Uso de componentes conexas para restauracion automatica de documentos digitalizados.

Marisa R. De Giusti[1]; María Marta Vila; Villarreal Gonzalo Luján

Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)
Proyecto de Enlace de Bibliotecas (PrEBi)
Servicio de Difusión de la Creación Intelectual (SeDiCI)
Universidad Nacional de La Plata
La Plata, Argentina

marisa.degiusti@sedici.unlp.edu.ar, {vilamm, gonetil}@sedici.unlp.edu.ar

1. **Introducción**

La digitalización de documentos es un proceso que se ha simplificado enormemente con el avance de la tecnología y el desarrollo de dispositivos de captura de imágenes (scanner, cámaras digitales), lo cual ha provocado que enormes cantidades de documentos sean digitalizados y estén en constante movimiento dentro de la red. Esto ha generado grandes ventajas en muchos entornos, gracias a la posibilidad preservar grandes volúmenes de documentos y de transferirlos por medios digitales de manera inmediata, permitiéndole a un gran número de personas acceder a información que, de otra manera, hubiese sido imposible. Un claro ejemplo se da en el marco de bibliotecas digitales, dentro del cual los proyectos PrEBi y SeDiCI están fuertemente involucrados generando servicios, desarrollos de herramientas de software y líneas de investigación relacionadas con temas afines.

A pesar de los avances mencionados, la digitalizacion de documentos continua generando algunos problemas originado a partior de errores humanos o tecnicos, y que resultan en imagenes con informacion redundante, documentos de dificil lectura, archivos excesivamente grandes, etc. Entre estos problemas podemos destacar:

- Documentos con bordes negros o difusos: Sucede comúnmente cuando los documentos originales se encuentran en mal estado, o cuando el scanner no se ha cerrado apropiadamente. Esto implica documentos de mayor tamaño, archivos digitales más pesados, mayor consumo de tinta al imprimirlos y dificultad para leerlos.
- Aparición de puntos aleatorios: Puede suceder tanto debido a la baja calidad de los documentos originales como a problemas en el scanner; esto genera, en el mejor de los casos, un efecto de documento “sucio” o desprolijo; mientras que en el peor de los casos, muchas partes del texto serán difíciles de leer debido a la gran cantidad de puntos que se intercalan con el texto (efecto de textura arenosa).
- Inclinación de las imágenes: Quizás por distracción del operador, por un mal funcionamiento de scanners automáticos o debido a la falta de espacio al digitalizar documentos (sucede comúnmente con libros de gran tamaño), muchos documentos poseen cierta inclinación respecto de la línea base. Esta inclinación puede afectar en mayor o menor medida al lector del documento. Por ejemplo, una inclinación con un ángulo menor a 5 grados no implica mayores problemas para el lector del documento, pero si la inclinación supera los 5 grados y está por debajo de los 85 grados, el problema se agudiza. Entre los 85 y los 95 grados, nuevamente no hay mayores inconvenientes (el lector puede simplemente cambiar la dirección del papel de horizontal a vertical).

El este articulo presentaremos un mecanismo automatizado y optimizado para mejorar documentos digitalizados que presentan algunos o incluso todos los problemas descriptos arriba. Primero se introducirá brevemente el concepto de componentes conexas, las cuales son esenciales para todo el proceso. A continuación, se explicará en detalle un mecanismo de detección de la inclinación de documentos, junto con un algoritmo básico de rotación de imágenes. Luego, se explicará cómo detectar y eliminar los bordes negros de las imágenes, así como también cómo detectar y eliminar puntos aleatorios. Por último, se propondrá un mecanismo para integrar todas estas tareas de modo de automatizar el proceso de restauración.

2. Componentes Conexas.

2.1 Descripción.

El uso de las componentes conexas para el procesamiento de imágenes digitales no es nuevo, pero dada la información que estas componentes recogen a partir de una determinada imagen se encuentran constantemente nuevas aplicaciones en varios mecanismos relacionados con el procesamiento de imágenes.
Básicamente, las componentes conexas permiten asociar elementos o partes de imágenes entre sí para luego realizar operaciones a partir de la composición de cada imagen en partes. Una componente conexa no es más que un conjunto de puntos o píxeles de las imágenes que se han agrupado a partir de cierta característica que los

1 Investigador Comisión de Investigaciones Científicas de la Provincia de Buenos Aires – CIC.

identifica. Esta agrupación permite, por ejemplo, discriminar partes de una imagen, encontrar relaciones entre los elementos de una misma componente o incluso relaciones entre distintas componentes, así como aplicar ciertas operaciones y filtros a porciones de una imagen.

Como se ha explicado en [1], un conjunto de componentes conexas puede verse como un conjunto de puntos agrupados en Clases de Equivalencia, donde un determinado punto "p" pertenece a exactamente una clase de equivalencia, y donde no existen clases de equivalencia vacías. Asimismo, en [1] se desarrolló un método para el procesamiento de CC y en [2] se han explicado algunas aplicaciones útiles. En este documento, se introducirán algunas modificaciones al algoritmo para el cálculo de las CC, introduciendo nuevas estructuras que permitirán mejorar los tiempos de respuesta y optimizar el cálculo de ciertos parámetros.

2.2 Optimizando la técnica.

El primer cambio introducido es el uso de una estructura de datos matricial del mismo tamaño que la imagen original, con valores enteros e inicializada en cero. Si bien esto consume una gran cantidad de memoria – especialmente cuando se procesan gran cantidad de imágenes de tamaño importante en paralelo- permite retrasar la aplicación de operaciones (que consisten básicamente en el cambio de valor de determinados puntos de la imagen) marcando los puntos a modificar y realizando los cambios al final del proceso.

Sea I la imagen original, de tamaño MxN, y sea CC[M,N] la matriz que contiene las componentes conexas, también de tamaño MxN. Se asume que I contiene al menos un punto negro.
El proceso de cálculo de las componentes conexas es el mismo descripto en [1], solo que ahora la matriz CC tendrá información acerca de las componentes conexas, organizada de la siguiente manera:

Sea $p_{i,j}$ un punto de la imagen I, donde 0 <= i < M; 0 <= j < N.
Sea CCs el conjunto de todas las CC de la imagen; CCs (por lo asumido previamente). Sea Ck CCs, Ck es una componente conexa y k > 0 es su etiqueta. Luego, la matriz CC queda compuesta de la siguiente manera:

- CC[i,j] = 0 $p_{i,j}$ no pertenece a ninguna componente conexa
- CC[i,j] = k $p_{i,j}$ pertenece a la componente conexa Ck CCs

La principal ventaja de esta estructura es que permite conocer inmediatamente a que componente conexa pertenece cualquier punto p I. Como desventaja, se utiliza una gran cantidad de memoria para almacenar una matriz de igual tamaño que la imagen para mantener las etiquetas.

Muchas operaciones requieren conocer el tamaño de la componente conexa a la que pertenece un determinado píxel. Sea pues size(Ck) una función que retorna la cantidad de puntos que forman la componente Ck. Claramente, size(Ck) > 0 dado que Ck siempre.
El tiempo requerido para el cálculo de esta función es un factor determinante al momento de procesar las componentes, debido que muchas operaciones requieren reiteradametne este dato y un calculo ineficiente puede provocar una importante degradación en el proceso completo.
Para maximizar la eficiencia de esta operación y llevarla a un orden de ejecución mínimo, se utilizó una segunda estructura de datos que almacena el tamaño de cada componente. Sea entonces S un arreglo de longitud igual a la cantidad de elementos del conjunto CCs. Así pues, si S[k] = m significa que la componente Ck CCs posee m elementos. De este modo, el orden de ejecución de la función size() se ha llevado un tiempo constante. Como desventaja, se acarrea una penalización al momento de calcular las componentes conexas, ya que ahora se realiza un nuevo procesamiento para calcular los tamaños. De todas maneras, en la práctica se ha encontrado que la clara mejoría en los tiempos de respuesta del resto de los algoritmos hace que esta penalización bien valga la pena.

3. Operaciones de mejoramiento de imágenes.

3.1 Corrección de la inclinación

Como se mencionó previamente, el problema de la inclinación es muy común en el proceso de digitalización de documentos y es evidente que un documento inclinado ofrece dificultades a la hora de su lectura.

El objetivo buscado es corregir automáticamente la inclinación de las páginas, evitando la intervención de un operador que modifique cada página del documento.
Para lograr este objetivo, se plantean dos problemas: cálculo de la inclinación actual y corrección de la inclinación.

3.1.1 Detección de la inclinación del documento

La detección automática del ángulo de inclinación de un documento es un proceso complejo y no siempre se logran los resultados esperados. En la literatura actual se describen muchas técnicas para detectar la inclinación de documentos, que se diferencian entre sí por la robustez y complejidad del método subyacente utilizado, la exactitud de los resultados y la eficiencia de los algoritmos, tanto en tiempo de ejecución como en recursos del sistema.

En [3] se describe un método robusto basado en componentes conexas que consiste en bajar la calidad de la imagen para así obtener grandes cuadros que representan palabras y luego detectar la inclinación de estas palabras. En [4] y [5] se describen dos métodos en el cual se toman de manera aleatoria partes de los documentos y se utiliza simultáneamente la correlación

cruzada vertical y horizontal, lo cual permite detectar la inclinación de documentos con caracteres chinos o japoneses. Estos métodos, aunque revisten mediana o alta complejidad, logran muy buenos resultados y a la vez gran eficiencia.

Existen asimismo métodos más sencillos (como se describe en [7] y [8]) en los cuales intenta maximizar la diferencia entre picos y valles del histograma horizontal de la imagen, para lo cual se realizan sucesivas rotaciones calculando proyecciones horizontales, y tomando los resultados a partir del análisis de estos histogramas se ajusta el ángulo de rotación en base a una especie de búsqueda dicotómica. Si bien los resultados no son tan precisos (el método es fuertemente afectado por el ruido generado en la digitalización, como por ejemplo los bordes negros) y los tiempos de ejecución algo mayores, la sencillez que estos métodos ofrecen permite el desarrollo de algoritmos de manera rápida y apropiados en determinados entornos.

Entre todos los métodos analizados, el método presentado en [6] mantiene un justo balance entre complejidad y eficiencia, logrando resultados de gran calidad. Este método ha sido desarrollado y se le han realizado ciertas modificaciones, a fin de lograr solucionar algunos problemas descriptos en la publicación original.

El algoritmo consta de 7 pasos en serie, pudiendo existir ciclos o iteraciones en esta secuencia hasta lograr el ángulo deseado.

Estos pasos son:

1. Detectar el conjunto de todas las CC de la imagen;
2. Calcular para cada CC: su centroide (Cx,Cy), su altura (h), su ancho (w) y el rectángulo que la encierra (puntos superior izquierdo e inferior derecho).
3. Para cada CC, se calcula la NNC (cadena de vecinos mas cercanos, o *nearest-neighbor chain*) :

a. Encontrar la CC más cercana.
Sean C_1 y C_2 CC de la imagen I. C2 es el vecino más cercano (NN) de C1 si $dc(C_1,C_2) = \min dc(C_1,C_m)$ m ; donde $dc(C_a,C_b) = x^2 + y^2$ es la función de distancia entre CC y $x=|x_{c1} - x_{c2}|$

b. Si se ha encontrado una componente cercana(C_2), se debe verificar que
- hC_1 hC_2 para x > y (siendo hC_i la altura de la CC i) ó wC_1 wC_2 para y > x (siendo wC_i el ancho de la CC i)
- $Cx_2 > Cx_1$ para x > y , ó
$Cy_2 > Cy_1$ para y > x
- $dg(C_1 , C_2) <$. max (hC_1 , hC_2)
= 1.2 (constante).

c. Si C2 cumple con las condiciones anteriores entonces se registra como NN de C1 y se identifica que C2 no puede ser raíz.

Por ejemplo:

# componente	NNC	Es raíz de la cadena?
1	16	SI
8	22	SI
16	32	NO
24	30	SI
30	40	NO

4. Identificar todas las K-NNC (NNC de longitud K)

Para cada raíz:

a. Calcular la longitud de la cadena L.
b. Incrementar la cantidad de cadenas de longitud L.

Con estos cálculos, se genera un vector que tendrá en cada posición i la cantidad de NNC de tamaño i y las raíces de dichas NNC.

Del ejemplo se obtiene como resultado del punto 4:

k	Cantidad de cadenas	Raíces
3	2	1, 24
2	1	8

A medida que se va procesando cada una de las cadenas, se guarda el k con mayor cantidad de cadenas

5. Buscar el mayor k: Se inicializa K como el mayor número de CC de todas las NNC, y se calcula la cantidad de K-NNCs. Si el valor encontrado es inferior a 3, se repite la búsqueda con K = K - 1. En este punto Lu et al. en [6] sugieren que se debe utilizar un umbral mínimo de 3. En ciertas ocasiones, como por ejemplo procesando documentos con poco texto y muchas imágenes dentro, se observo que era conveniente bajar el umbral hasta 2 de ser necesario.

6. Procesar cada K-NNC obtenida en el paso 5 para obtener el ángulo de inclinación de cada una.

7. Obtener el ángulo de inclinación del documento (S_D) como la media de los ángulos obtenidos en el paso anterior, y luego se calcula el ángulo de ajuste:

$$S_D = \arctan(S_D) \times 180 / \quad .$$

3.1.2 Rotación del documento

Una vez que se ha encontrado el ángulo de inclinación del documento, se debe realizar una rotación inversa del documento, o sea rotar en - para corregir el problema[2]. El algoritmo de rotación es muy

2 El ángulo de rotación debe estar representado en

sencillo, y la única restricción que tiene es que debe mantener el centro de la imagen.

Para rotar una imagen se aplica una transformación afin[3], o lo que es lo mismo, se aplica una matriz afin sobre la imagen I. La transformación que se tiene que aplicar sigue la lógica de otras transformaciones:

$$R(x,y) = A(f1(x,y),f2(x,y));$$
$$f1,f2 : (\mathbf{N} \ , \ N) \rightarrow R$$

Esta matriz afin se conoce como Matriz **M** Afin de Rotación, o Rotation Matriz, la cual se calcula del siguiente modo

$$M = \begin{matrix} [\ a \ ß \ | \ (1\text{-}a) \ . \ x_p - ß \ . \ y_p \ ] \\ [\ -ß \ a \ | \ ß \ . \ x_p + (1\text{-}a) \ . \ y_p \ ] \end{matrix}$$

donde a=scale*cos(α), ß=scale*sin(α), y scale representa la escala de la imagen rotada (idealmente, se debe mantener el tamaño original). Podemos entonces definir una base coordenada en cada punto del espacio, mediante los vectores tangentes a las líneas coordenadas. Esta nueva base puede relacionarse con la base fundamental de las coordenadas cartesianas mediante las relaciones, con lo cual podremos aplicar la transformada definiendo como se mapea cada punto del sistema origen con el sistema destino.

$$w' = h. |sen(\alpha)| + w .|cos(\alpha)|$$
$$h' = w. |sen(\alpha)| + h .|cos(\alpha)|$$

donde w' y h' representan el tamaño requerido para la imagen rotada. Esto nos permite calcular la escala, de la siguiente manera:

sw = w_d / w' ; escala del ancho, w_d es el ancho de la imagen destino
sh = h_d / h' ; escala del alto, h_d es el alto de la imagen destino
scale = min(sw,sh) ; la escala real será aquella que más se ajuste a la imagen destino

Finalmente, se ajusta la rotación al centro la imagen destino.

Una vez calculada la matriz M, podremos aplicar la transformada R de rotación.

En las siguientes figuras siguientes se observan los resultados obtenidos. En cada imagen original se aplica el algoritmo que calcula el ángulo y luego la rotación de la misma.

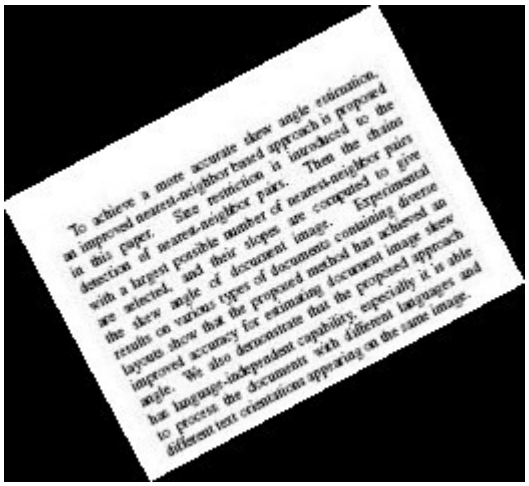


Imagen original

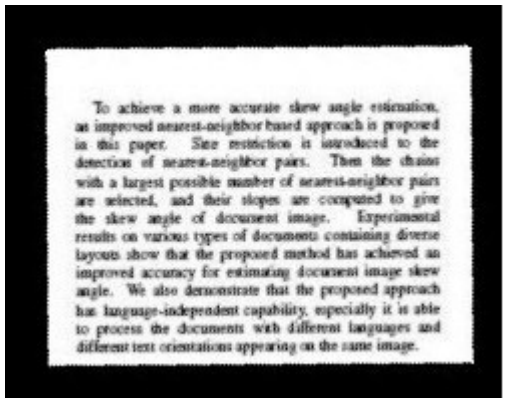
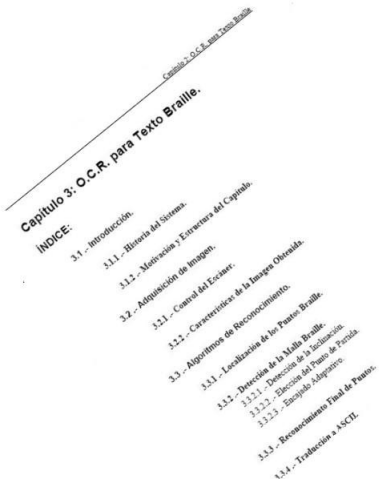
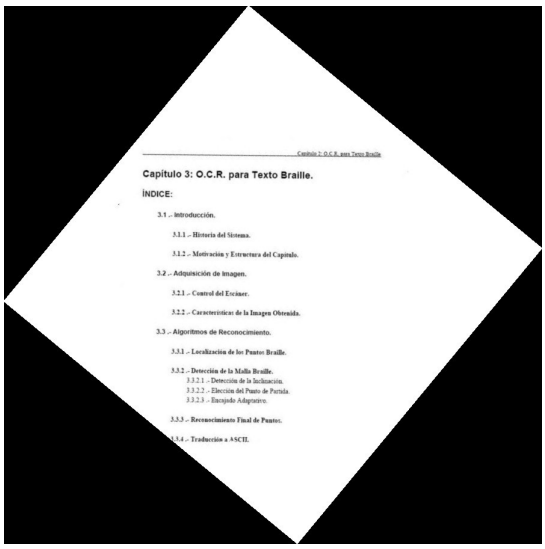


Imagen rotada



radianes, con lo cual si se encuentra en grados debe transformarse: α = (degree . π)/180

3 Una transformación afin es cualquier transformación de Rotación, Traslación, Escala e Inclinación, o incluso cualquier combinación entre estas



Arriba Imagen Origina. Abajo Imagen rotada

Arriba Imagen original ; Abajo Imagen rotada.

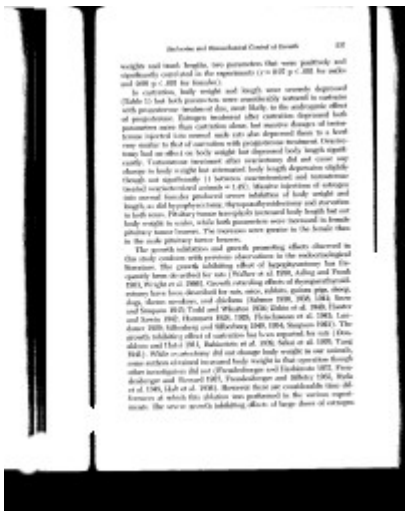
3.2 Eliminación de bordes negros

Los bordes negros de las imágenes pueden surgir como consecuencia del cierre incorrecto del equipo digitalizador debido tanto a negligencia del operador como a la necesidad de digitalizar libros o revistas de gran volumen. Esto genera un borde negro alrededor de la imagen original, del mismo tamaño del escaner, lo cual dificulta la lectura del documento, genera imágenes de mayor tamaño y principalmente, provoca un uso indebido de tinta o tóner al imprimir el documento.

El uso de las componentes conexas puede ser de gran utilidad al momento de buscar una solución automatizada a este problema. Un borde negro, o al menos una franja negra, formará una sola componente conexa y la cantidad de puntos que pertenecen a dicha componente será relativamente grande, especialmente si se la compara con el tamaño promedio de las restantes componentes.

Un problema que puede traer aparejado este método es la alteración de imágenes o cuadros que se encuentren dentro de la imagen, ya que los mismos también generarán componentes conexas de gran tamaño en relación a la media de la imagen. Y el objetivo de esta técnica es modificar solo zonas periféricas del documento.

Otro problema común surge cuando los bordes negros no afectan a toda la periferia, o existe más de una zona negra que rodea la imagen. Como puede observarse las Figuras y , la eliminación de las componentes periféricas no asegurará la total limpieza de los bordes negros.



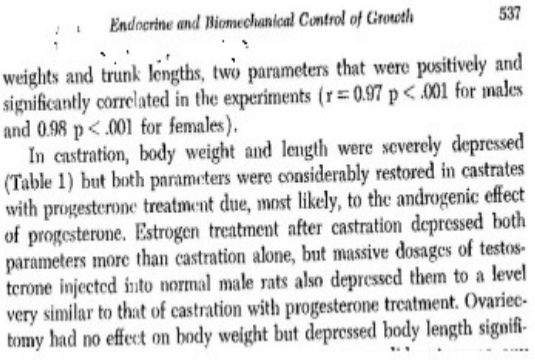
Original

Primera limpieza

Para solucionar estos problemas, se puede realizar una limpieza en etapas, partiendo de las componentes que se encuentran mas cerca del borde, y luego continuar hacia el centro de la imagen hasta encontrar componentes que no tengan el tamaño suficiente para ser eliminado.

3.3 Eliminación de puntos aleatorios (despeckle)

Un problema muy común que surge a partir de la digitalización de documentos es la generación de puntos aleatorios en cualquier parte de la imagen, como se muestra en la Figura . Esto puede suceder debido al mal estado del documento original, a la presencia de suciedad en el escáner o a la baja calidad de las imágenes almacenadas.



Endocrine and Biomechanical Control of Growth 537

weights and trunk lengths, two parameters that were positively and significantly correlated in the experiments ($r = 0.97$ $p < .001$ for males and 0.98 $p < .001$ for females).

In castration, body weight and length were severely depressed (Table 1) but both parameters were considerably restored in castrates with progesterone treatment due, most likely, to the androgenic effect of progesterone. Estrogen treatment after castration depressed both parameters more than castration alone, but massive dosages of testosterone injected into normal male rats also depressed them to a level very similar to that of castration with progesterone treatment. Ovariectomy had no effect on body weight but depressed body length signifi-

Texto con puntos aleatorios

Las componentes conexas permitirán nuevamente solucionar este problema; estos puntos pertenecerán a componentes conexas muy pequeñas, de a lo sumo 2 ó 3 píxeles, con lo cual la solución al problema se reduce a recorrer la imagen buscando componentes pequeñas. Pero gracias a que existe un vector con los tamaños, la solución se puede hacer muy eficientemente. Sea p un punto cualquiera de la imagen I, y sea el umbral que determinará el tamaño mínimo que una componente conexa debe tener para no ser eliminada. Sea cc = CCs[p_x,p_y]. Luego,

Si size(cc) < => $I_{x,y}$ = b (donde b representa un punto blanco)

Gracias a que la implementación de la función size() hace que requiera tiempo de ejecución constante, esta operación es muy eficiente.

4. Proceso automatizado.

Las operaciones de limpieza permiten obtener muy buenos resultados, pero requieren realizar sucesivas pasadas por la imagen (al menos una pasada por cada operación), tanto para realizar cálculos como para aplicar modificaciones. Se propone aquí un mecanismo automatizado para minimizar la cantidad de pasadas y trabajar con áreas de mayor relevancia, evitando de este modo procesar zonas de la imagen que no tienen sentido y aumentando la velocidad del proceso en general.

La primera operación será, inevitablemente, la de detección de la inclinación y posterior rotación de la imagen; debe realizarse antes que el resto debido a que podrá generar cambios en la matriz de componentes conexas provocando que las siguientes operaciones requieran recalcular esta matriz.

A continuación, se procede con la limpieza del ruido presente en la imagen. Se puede mejorar la performance del sistema desde dos frentes: minimizando la cantidad de pasadas sobre la imagen y minimizando el área sobre la que se trabajará.

Para minimizar la cantidad de pasadas sobre la imagen, aprovechando el hecho de que siempre se eliminan componentes conexas completas, se propone marcar cuales serán las componentes conexas a eliminar en cada uno de los pasos de la limpieza y luego recorrer la imagen una sola vez eliminando los puntos de dichas componentes. Nuevamente, con la misma idea que se aplicó para calcular los tamaños de manera eficiente, se puede utilizar un arreglo de componentes a eliminar. Sea E dicho vector, de tamaño k (donde k es la cantidad de componentes conexas de la imagen). Al iniciar el proceso de limpieza, E[i] = 0 para todo 0 <= i < k. Luego, al finalizar el proceso tendremos:

E[i] { 1 si la componente conexa i debe ser eliminada
0 en caso contrario

Al utilizar este vector de marcas, no se realizan cambios en la imagen constantemente mientras se identifican las componentes a eliminar, sino que se retrasan todos los cambios para el final. Una vez identificadas todas las componentes conexas, se realizará una única pasada por la imagen, aplicando una función condicional muy sencilla

$I_{x,y}$ = b (donde b representa un punto blanco) si y solo si E[i] = 1

Se puede aún mejorar más la eficiencia del algoritmo. Como se mencionó previamente, la

eliminación de bordes negros debe realizarse en secuencia, avanzando de afuera hacia adentro. Pero una vez que se ha detectado el último borde negro (el más cercano al centro de la imagen), este puede considerarse como el mínimo "bounding box". Esto significa que todos los puntos que estén fuera de la imagen serán ignorados, lo cual ahorrará mucho tiempo de CPU ya que no se requerirá la eliminación ni de los bordes negros ni de los puntos aleatorios fuera de esta área. Adicionalmente, se optimiza el tamaño de la imagen con el que se esta trabajando pero sin perder calidad, ya que solamente se considera el espacio útil de la página. Esto trae como ventajas no solo la reducción del archivo resultante, sino que permitirá obtener imágenes con información relevante, lo cual es especialmente útil si se implementará algún tipo de reconocimiento de patrones o procesamiento posterior utilizando algoritmos evolutivos, ya que estas técnicas tienden a confundirse ante la presencia de ruido en las imágenes.

Calculo de eficiencia.

Es deseable conocer el orden de ejecucion de los algoritmos aquí presentados, para poder detectar problemas de eficiencia y plantear posibles mejoras. Para ello, se calcula el maximo de todos los ordenes de ejecucion O(n) de cada uno de los pasos del algoritmo.
Se realizan algunas consideraciones previas:

1. Se toman imagenes cuadradas, o sea se NxN . Es equivalente a tomar el "peor caso" en una imagen de NxM con N > M.
2. Esto hace que cada pasada determine como minimo un O(n) cuadratico, $O(n^2)$, debibo a que se analizan los n^2 puntos.
3. Siempre se considera el peor caso, pero esto no es 100% realista. Por ejemplo, en el peor de los casos, una imagen posee $n^2/2$ componentes conexas (con 4 vecindad), pero esto no tendria sentido ya que seria una imagen con puntos blancos y negros sin ningun vecino a los lados, arriba y abajo (una malla conectada diagonalmente).

La tecnica de rotacion consta de una secuencia de pasos bien definida:
Calculo de Componentes Conexas
Deteccion del angulo de inclinacion
Rotacion de la imagen

1. El orden de ejecucion para calcular las componentes conexas es de $O(n^2)$.
2. Como se menciono, en el peor de los casos habria un total de $n^2/2$ componentes conexas. De aquí, el algoritmo de deteccion del ángulo de rotación es de $O(n^4)$, ya que procesa todas las componentes conexas, calcula las NNC, obtiene y procesa las K-NNC y finalmente obtiene el angulo buscado. Los tiempos de ejecucion para cada uno de los pasos involucrados son:
    - Paso 1: $O(n^2/2)$
    - Paso 2: $O(n^4)$
    - Paso 3: $O(n^4)$
    - Paso 5 y 6: O(1)
    - Paso 7: $O(n^2/2)$
    - Paso 8: O(1)
3. Finalmente, el orden de ejecución de la rotación es de $O(n^2)$, ya que solo implica obtener la matriz de rotacion - $O(n^2)$ - y aplicarla a la imagen – $O(n^2)$ -.

De 1, 2 y 3 se puede facilmente concluir que el orden de ejecucion esta determinado por el punto 2, o sea $O(n^4)$.

Conclusiones.

En el campo del procesamiento de imágenes digitales existen numerosos algoritmos y técnicas para realizar las más diversas tareas; en particular, las técnicas relacionadas con el área de mejoramiento de documentos digitalizados han mejorado considerablemente y se han obtenido muy buenos resultados. De todos modos, muchas de las técnicas pueden ser aun optimizadas, agrupadas junto a otras que realicen tareas similares o relacionadas, o ajustadas/adaptadas para nuevas aplicaciones.

En este documento, la técnica introducida en [6] ha sido modificada para, por un lado optimizar su eficiencia pensando especialmente en el procesamiento en secuencia de numerosos documentos, y por el otro orientado a procesar documentos con imágenes dentro o incluso documentos manuscritos.

En la practica, esta tecnica se ha utilizado para descomponer documentos con formato PDF en un conjunto de imagenes (una por pagina), a las cuales se les han aplicado los algoritmos descriptos para finalmente reconstruir nuevamente el archivo en formato PDF. Los resultados obtenidos han permitido generar un proceso secuencial para realizar un mejoramiento automatizado y eficiente, y han sentado las bases para continuar desarrollando técnicas orientadas al mejoramiento de imágenes digitales de documentos impresos y manuscritos, mientras que paralelamente se continuan mejorando los algoritmos ya desarrollados buscando por un lado optimizar el uso de recursos y por el otro aprovechar las ventajas ofrecidas por las nueva tecnologías como procesadores de varios núcleos o incluso procesamiento en cluster.

5. Referencias.

[1] Marisa R. De Giusti, Maria Marta Vila, Gonzalo Luján Villarreal. "Manuscript Document Digitalization and recognition: a first approach". *Journal of Computer Science and Technology*. Vol 5. No. 3. 158-163,2005.

[2] Marisa R. De Giusti, Maria Marta Vila, Gonzalo Luján Villarreal. "Manuscript Character Recognition Overview of features for the Feature Vector". *Journal of Computer Science and Technology*. Vol 6. No. 2. 92-98,2006.

[3] Oleg Okun, Matti Pietikäinen and Jaakko Sauvola, "Robust Document Skew Detection Based on Line Extraction". [1]Machine Vision and Media Processing

Group, Infotech Oulu and Dept. of EE, University of Oulu

[4] Avanindra and S. Chaudhuri, Robust Detection of Skew in Document Images, *IEEE Trans. on Image Processing*, Vol. 6, No. 2, 1997, pp. 344-349.

[5] Ming Chen, Xiaoqing Ding "A Robust Skew Detection Algorithm for Grayscale Document Image". Fifth International Conference on Document Analysis and Recognition (ICDAR'99). p. 617

[6] Yue Lu, Chew Lim Tan. "A nearest-neighbor chain based approach to skew estimation in document image". Pattern Recognition Letters 24 (2003) 2315–2323

[7] Pastor Gaeda, Moises, Aportaciones al reconocimiento automático de texto manuscrito**,** Dpto. de Sistemas Informáticos y Computación, Universidad de Valencia.

[8] H.S. Baird. The skew angle of printed documents. In *Proc. of the Conf.Society of Photographic Scientists and Engineers*, pages 14–21, 1987.