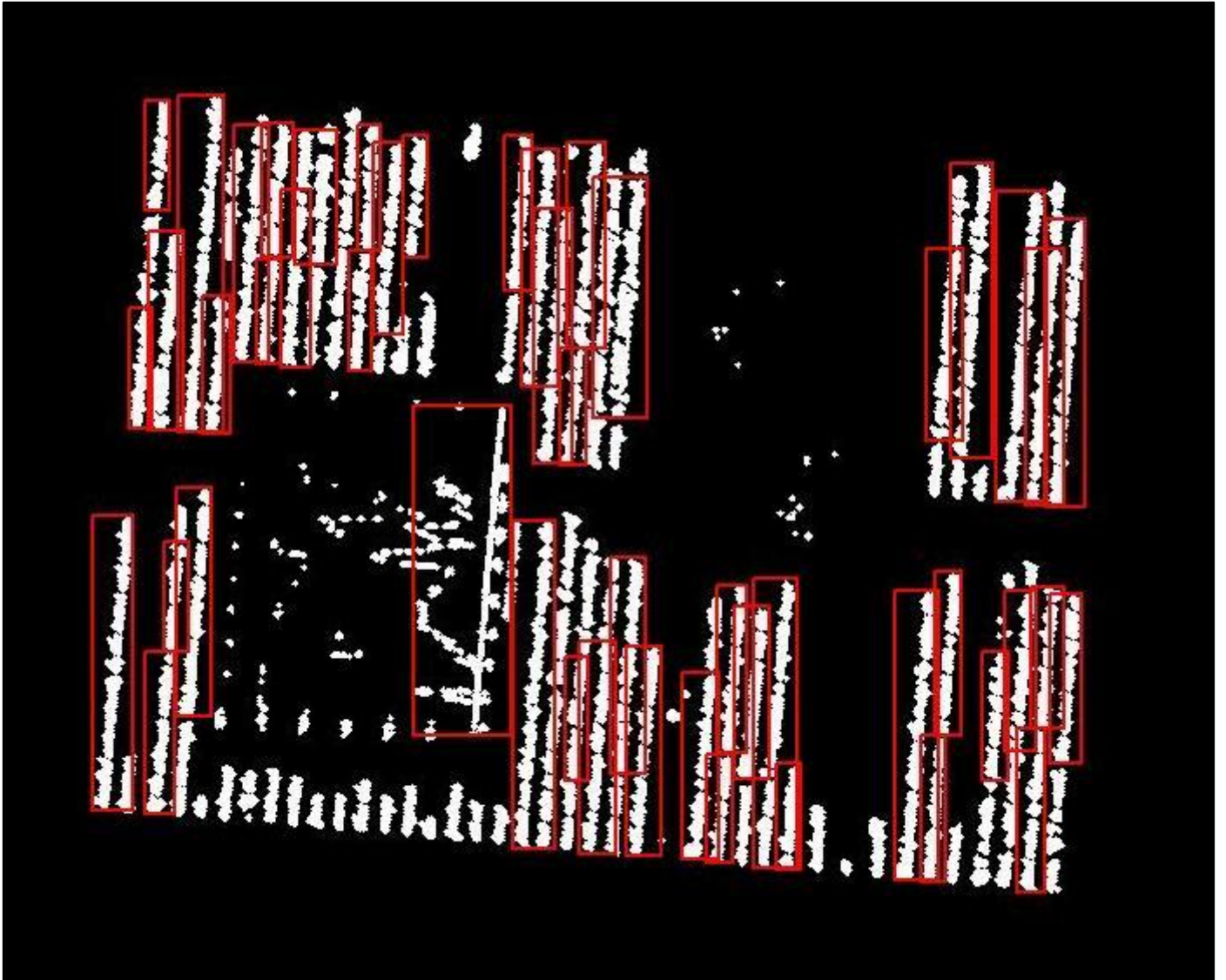
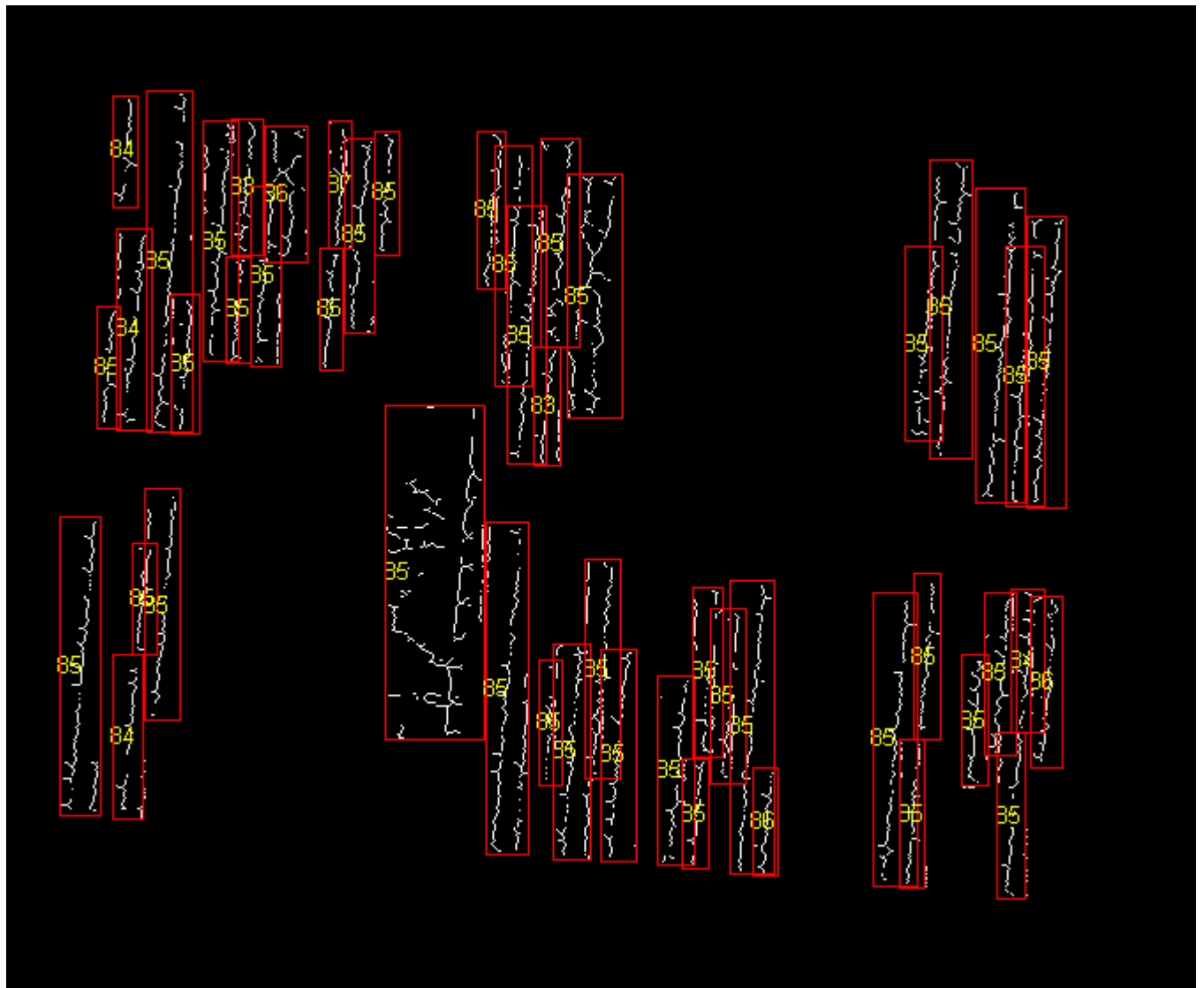


Figura A3.4. Resultado obtenido al preprocesar la imagen segmentada



**Figura A3.5.** Resultado obtenido al seleccionar las regiones significativas. Cada región se encierra en su MBR, dibujado en rojo. Se seleccionaron 49 componentes conectadas como regiones significativas, de las 191 componentes conectadas existentes.



**Figura A3.6.** Resultado obtenido al procesar las regiones significativas, y calcular la inclinación local en cada una de ellas. En dos regiones la inclinación fue  $83^{\circ}$ , en cuatro regiones la inclinación fue  $84^{\circ}$ , en 39 regiones la inclinación fue  $85^{\circ}$ , en tres regiones la inclinación fue  $86^{\circ}$ , y en una región la inclinación fue  $87^{\circ}$ .

rounded automatically by the integer part function function.

If the colormap contains a greater number of colors than the image, the function ignores the extra colors in the colormap.

If the colormap contains fewer colors than the image requires, the function uses all available colors up to the limit of the colormap's capacity to the last color in the colormap. For example, if an image of size 100x100 contains 256 colors, and you display it with a colormap that contains only 16 colors, all pixels with a value of 15 or higher are displayed with the last color in the colormap.

**Displaying Grayscale Images**

To display a grayscale image, using either imshow or imagesc, specify the image matrix as an argument. This documentation uses the variable name I to represent a grayscale image in the examples.

imshow(I)

or

imagesc(I)

Both functions display the image by scaling the intensity values to range as indices into a grayscale colormap.

If I is double, a pixel value of 0.0 is displayed as black, a pixel value of 1.0 is displayed as white, and pixel values in between are displayed as shades of gray. If I is uint8, then a pixel value of 255 is displayed as white. If I is uint16, then a pixel value of 65535 is displayed as white.

Grayscale images are similar to indexed images in that each uses an m-by-n RGB colormap, but you normally do not specify a colormap for a grayscale image. MATLAB displays grayscale images by using a grayscale system colormap (which is G=256). By default, the number of levels of gray in the colormap is 256 on systems with 24-bit color, and 64 on 32-bit or other systems. (See [Displaying Images](#) for a detailed explanation.)

**Displaying Grayscale Images That Have Nonstandard Ranges**

In some cases, the image data you want to display as a grayscale image might have a display range that is outside the conventional toolbar range (i.e., [0,1]) for single or double arrays, [0,255] for uint8 arrays, [0,65535] for uint16 arrays, or [-32767,32767] for int16 arrays. For

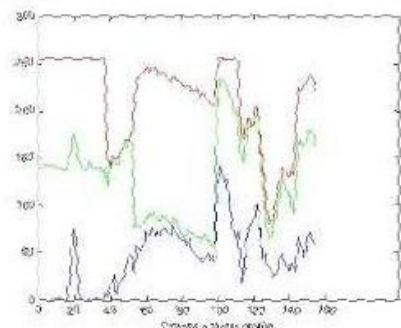


Figura A3.7. Resultado obtenido al rotar la imagen original para corregir la inclinación de 85°, estimada por el algoritmo propuesto en esta investigación.



## Anexo 4. Codificación en MATLAB del algoritmo propuesto

En este anexo se presenta el código fuente obtenido al implementar en MATLAB el algoritmo propuesto. En el código se obvian algunas validaciones obligatorias en entornos no académicos, para así facilitar la claridad y comprensión del mismo.

### A4.1. main.m

```
function[output, skewangle]= main(IMG)
% MAIN Corrects the skew of a document image.
%
% USAGE: [OUTPUT, SKEWANGLE]= MAIN(IMG)
% -----
% Corrects the skew of the document image IMG and stores the deskewed image
% in the image OUTPUT. The output argument SKEWANGLE stores the value
% of the skew corrected from the image IMG.
%
% - IMG is the image to be skew corrected.
% - OUTPUT is the image resulting from the skew correction process.
% - ANGLE is the number that represents the value of the skew corrected
% from the image IMG. It is expressed in degrees and measured in a
% counterclockwise direction.

% Rafael L. Cardero Álvarez
% Revision: 1.0.0   Last update: 23/5/2011 12:47 PM

% Function preparation
% Note: This method is intended to be used only with RGB and grayscale
% images
I= IMG;
if(ndims(IMG)==3)
    I= rgb2gray(IMG);
end

THETAS= -89:90;

% Step 1. Enhance image
[I]= enhance(I);

% Step 2. Segment image
[BW]= segment(I);

% Step 3. Preprocess image
[BW]= preprocess(BW);

% Step 4. Estimate skew angle
[skewangle]= estimate(BW, THETAS);

% Step 5. Rotate image to correct the estimated skew
[output]= imrotate(IMG, -skewangle);

end
% -----
```

### A4.2. enhance.m

```
function[enhanced]= enhance(I)
% ENHANCE Enhances a document image.
```

```

%
% USAGE: [ENHANCED]= ENHANCE(I)
% -----
% Applies digital image enhancement techniques on the document image I and
% stores the result of the enhancement in the grayscale image ENHANCED.
%
% - I is the grayscale image to be enhanced.
% - ENHANCED is the grayscale image resulting from the enhancement process.

% Rafael L. Cardero Álvarez
% Revision: 1.0.0   Last update: 15/10/2010 11:08 AM

% Step 1. Adjust contrast of the grayscale image I
enhanced= imadjust(I);
end
% -----

```

### A4.3. segment.m

```

function[segmented]= segment(I)
% SEGMENT Segments a document image.
%
% USAGE: [SEGMENTED]= SEGMENT(I)
% -----
% Applies digital image segmentation techniques on the document image I
% and stores the result of the segmentation in the binary image SEGMENTED.
%
% - I is the grayscale image to be segmented.
% - SEGMENTED is the binary image resulting from the segmentation process.
% Background pixels are set to black (false) and foreground pixels are set
% to white (true).

% Rafael L. Cardero Álvarez
% Revision: 1.0.0   Last update: 15/10/2010 11:18 AM

% Step 1. Binarize grayscale image I using Otsu's method
t= graythresh(I);
segmented= im2bw(I, t);

% Step 2. Apply a negative operation to set background as black (false) and
% text as white (true)
segmented= ~segmented;
end
% -----

```

### A4.4. preprocess.m

```

function[preprocessed]= preprocess(BW)
% PREPROCESS Preprocesses a document image.
%
% USAGE: [PREPROCESSED]= PREPROCESS(BW)
% -----
% Preprocess the document image BW in order to remove all the regions that
% are non-significant to estimate the skew. The result of the preprocessing
% is stored in the binary image PREPROCESSED.
%
% - BW is the binary image from which the non-significant regions will be
% removed.
% - PREPROCESSED is the binary image resulting from the preprocessing.

```

```

% Rafael L. Cardero Álvarez
% Revision: 1.0.0   Last update: 30/5/2011 10:53 AM

% Function preparation
P_PERCENT= 0.01;

% Step 1. Find connected components (request 'Area' and 'PixelIdxList' from each
component)
L= bwlabel(BW);
stats= regionprops(L, 'Area', 'PixelIdxList');

% Step 2. Generate a histogram containing the area of each connected component
h= [stats.Area];

% Step 3. Purge a 'P_PERCENT' of components, starting from the component
% with biggest area
N= ceil (P_PERCENT* length(h));
for(i=1:N)
    [c, idx]= max(h);
    BW(stats(idx).PixelIdxList)= false;
    h(idx)=0;
end

% Set result
preprocessed= BW;

```

% -----

#### **A4.5. estimate.m**

```

function[angle]= estimate(BW, thetas)
% ESTIMATE Estimates the skew of a document image.
%
% USAGE: [ANGLE]= ESTIMATE(BW, THETAS)
% -----
% Estimates the skew of the document image BW, taking into account the
% angles vector THETAS.
%
% - BW is the binary image from which the skew will be estimated.
% - THETAS is the vector containing the angles may be taken into account
% during the skew estimation process. Each angle must be expressed in
% degrees and measured in a counterclockwise direction.
% - ANGLE is the number that represents the value of the estimated skew.
% It is expressed in degrees and measured in a counterclockwise direction.

% Rafael L. Cardero Álvarez
% Revision: 1.0.0   Last update: 30/5/2011 11:00 AM

% Function preparation
thetas= thetas- 90; % Needed according to RT behaviour

% Step 1. Dilate the image BW to join words and perhaps lines
radius= calculateradius(BW);
str= strel('disk', radius);
BW= imdilate(BW, str);

% Step 2. Find connected components (request the 'Area' and 'BoundingBox'
% properties)
L= bwlabel(BW);

```

```

stats= regionprops(L, 'Area', 'BoundingBox');

% Step 3. Calculate criteria to select skew-representative components
criteria= calculatecriteria(stats);

% Step 4. Analyze skew-representative components
angles=[];

for(i=1:length(stats))
    selected= stats(i).Area>= criteria;
    if(selected)
        % Crop region on original image, fill holes and thin objects
        region= imcrop(BW, stats(i).BoundingBox);
        region= imfill(region, 'holes');
        region= bwmorph(region, 'thin', Inf);

        % Analyze RT of current region
        [regionangle]= analyzert(region, thetas);
        angles= [angles regionangle];
    end
end

% Step 5. Calculate mode of the angles vector and add 90 degrees
angle= mode(angles)+ 90;

end
% -----
function[radius]= calculateradius(BW)

% Step 1. Find connected components, and for each one, find the diameter of
% the circle with the same area as the object
L= bwlabel(BW);
stats= regionprops(L, 'EquivDiameter');

% Step 2. Find mode of the diameter-vector
vector= [stats.EquivDiameter];
radius= mode(vector);

% Get the highest possible value
radius= ceil(radius);

end
% -----
function[angle]= analyzert(BW, thetas)

% Step 1. Calculate Radon Transform of image BW
R= radon(BW, thetas);

% Step 2. Analyze RT to estimate the skew of image BW
values= max(R);
[c,idx]= max(values);
angle= thetas(idx);

end
% -----
function[criteria] = calculatecriteria(stats)

% Function preparation
PERCENT= 0.25;

```

```
% Step 1. Get area vector
vector= [stats.Area];

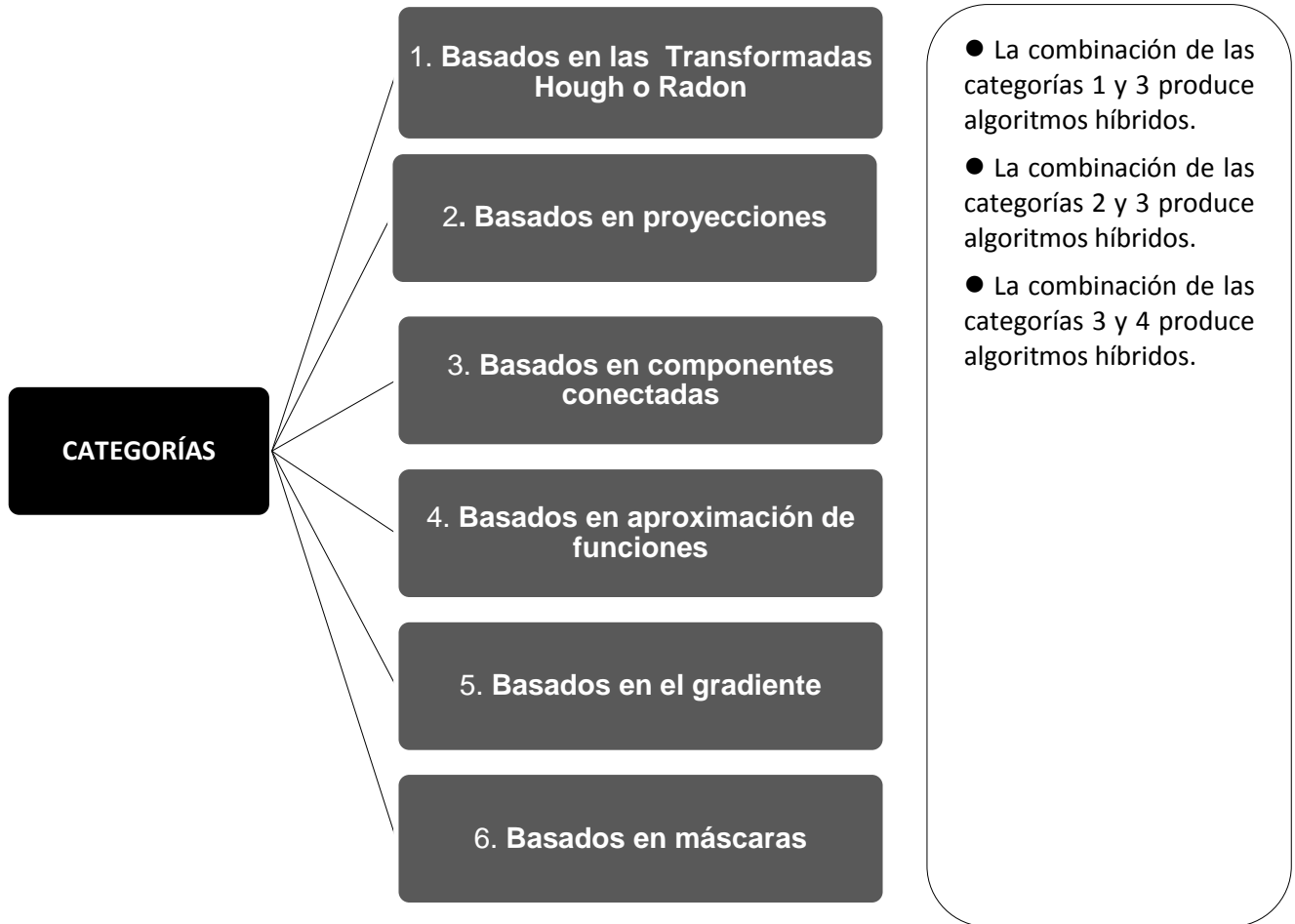
% Step 2. Sort vector ascendently
vector= sort(vector);

% Step 3. Select criteria in way that only a 'PERCENT' of components is
% used to estimate the skew, starting from the component with biggest area
index= (1- PERCENT)* length(stats);
index= floor(index);
criteria= vector(index);

end
% -----
```

## Anexo 5. Organigrama de algoritmos

En este anexo se presenta un organigrama que refleja jerárquicamente las categorías de los algoritmos que corrigen la inclinación en imágenes de documentos. Estas categorías fueron identificadas durante la revisión bibliográfica realizada como parte de la investigación. Nótese la existencia de algoritmos híbridos (Fig A5.1).



**Figura A5.1.** Organigrama de los algoritmos que corrigen la inclinación en imágenes de documentos

## GLOSARIO DE TÉRMINOS

En esta sección se expone el significado atribuido a términos utilizados repetidamente a lo largo del documento de tesis.

**Tabla G.1.** Elementos del glosario de términos.

<b>Término</b>	<b>Significado</b>
Hueco	Conjunto de puntos de fondo totalmente encerrados por puntos de objeto.
Línea de texto	Conjunto de símbolos adyacentes y relativamente cercanos, por encima de los cuales una recta imaginaria (horizontal) puede ser dibujada.
MBR	Rectángulo mínimo, orientado horizontalmente, que encierra a una componente conectada. Su nombre proviene de "Minimum Bounding Rectangle".
Modelo de color	Especificación de un sistema de coordenadas y de un subespacio dentro de dicho sistema, donde cada color se representa por un único punto.
Modelo de color RGB	Modelo de color basado en el sistema de coordenadas cartesiano y que forma un subespacio en forma de cubo dentro de dicho sistema. Los ejes del sistema son rojo (R: red), verde (G: green) y azul (B: blue). Por tanto cada color es obtenido combinando el valor de los colores primarios rojo, verde y azul.
Puntos de fondo	Puntos de una imagen que no conforman el área de interés, de acuerdo a un algoritmo de procesamiento de imágenes digitales determinado.
Puntos de objeto	Puntos de una imagen que conforman el área de interés, de acuerdo a un algoritmo de procesamiento de imágenes digitales determinado.
Símbolo del texto	Cualquier carácter alfanumérico, signo de puntuación, o carácter especial.