UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

Procesamiento de Textos Manuscritos - Técnicas de Agrupamiento de Imágenes de Palabras

Luis Eduardo Romero Véjar
Tutor: José Colás Pasamontes

JULIO 2016

Procesamiento de Textos Manuscritos - Técnicas de Agrupamiento de Imágenes de Palabras

AUTOR: Luis Eduardo Romero Véjar
TUTOR: José Colás Pasamontes

HCTLab
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2016

Resumen (castellano)

El procesamiento de textos manuscritos, principalmente si se trata de textos históricos (incunables), es un área de gran interés en países donde el patrimonio cultural es muy amplio y de gran valor como es el caso de España. Hasta la fecha los archivos y fondos documentales que custodian este tipo de documentos históricos sólo ha podido digitalizar (fotografiar) los mismos para permitir el acceso a través de Internet y preservar esta riqueza documental que, en nuestro caso, data de muchos siglos atrás. El objetivo de este trabajo es desarrollar una herramienta informática que permitan la extracción del texto a partir de las fotografías de estos documentos. Para este fin se integrarán en la herramienta distintos bloques que permitirán preprocesar las imágenes de páginas de documentos históricos cuya calidad generalmente dista de estar en condiciones óptimas de inteligibilidad. Una vez preprocesada la imagen procederá a ser segmentada a nivel de línea y de palabra; de este modo se extraerán las imágenes de palabras que conforman cada página que posteriormente serán agrupadas por similitud para generar diccionarios visuales que permitan la transcripción manual de los mismos a especialistas y de esta forma el proceso de transcripción pueda ser automatizado. Se construirá una herramienta en MATLAB que permitirá integrar todos los procesos llevados a cabo para que pueda ser utilizada por gente no especializada en el ámbito de la informática. También, se desarrollarán técnicas de evaluación objetivas gracias a los recursos documentales de esta naturaleza disponibles en el grupo HCTLAB.

Abstract (English)

Manuscript processing, mainly if they are historical texts (incunabula), is an area of great interest in countries where cultural heritage is very broad and valuable as is the case in Spain. To this day, files and collections of historical documents have only been able to digitize (photograph) them to allow access through the Internet and preserve this documentary wealth which, in our case, dates back many centuries. The aim of this work is to develop a software tool that allows text extraction from photographs of these documents. For this purpose, the tool will consist of different blocks that will preprocess the images of pages of historical documents whose quality is generally far from being in optimum conditions of intelligibility. Once the image is preprocessed it will be segmented (line and word level); thus the images of words that make up each page subsequently be grouped by similarity to generate visual dictionaries that allow manual transcription by specialists and so the transcription process can be done automatically within a tool in MATLAB that will integrate all processes carried out so that it can be used by people not specialized in the field of information technology. Also, evaluation techniques will be developed and objective assessment techniques are also developed thanks to the documentary resources of this nature available in the HCTLAB group.

Palabras clave (castellano)

Preproceso, segmentación, incunable, agrupamiento, *bounding box*, *skew*, *slant*, CCL, actas, histograma, marco de escaneado, binarización, RGB, GUI, ruido impulsional, filtrado, etiquetas, esqueletización.

Keywords (inglés)

Pre-processing, segmentation, incunabula, grouping, *bounding box*, *skew*, *slant*, CCL, transactions, histogram, scanning frame, binarization, RGB, GUI, impulse noise, filtered, tags, skeletization.

Agradecimientos

A José Colás Pasamontes por permitirme trabajar con él y por su gran ayuda durante el desarrollo del proyecto.

A mis padres por su apoyo durante la carrera.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

1 Introducción

1.1 Motivación

El procesamiento de textos manuscritos, especialmente en documentos históricos antiguos es un área en la que existe actualmente escasa investigación y desarrollo de herramientas que permitan tanto digitalizar (con el objetivo de preservar) como clasificar de forma automática archivos y fondos documentales. Los procesos actuales son, por lo general, lentos y costosos [1] por lo que no pueden reconocer y digitalizar de forma eficiente grandes volúmenes de documentos históricos.

Existe una ingente cantidad de archivos de textos manuscritos de este tipo que no han sido transformados a un formato de digital, para que puedan ser consultados e indexados con mayor facilidad por especialistas y para que sean accesibles al público no especializado debido a la baja inteligibilidad que presentan los archivos de origen.

Actualmente, el creciente número de librerías digitales en línea de textos antiguos hace que el reconocimiento de textos manuscritos sea un campo de investigación notable dado que hasta el momento no se han encontrado soluciones óptimas donde se agrupen todas las técnicas necesarias para transcribir el texto presente en archivos históricos a formatos digitales comúnmente usados hoy en día. Entre las técnicas necesarias para efectuar la transcripción se encuentra el preproceso del archivo de origen (puesto que suelen estar en condiciones distantes de las ideales [8] principalmente por el filtrado de la tinta de las páginas adyacentes), la segmentación a nivel de palabra (para obtener imágenes de palabras que pasarán a ser interpretadas como texto) y finalmente el agrupamiento de dichas imágenes (ya que hacer la transcripción de cada imagen de palabra de forma individual conlleva unos costes muy altos).

Este campo de la investigación guarda similitud con el Reconocimiento Óptico de Caracteres (OCR), no obstante, dicho procedimiento no ofrece una solución al problema de estudio de este Trabajo de Fin de Grado dado que los caracteres de textos manuscritos no pueden ser aislados de forma automática ya que las formas irregulares y difusas de los mismos [3] sumado al resto de imperfecciones presentes en cada página de un documento histórico hacen que el número de errores al aplicar esta técnica sea muy elevado.

La motivación de este proyecto es desarrollar una herramienta que dé solución al problema de la clasificación automática de textos históricos, es decir, que integre todas las partes necesarias para una correcta extracción de información de los mismos, en particular aquellos con una alta complejidad que presenten una gran degradación y baja inteligibilidad; una herramienta parametrizable en función de las características de cada obra que permita a los historiadores un manejo sencillo de textos digitalizados facilitando así el acceso a los mismos al público general y que acciones como efectuar búsquedas sobre ellos sean mucho más sencillas además de permitir la conservación y archivado en el tiempo de cualquier obra manuscrita sin importar la complejidad del texto y la calidad del archivo.

1.2 Objetivos

El objetivo de este Trabajo de Fin de Grado es el desarrollo de una herramienta que permita la extracción y posterior tratamiento del texto de archivos históricos. Debido a que no se puede automatizar esta tarea totalmente, dada la naturaleza cualitativa del procesado de textos, se pretende poner la mayor potencia de cómputo actual al servicio del procesado de textos manuscritos históricos; para ello la herramienta cubrirá las tres secciones principales que, en conjunto, permitirán digitalizar y transcribir gran parte de dichas obras de una forma eficiente con el mayor nivel de automatización posible.

El primer objetivo será efectuar un preproceso adecuado de la imagen, donde ajustando diversos parámetros en función de la calidad del texto se eliminarán las imperfecciones de la misma como el *skew*, el ruido impulsional, etc. Todas las opciones deben ser configurables por el usuario de una forma sencilla e intuitiva desde el menú de la herramienta.

El segundo objetivo será identificar y extraer las palabras y líneas del texto (segmentar) mediante la utilización de diversas técnicas ya existentes [4] y otras diseñadas como parte de este TFG de cara a mejorar la segmentación.

El tercer objetivo es agrupar las palabras obtenidas en la fase anterior de cara a una rápida y eficiente clasificación a nivel de palabra.

Todos los bloques que cubren todas las áreas explicadas anteriormente estarán presentados bajo una herramienta informática intuitiva a través de la cual un especialista en el reconocimiento de textos podrá clasificar los mismos y parametrizar la herramienta sin necesidad de poseer conocimientos especializados de programación o ejecución de comandos.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- En el capítulo 1 se explica la motivación de este TFG donde se refleja el problema actual de la clasificación automática de textos manuscritos.
- En el capítulo 2 se hace referencia al estado del arte.
- En el capítulo 3 se explica el diseño de la herramienta construida en este TFG.
- En el capítulo 4 se explica detalladamente la base de datos de documentos usada para llevar a cabo este Trabajo de Fin de Grado.
- En el capítulo 5 se efectuará un análisis de los resultados obtenidos tras poner a prueba el sistema.
- Finalmente, en el capítulo 6 se explicará el trabajo futuro de este proyecto además de las conclusiones.

2 Estado del arte

En esta sección, se va a realizar una introducción al estado del arte existente. Antes de profundizar conviene justificar la necesidad de la digitalización de los archivos históricos siendo el motivo principal la conservación y difusión de los archivos que constituyen estos repositorios.

Las formatos obsoletos como el papel han dado paso al formato digital que presenta ventajas lo suficientemente notables como para que tanto las administraciones como el sector privado hayan depositado en el entorno digital una gran expectación [2]. La ventaja más destacable de dicho formato es una mayor rapidez en la recuperación de la copia digital deseada, que dependerá de los metadatos asociados a dicha imagen.

También existen sistemas de Reconocimiento Óptico de Caracteres (OCR) mediante los que se pueden extraer caracteres de la imagen en formato digital y, sobre ellos, realizar búsquedas a texto completo sin embargo estos sistemas no son válidos para gran parte del fondo histórico [3] puesto que se conserva en diferentes tipos de escritura anteriores a la humanística actual, algo sobre lo que se profundizará más adelante.

Por último, la necesaria migración a otros soportes o cambios de formato es mucho más sencilla y rápida al procesar los datos digitalmente; en definitiva, la incorporación de las Tecnologías de la Información y la Comunicación en los archivos históricos puede aportar relevantes mejoras tanto en la conservación de los mismos como en su difusión.

2.1 Preproceso de imagen

Muchos textos antiguos sufren numerosas imperfecciones como el ruido impulsional, diferencia de iluminación en la imagen, manchas de tinta de la cara anterior, manchas de humedad, etc. Estas imperfecciones dificultan su inteligibilidad sobremanera, de forma creciente, por regla general, cuanto más antiguo es el archivo.

Figura 1 : Ejemplo de documento manuscrito histórico

Además, hay que añadir otro tipo imperfecciones [3] como partes de la página tachadas, diferentes tipos de caligrafía usadas, etc. Para obtener un resultado final bueno en los

procesos de digitalización de archivos históricos es necesario obtener una imagen con calidad suficiente para que pueda ser usada en las siguientes partes del proceso de segmentación de palabras.

Dentro de esta etapa de preproceso de imagen el primer elemento de los esquemas típicos [7] suele centrarse en tratar de minimizar el ruido presente en la misma, corregir el ángulo de escritura de la palabra con respecto a la dirección horizontal (*skew*), etc.

Actualmente existen diversas formas de eliminar el ruido presente en imágenes siendo usados de forma común los filtros bilaterales, donde el valor de intensidad en cada píxel de una imagen se sustituye por una media ponderada de los valores de intensidad de los píxeles cercanos; o el filtro de media, una técnica de filtrado no lineal digital donde se reemplaza cada entrada con la mediana de los píxeles vecinos. Además también es un problema común de este tipo de archivos el ruido impulsional que se produce debido a los píxeles de una imagen son muy diferentes en color o intensidad a los píxeles circundantes.

Asimismo, en determinados textos es necesario hacer más fino el trazo con el que está escrito de cara a obtener los resultados óptimos en las siguientes etapas de reconocimiento. Los algoritmos de adelgazamiento modifican una imagen binaria conservando la topología original (extensión y conectividad) mientras eliminan la mayoría de píxeles redundantes de la imagen de texto.

La forma más común de procesar las imágenes es minimizando el ruido o imperfecciones en determinados sectores de interés de la misma o que permitan efectuar cualquier tipo de filtrado en ella; un ejemplo de aplicación es la Toolbox de Procesado de Imagen integrado en MATLAB.

Por otro lado, en lo relativo a la detección del *skew*, un método extendido para este efecto [5] consiste en la proyección paralela en la dirección de alineamiento de la palabra y determinar la varianza en el número de píxeles negros, correspondientes a la escritura, por línea proyectada. Cada línea proyectada en la imagen colisionará con casi ningún píxel negro (puesto que pasará en el espacio interlineal) o bien con muchos píxeles negros debido a que coincide con la dirección en la que están escritas las líneas de texto. La acumulación de píxeles negros y, entre ellos, la ausencia de ellos, al no colisionar ninguna línea proyectada con texto por la inclinación del mismo detecta que la línea de texto no está alineada con la horizontal, es decir, hay presencia de *skew*.

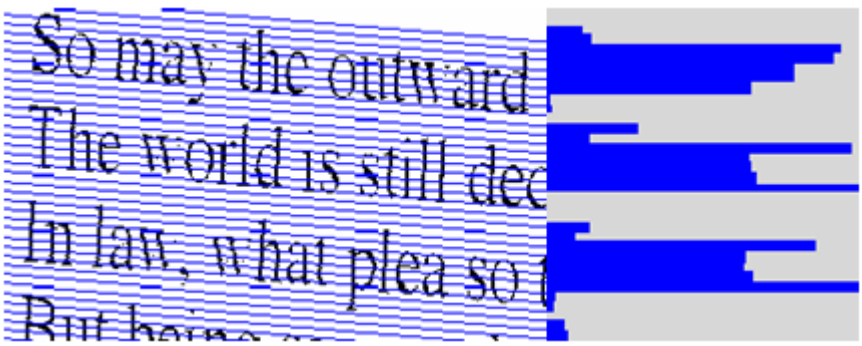


Figura 2: Detección de skew por proyección paralela a las líneas de texto

En cambio, las proyecciones oblícuas pasarán indistintamente por líneas de texto y por los espacios entre las líneas, lo que no evidencia la detección de una inclinación en el texto, por lo que la distribución obtenida será aproximadamente uniforme, a diferencia del ejemplo anterior.

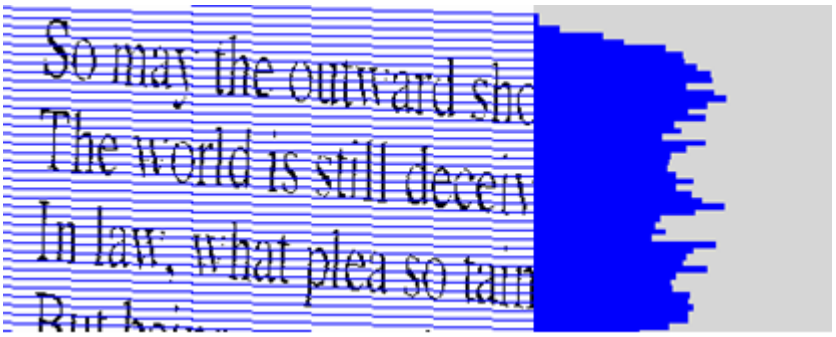


Figura 3: Detección de skew por proyección oblícua

2.2 Segmentación a nivel de palabra

En lo relativo a la extracción de las palabras de un texto manuscrito histórico existe un una gran dependencia con la parte anterior, en la que se debe haber hecho un preproceso óptimo y también se debe tener en cuenta otros aspectos como palabras que se solapan o líneas de texto adyacentes que lo hacen.

Para colecciones de textos históricos manuscritos escritos por un mismo autor las imágenes suelen ser parecidas, y siempre presentan en mayor o menor medida una serie de características comunes: baja calidad y a diferencia de los documentos impresos hay variación en la forma en que cada palabra se ha escrito, por lo tanto es un área de la investigación que presenta grandes retos.

Los algoritmos de segmentación a nivel de palabra se basan en la relación geométrica de componentes adyacentes. Actualmente, la metodología más usada incluyen dos etapas [7]: la primera se encarga del cálculo de las distancias de un componente adyacente en la línea de texto de la imagen mientras que la segunda etapa se encarga de la clasificación de las distancias previamente calculadas como espacios entre palabras o caracteres.

Para el primer paso se tiene en cuenta la métrica de la distancia Euclídea y la métrica de la envolvente convexa. La clasificación de las distancias calculadas es efectuada usando un método de técnicas populares de clustering no supervisado: la Mezcla de Gaussianas. También es necesario recalcar que las soluciones varían en función de múltiples variables como la caligrafía utilizada, el año de escritura del texto, el idioma, etc. Para este fin, la utilización de un Reconocedor Óptico de Caracteres (OCR) está lejos de dar resultados con un índice alto de éxito [3] puesto que los textos históricos manuscritos presentan por lo general degradación y formas estrambóticas en la caligrafía utilizada lo que dificulta sobremanera la aplicación de este tipo de técnicas en dichos textos.

Figura 4: Ejemplo de degradación en un texto histórico manuscrito

2.3 Clasificación de palabras.

La investigación realizada en este campo indica que el buen rendimiento de un algoritmo indicador de similitud de imágenes de palabras puede lograrse mediante técnicas que varíen el *skew*, cambien el tamaño y alineen dos imágenes de palabras candidatas y las comparen píxel a píxel.

Ejecutar un algoritmo de coincidencia puede ser poco eficiente con colecciones de tamaño grande, por lo que las técnicas de poda (que pueden descartar coincidencias poco probables) son técnicas extendidas. Además de esta técnica, también se usa el algoritmo *Dynamic Time Warping* [6] porque ofrece flexibilidad adicional para compensar las variaciones de la escritura a mano.

2.3.1 Técnicas de poda.

Esta es una técnica rápida para determinar si un par de imágenes son muy diferentes o guardan cierta similitud entre sí. En la aplicación de esta técnica en pares de palabras se tiene en cuenta el área y la relación entre sus cuadros delimitadores. La idea principal es comparar dos imágenes, con rasgos similares como el área del marco de la imagen que contiene la palabra (bounding box), que serán posteriormente comparadas.

2.3.2 Dynamic Time Warping

Es una técnica usada para calcular la distancia entre dos series. La primera de ellas es una lista de muestras tomadas de una señal (*features* de la imagen), ordenadas cronológicamente según fueron obtenidas.

Un enfoque sencillo para calcular las distancias podría ser la de un nuevo muestreo de uno de ellos y a continuación, comparar las series muestra a muestra. El inconveniente de este método es que no produce resultados intuitivos, ya que se compara muestras cuya correspondencia no sería la adecuada.

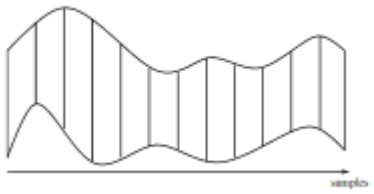


Figura 5: Series no alineadas

El algortimo de *Dynamic Time Warping* resuelve este problema entre intución y el cálculo de la distancia mediante las alineaciones óptimas entre los puntos de muestreo de ambas series. La alineación es óptima en el sentido en el que minimiza la distancia acumulada que consiste en distancias “locales” entre muestras alineadas.

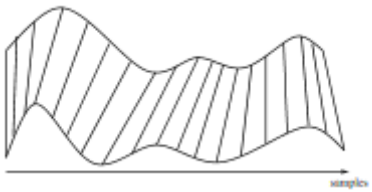


Figura 6: Series alineadas usando DTW

El procedimiento se llama "*Time Warping*" ("Distorsión de tiempo") porque distorsiona los ejes de tiempo de las dos series [6] de una forma en la que las muestras parecen tener la misma posición en un eje común de tiempo. La distancia DTW entre dos series x1…xm y y1…yn es D(M,N), que son calculadas en un enfoque de programación dinámica.

Mientras que el *slant* y el ángulo de inclinación con el que una persona escribe suele ser constante, el espacio entre caracteres está sujeto a más variaciones. DTW ofrece una forma más flexible de compensar dichas variaciones. Para llevar a cabo esto, primero hay que normalizar el *slant* y el ángulo de *skew* de las imágenes candidatas para compensar las variaciones entre palabras; posteriormente se extraen cuatro *features* por cada columna de imagen y son combinadas en una única serie de tiempo compuesta por varias muestras.

Las imágenes con las que se opera están representadas en una escala de grises de 256 niveles de intensidad. Antes de extraer los *features* de la imagen, las variaciones entre palabras como los ángulos de *skew* o *slant* deben ser detectados y normalizados. Todas los *features* de las columnas que se describen a continuación son normalizados en el rango [0…1]. Seguidamente, la proyección del perfil captura la distribución de tinta en una de las dos dimensiones de la imagen de la palabra. La proyección vertical es calculada sumando el valor de la intensidad en cada columna de la imagen por separado.

$$pp(I,c) = \sum_{r=1}^{h} (255 - I(r,c)).$$

Figura 7: Fórmula para el cálculo de la proyección vertical

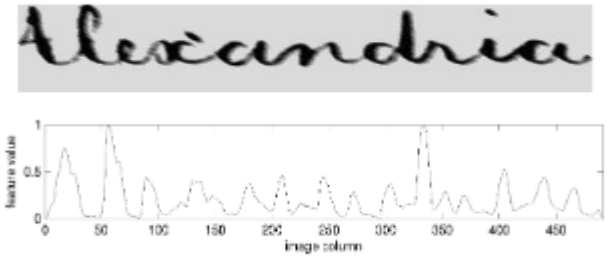


Figura 8: Proyección de la imagen normalizada, filtrada y sin skew

Debido a las variaciones en la calidad de las imágenes escanedas las proyecciones de los perfiles no varían generalmente en el mismo rango; para poder hacer una comparación entre ellos se normalizan en el intervalo [0…1]. Para capturar la parte de la estructura interior de una palabra, en este método se registra el número de transiciones entre el fondo y la tinta en una columna de la imagen como el último feature.

Los resultados de la aplicación de esta técnica han sido probados y produce mejores resultados que otras técnicas (como Mapeo de la Distancia Euclídea, Técnica de Scott & Longuet-Higgins, etc.) ofreciendo una precisión del 65%; además ofrece la ventaja de ser mucho más rápida que otros métodos examinados. El trabajo futuro sobre esta técnica debe ser mejorar su precisión así como su velocidad, para ello una opción podría ser incrementar el número de features para que se diferencien mejor unas palabra de otras.

3 Diseño

3.1 Arquetipo de la herramienta.

En este apartado definiremos los componentes integrados en la herramienta objetivo de este Trabajo de Fin de Grado, donde se distinguen tres bloques.

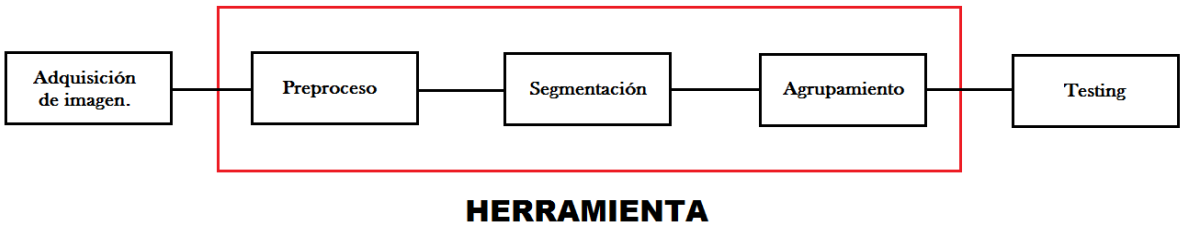


Figura 9: Esquema de diseño de la herramienta

El primero de ellos se encarga del preproceso de la imagen, con múltiples opciones configurables para efectuarlo según las necesidades de la obra que se quiera tratar puesto que al presentar cada texto imperfecciones de diferente naturaleza según múltiples variables (la época en la que fue escrito, el número de escribanos, etc.) es necesario calibrar previamente este bloque con diferentes muestras de páginas de texto.

Esta calibración de la herramienta es algo básico para el buen funcionamiento de la misma, es decir, para obtener imágenes en esta etapa con la mínima cantidad de impurezas posibles, algo necesario para el correcto funcionamiento de los siguientes bloques. Dicho proceso se llevará a cabo con una muestra de páginas pertenecientes a una determinada obra.

Adicionalmente, dentro de este primer bloque se hará un clasificador automático en función de la calidad de las páginas de las obras completas donde se ahorrará coste computacional descartando imágenes en función de un umbral de calidad prefijado con una muestra de archivos de páginas cuyas imperfecciones hacen que los métodos generales aplicados a las páginas obra de forma general no sean válidos o presenten unos resultados con un índice de éxito mucho más bajo que la media, para lo que sería necesario tratar de forma individual cada página debido a las singularidades que presentan, algo que hace imposible obtener buenos resultados de las mismas tratándolas de forma genérica.

El segundo bloque de la herramienta es el encargado de segmentar las páginas de imágenes de texto a nivel de palabra para generar una base de datos de imágenes de palabras sobre la que luego se efectuarán diversas técnicas de clasificación de las palabras que conforman la obra completa. En este bloque primero se procederá a segmentar las líneas de texto

presentes en cada página para posteriormente hacerlo con las palabras que forman dicha línea.

El tercer bloque se encargará de agrupar las imágenes de palabras extraídas de una página completa de texto probando diferentes técnicas.

Todos los módulos se integrarán en una interfaz gráfica de usuario (GUI) dado que permiten un control sencillo de las aplicaciones de software, lo que elimina la necesidad de aprender un lenguaje de programación y escribir comandos a fin de ejecutar una aplicación, algo esencial en este proyecto dado que se busca generar una herramienta manejable por gente de diversas disciplinas sin conocimientos especializados en informática.

3.2 Arquitectura del sistema

En este apartado se describirán los detalles específicos entorno a la arquitectura diseñada para la herramienta, en la que existen tres bloques diferenciados: el módulo de preproceso, el de segmentación y el de agrupamiento de imágenes de palabras.

3.2.1 Módulo de preproceso de imagen.

Para una aplicación efectiva de las diversas técnicas incluidas en el preproceso es necesario hacer una clasificación automática previa en función de la calidad de las imágenes donde se separarán aquellas cuyas imperfecciones son muy significativas, es decir, las de inteligibilidad muy baja, de aquellas que, si bien presentan imperfecciones, no sobrepasan un umbral fijado tras una serie de pruebas con muestras aleatorias para entrenar el clasificador automático.

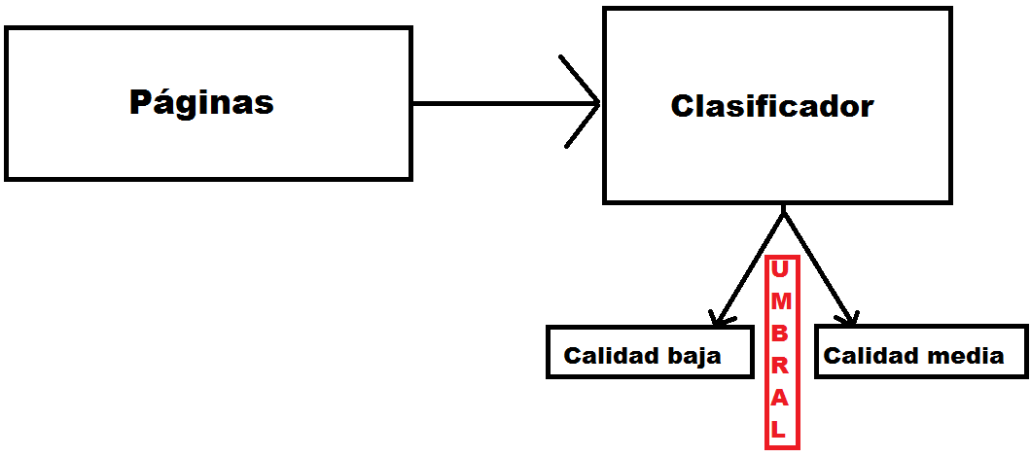


Figura 10: Clasificación de las páginas del documento en función del umbral

Seguidamente, para esta parte de la herramienta se incorporan diferentes opciones de preproceso de imagen dentro de un menú dando la opción de elegir opciones de forma individual o combinada para ver cuál da mejor resultado y qué imagen es la que posee más calidad tras efectuar diferentes filtrados sobre una muestra; dicha configuración puede ser exportada para tratar todas las imágenes de un directorio del mismo modo.

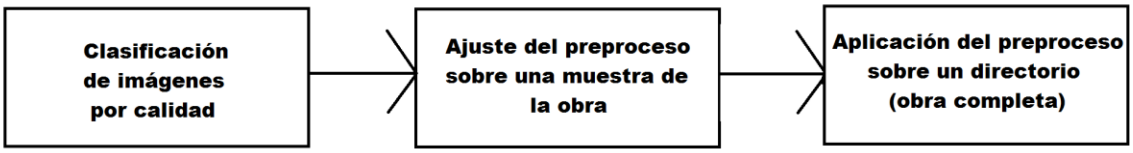


Figura 11: Proceso de parametrización de la aplicación

Debido a que en muchos textos antiguos se suelen producir imperfecciones similares se da la opción de elegir opciones predeterminadas que agrupan las que han dado mejores resultados en imágenes de calidad similar.

Las opciones que se pueden escoger dentro de este menú son válidas para cualquier formato de la imagen de origen (RGB, binarias, etc.). En el caso del formato más complejo al poseer más de una dimensión (RGB), se tendrá que seleccionar las opciones de escala de grises y binarizado para dejar la imagen en un formato unidimensional, lo que facilitará su procesamiento posterior. También se podrá seleccionar las opción de corrección de los ángulos de las letras y palabras respecto a la dirección horizontal, es decir, corrección de *slant* y *skew*.

Por último, habrá una opción para filtrar el ruido entre líneas de texto y en los bordes de la página, debido a que es donde mayor cantidad de ruido se suele acumular, en base a un filtrado hecho midiendo la acumulación de píxeles por fila y columna lo que da como resultado un histograma de concentración de píxeles cuyos umbrales pueden ser modificados para variar la tolerancia del borrado.

Asimismo, existe una característica común a la mayoría de imágenes utilizadas, la presencia de un marco negro consecuencia del proceso de escaneado; dentro de este módulo también habrá una opción para eliminar dicho marco de píxeles negros que afecta en la extracción de medidas de cada página.

Finalmente, el afinado del trazo con el que están escritos los textos es otra opción presente en esta etapa debido a que en determinadas ocasiones es necesario el adelgazamiento del trazo con el que está escrito una obra de cara a mejorar las posteriores etapas de segmentación y agrupamiento.

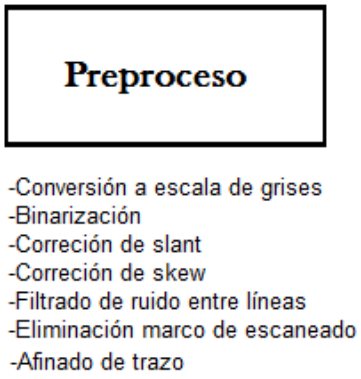


Figura 12: Menú de opciones de preproceso

Una vez configurada la herramienta y analizado los resultados sobre una muestra de imágenes se guarda esta configuración para aplicarla a las diferentes páginas del texto y generar por tanto imágenes con mayor grado de inteligibilidad respecto a las que se tenían en principio; dichas imágenes serán almacenadas en el directorio seleccionado por el usuario en la configuración previa de la herramienta y con esto se podrá transferir una imagen tratable al siguiente bloque de la herramienta, habiendo descartado aquellas que no hayan superado la criba de calidad del clasificador automático.

3.2.2 Módulo de segmentación a nivel de línea y de palabra.

Tras haber mejorado la calidad de la imagen de la página del documento original en el bloque anterior, el usuario ahora procederá ahora a configurar el bloque destinado a la extracción de palabras de la página de texto manuscrito para obtener imágenes de palabras segmentadas extraídas de la misma página que posteriormente serán agrupadas por similitud en el siguiente bloque.

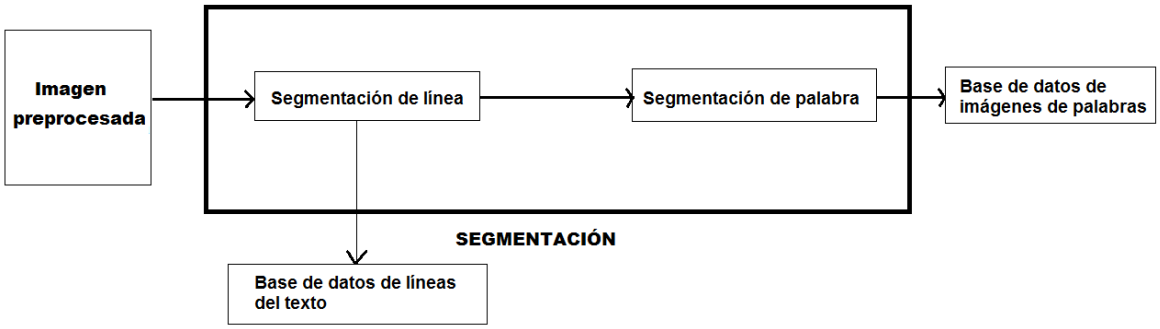


Figura 13: Esquema de obtención de bases de datos de imágenes a nivel de línea y palabra

Primeramente, partiendo de la imagen preprocesada, se procede a segmentar todas las líneas de la imagen de texto para, posteriormente, segmentar a nivel de palabra sobre un conjunto mucho más reducido de píxeles sobre el que la operativa es más sencilla. Las líneas extraídas se guardan en un directorio indicado por el usuario generando una base de datos de líneas de texto segmentadas.

Una vez hecho efectuado el proceso de segmentación de línea se detectan las palabras en las mismas haciendo uso del algoritmo *Connected-Component Labeling* y posteriormente se extraen en función de si son mayores de un umbral prefijado por el usuario de la aplicación (para evitar que se introduzcan elementos no deseados producto del ruido presente en la imagen que se haya podido detectar erróneamente como una palabra) generando una base de datos de imágenes de palabras en un directorio indicado por el usuario. Dentro de este mismo bloque, de cara a la evaluación del nivel de eficacia del proceso se extraerán las medidas de las palabras y líneas segmentadas en un archivo de texto, para posteriormente compararla con los resultados extraídos manualmente.

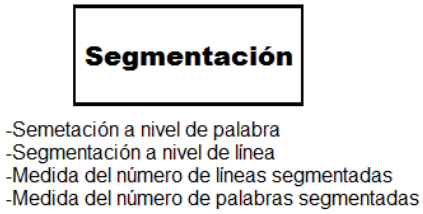


Figura 14: Menú de opciones de segmentación

3.2.3 Módulo de clasificación de imágenes de palabras.

En el último bloque que conforma la herramienta se agruparán por similitud las palabras segmentadas en la etapa anterior. El objetivo de esto es reducir de forma notable el tiempo en el que se realiza de forma habitual la transcripción de textos históricos agrupando las imágenes de palabras segmentadas y transcribiendo una de ellas. La nomenclatura de las imágenes de palabras debe indicar en qué página del documento está presente dicha palabra y además en qué línea y posición dentro de la línea.

A todas las palabras dentro de un mismo grupo se les añade un marcador de tal modo que con hacer la transcripción de una palabra el resto estarán análogamente transcritas ahorrando así recursos al no tener que analizar todas las palabras identificadas como parte de ese agrupamiento.

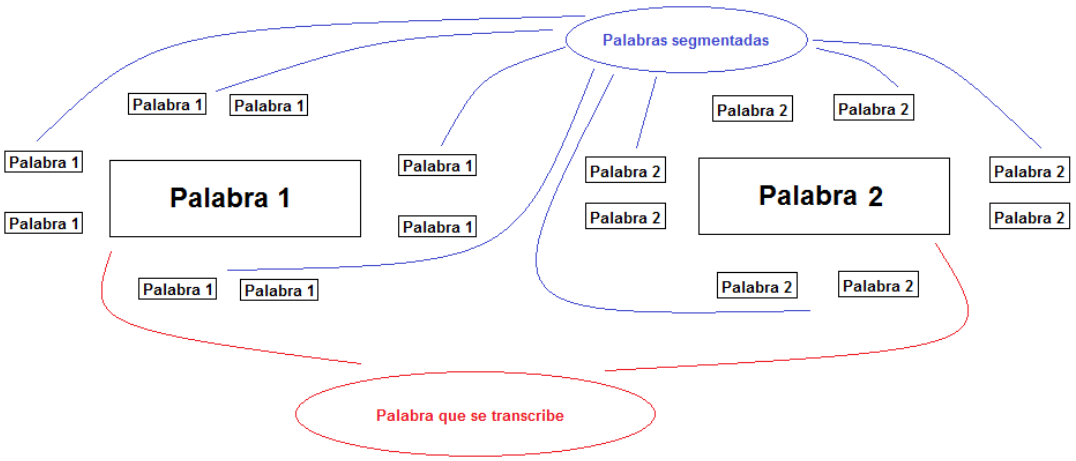


Figura 15: Proceso de agrupamiento de imágenes de palabras

Una vez hecha la transcripción de todas las imágenes se tendrá que hacer el proceso inverso, es decir, escribir en un formato de texto, en la posición en la que estaba la parte de la imagen correspondiente a una palabra extraída la misma palabra transcrita.

El grado de similitud entre las imágenes de palabras segmentadas anteriormente se obtendrá usando técnicas de poda que cuantificarán en función de diversas variables como las dimensiones del marco y coincidencia de píxeles entre otras el parentesco de una imagen de palabra con otra.

4 Base de datos

Para llevar a cabo este TFG se ha partido de una base de datos de Actas del Puerto de Tarragona; dicho repositorio se creó con el objetivo de custodiar el patrimonio documental de la administración portuaria desde sus orígenes hasta la actualidad. Se abrió al público el día 4 de julio de 1990 con motivo de los actos de celebración del Bicentenario del Puerto Moderno. En aquel momento se le denominó Archivo Histórico, aunque, con el tiempo, acogió también la documentación más reciente, llegando a ser el Archivo General del organismo con documentación, tanto histórica como administrativa.

Con la creación de este equipamiento cultural, el Puerto de Tarragona, se convirtió en uno de los pioneros en el campo de la recuperación del patrimonio escrito y documental, actuación que iba fuertemente vinculada a la inquietud por los temas culturales.

El Archivo del Puerto está al servicio de la administración portuaria, pero también de los investigadores y del público en general por lo que con los años se ha convertido en una referencia para todos aquellos interesados en indagar sobre el pasado del Puerto, las personas que han trabajado en él, los proyectos y las obras llevadas a cabo a lo largo del tiempo, la actividad económica que ha generado, o la actividad cultural, la relación puerto-ciudad, etc.

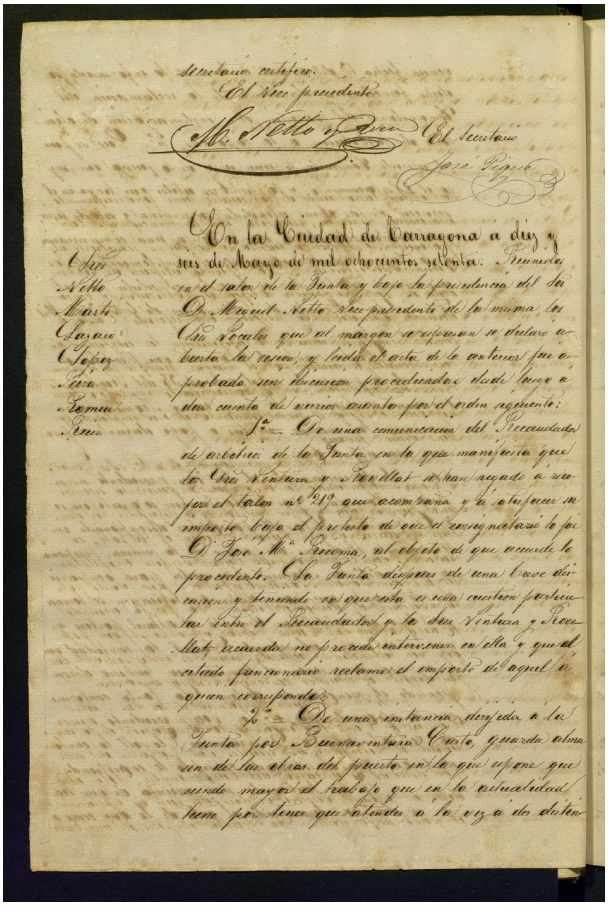


Figura 16: Página de un Acta del Puerto de Tarragona

La documentación más antigua que se conserva en el Archivo referente al Puerto de Tarragona data del 1790, año en que se iniciaron las obras del puerto moderno. Desde aquella fecha hasta la actualidad, la documentación ha aumentado paulatinamente, hasta llegar a constituir un fondo considerable a partir de cuyo estudio, es posible reconstruir la historia del Camp de Tarragona y el papel que el puerto ha representado en él, tanto económica como socialmente.
El Archivo del Puerto forma parte del colectivo de archivos portuarios del Estado español y de algunos de los Grupos de Trabajo formados a partir de las Jornadas Técnicas de Archivos Portuarios.

La obtención de la base de datos de imágenes con las que se trabajará en este TFG se ha obtenido ejecutando la aplicación PDFMate que extrae de los archivos en formato PDF de las actas todas las imágenes de las páginas de la misma (una imagen por página), dichos ficheros JPG contienen el número del acta y el número de página de la misma.

Figura 17: Programa PDFMate

Tras esta conversión cada página del acta (siendo un acta un archivo PDF completo) es extraída en un directorio con el nombre de la misma en formato JPG. Para poder efectuar evaluaciones sobre los resultados obtenidos es necesario cambiar la nomenclatura de cada página del acta dentro del directorio donde se encuentran; el formato que se le dará es el número identificativo del acta seguido del número de página dentro de la misma (IDacta_numPágina).

Al estar tratando con un gran volumen de imágenes se ha recurrido al uso del programa Bulk Rename Utility de cara a agilizar el proceso de renombrad de los elementos de la base de datos que se usará a lo largo del proyecto.

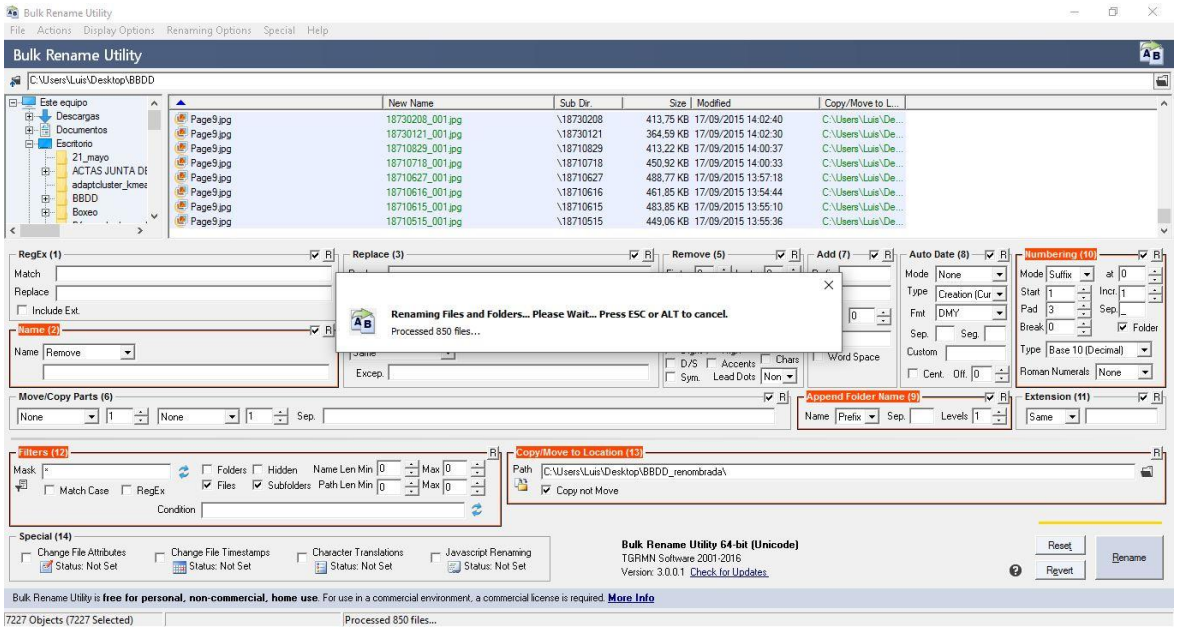


Figura 18: Programa Bulk Rename Utility

En las páginas originales se ha efectuado un análisis cualitativo en las que se han detectado diferentes prototipos de página con distinto nivel de ruido generado por el escaneo; además se dispone en formato Word la transcripción revisada por expertos de las actas manuscritas, algo fundamental para probar la efectividad de la herramienta informática. Dichos archivos Word se han pasado a formato TXT y se han corregido para que coincida exactamente con el número de palabras que hay presentes en cada línea.

4.1 Adquisición de imagen

Para digitalizar las imágenes y obtener resultados con una calidad suficiente para poder procesar posteriormente las páginas de este tipo de documentos es necesario cumplir una serie de condiciones detalladas a continuación.

Primeramente se debe contar con un escáner occipital con las medidas de calidad de escaneado adecuadas como la resolución (la densidad de información que las muestras de escáner, generalmente expresados en puntos por pulgada) y la profundidad de bits (la cantidad de información obtenida de un punto, que en general varía de un bit por punto para el negro e imágenes en blanco a veinticuatro bits por punto de color de alta calidad).

Además hay algunas reglas generales para el escaneo de textos históricos; en el caso de tener intención de extraer el texto de la página usando reconocimiento óptico de caracteres (OCR) en lugar de mostrar los escaneos como imágenes de las páginas, es necesario únicamente un bit blanco y escaneando a una alta resolución de 300 a 600 puntos por pulgada, algo inviable debido a la complejidad en la escritura presente en los textos de las actas que se van a tratar.

Sin embrago, a la hora de obtener las imágenes de texto es necesario evaluar el volumen de documentos del proyecto ya que esta es una etapa que, debido a la fragilidad de los documentos, puede consumir una parte notable del tiempo del proyecto ya que muchas obras no se pueden manipular con soltura; tienen una apertura máxima en grados para

evitar que se dañen o es necesario diseñar una superficie de escaneo que no sea plana debido a estas restricciones, a menudo es más económico externalizar este tipo de trabajo.

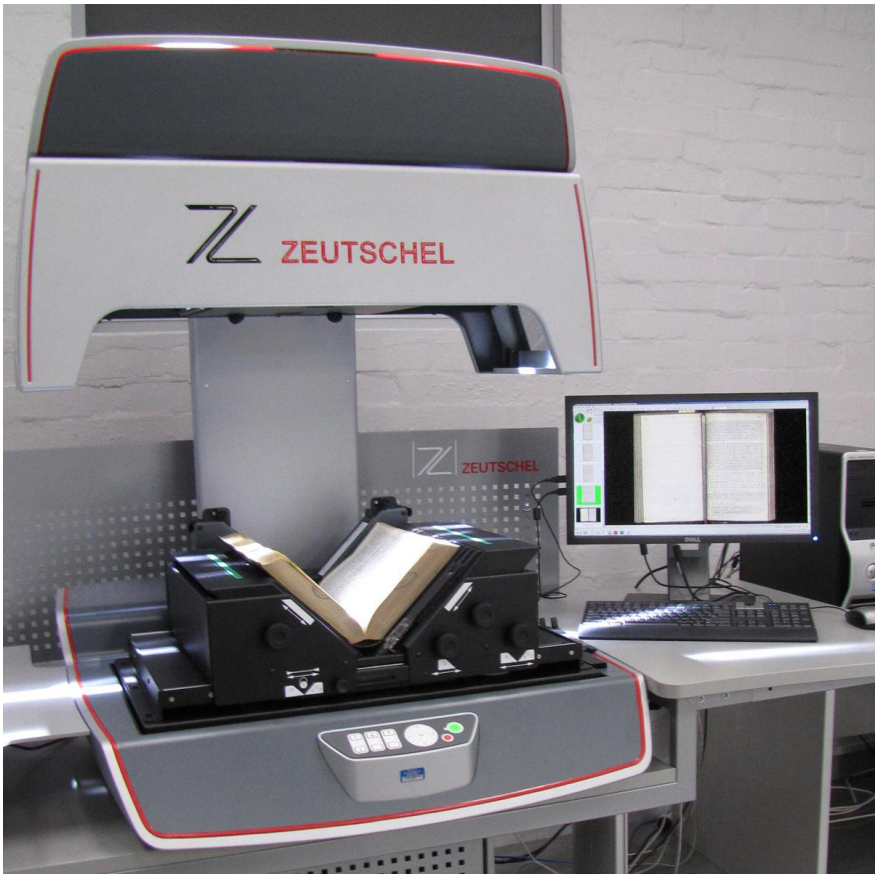


Figura 19: Escáner occipital

La forma más eficiente de llevar a cabo este proceso es utilizando un reconocedor óptico de caracteres (OCR) debido a que es una solución rápida, barata y sobre todo automática; no obstante no es una solución que se pueda aplicar en un proyecto de transcripción de textos históricos dado que hasta el mejor OCR tiene limitaciones como por ejemplo el no reconocer las letras que no son de alfabeto latino o letras pequeñas, algunos tipos de fuentes, distribuciones poco usuales de páginas, presencia de gráficos, tablas, ruido de fondo, símbolos matemáticos o químicos y, en general, pocos textos anteriores al siglo XIX. Por ello tras la adquisición de imagen, para textos históricos antiguos, no es posible digitalizar inmediatamente el documento a través del escaneado, se debe llevar a cabo una serie de pasos en función de la calidad y complejidad de la obra parametrizando software especializado.

5 Desarrollo

5.1 Herramientas empleadas

Para realizar este TFG se ha usado la herramienta de software matemático MATLAB, cuyas prestaciones básicas son: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos y la creación de interfaces de usuario (GUI).

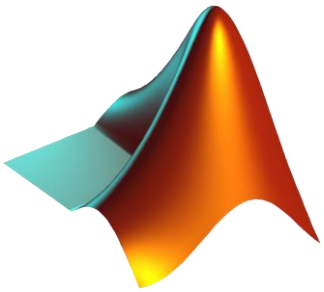


Figura 20: Logotipo de MATLAB

Al tratar con imágenes a lo largo de este proyecto, MATLAB ofrece una interacción sencilla con ellas dado que las almacena como vectores bidimensionales (matrices), en el que cada elemento de la matriz corresponde a un sólo píxel (trabajar con imágenes en Matlab es equivalente a trabajar con el tipo de dato matriz) además de traer integradas múltiples funciones que permiten tratar las mismas de forma sencilla. Adicionalmente, todo el código creado puede ser integrado en una aplicación GUI.

Las interfaces gráficas de usuario (GUI) permiten un control intuitivo de las aplicaciones de software, lo cual elimina la necesidad de aprender un lenguaje y escribir comandos para ejecutar una aplicación.

Las aplicaciones de MATLAB son programas autónomos de MATLAB con un frontal gráfico de usuario que automatizan una tarea o cálculo. Generalmente, la GUI incluye controles como menús, botones, barras de herramientas, y controles deslizantes.

Asimismo es posible crear apps personalizadas propias, incluidas las interfaces de usuario correspondientes, para que otras personas las utilicen.

5.2 Implementación

En este apartado se verá detalladamente la implementación software de los componentes que forman la herramienta que se parametrizará para su aplicación sobre las Actas de Tarragona.

5.2.1 Preprocesado

5.2.1.1 Clasificador

Para aumentar la eficiencia del proceso y descartar antes de poner en marcha la segmentación de aquellas páginas con una calidad muy por debajo de la necesaria para

Figura 21: Clasificación de páginas por calidad

poder efectuar una correcta segmentación a nivel de palabra, es decir, que se obtengan palabras inteligibles, se ha implementado un clasificador de las imágenes de las actas.

Debido a la naturaleza cualitativa de este problema, tomando una muestra de las actas se han fijado unos umbrales representativos de cada tipo de calidad (baja, media y alta); únicamente se procederá a segmentar aquellas de calidad media y alta.

En la siguiente imagen se presenta un ejemplo de los diferentes tipos de calidad de imagen presentes en la obra.

Las imágenes de baja calidad se caracterizan por tener un gran ruido de fondo presente por el filtrado de la tinta de la página anterior a lo largo del tiempo lo que dificulta el procesado de dicha imagen ya que a la hora de binarizar las mismas gran parte del ruido es interpretado como el texto que se busca introduciendo errores en la segmentación mientras que en aquellas páginas de calidad media y alta los resultados de la segmentación tienen un índice de éxito mucho mayor.

```
segun (numPixelsNegros ) {
      caso buenaCalidad
         Clasificar página como buena.
      caso calidadIntermedia
         Clasificar página como intermedia.
      caso malaCalidad
         Descartar página
      defecto:
         No clasificar
}
```

Figura 22: Pseudocódigo del clasificador

Tras la clasificación se crean automáticamente tres carpetas correspondientes a sendos niveles de calidad de la página del acta donde se guardarán las imágenes.

Figura 23: Directorios de clasificación en función de la calidad

5.2.1.2 Binarización

Para obtener imágenes con calidad suficiente sobre las que poder segmentar tanto a nivel de línea como a nivel de palabra es necesario efectuar una serie de operaciones sobre la imagen original con el objetivo de eliminar la máxima cantidad de imperfecciones procedentes de la misma.

Se parte de imágenes de páginas con formato RGB, siendo una imagen RGB un formato en el que se almacena una matriz de datos de dimensiones tres dimensiones mxn que define componentes de color rojo, verde y azul para cada pixel individual.

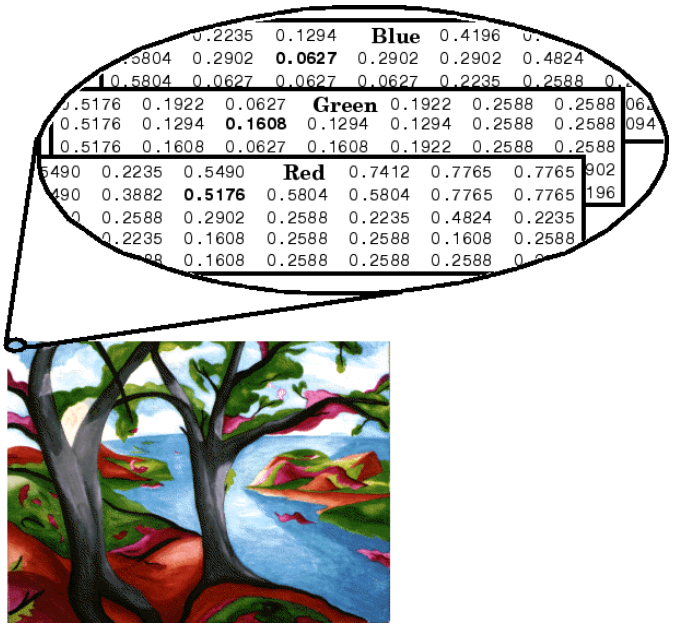


Figura 24: Ejemplo de imagen RGB

Las imágenes RGB no utilizan una paleta, a diferencia de otros formatos, ya que el color de cada pixel se determina por la combinación de las intensidades de color rojo, verde y azul almacenados en cada plano de color en la ubicación del píxel.
Los archivos de formatos de imágenes RGB se almacenan como imágenes de 24 bits, donde los componentes rojo, verde y azul son de 8 bits cada uno.

En esta etapa se empieza convirtiendo la imagen original RGB, a un formato con el que la manipulación de imágenes de texto sea más sencilla. Puesto que los colores presentes en la misma no son relevantes para las siguientes etapas, se convierte el formato RGB a una imagen en escala de grises a través de la función rgb2gray, que la convierte a escala de grises mediante la eliminación de la información de tono y saturación, manteniendo la luminancia, lo que simplifica sobremanera el tratamiento posterior de las imágenes de páginas de texto.

Posteriormente, la imagen obtenida en escala de grises se simplifica haciendo uso de la función im2bw, que convierte la imagen en escala de grises en una imagen binaria. La imagen de salida sustituye a todos los píxeles de la imagen de entrada con luminancia mayor que el nivel con el valor 1 (blanco) y sustituye a todos los otros píxeles con el valor 0 (negro). Esta gama es relativa a los niveles de señal posibles para la clase de la imagen, por lo tanto, un valor de nivel de 0,5 está a medio camino entre el blanco y negro, independientemente de su clase. Para calcular el nivel automáticamente, se puede utilizar la función de graythresh. Si no se especifica, im2bw utiliza el valor 0,5 o también se puede introducir manualmente desde la herramienta.

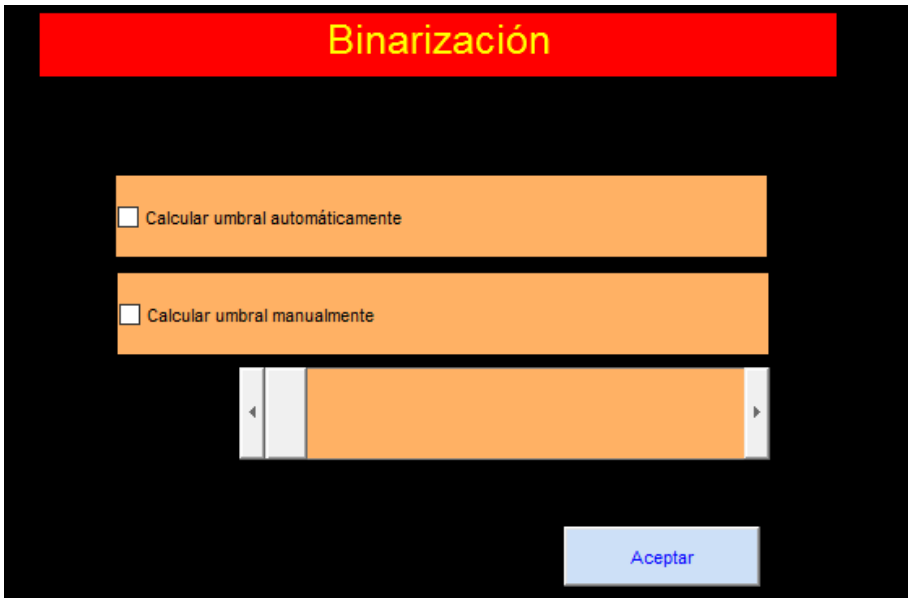


Figura 25: Ajustes de binarización dentro de la herramienta final

Se puede observar en las variables cómo estas han ido reduciendo su complejidad, siendo la variable correspondiente a la imagen original tridimensional y la de la imagen binaria unidimensional (con valores binarios).

Name	Value
imagen_binaria	<1267x843 logical>
imagen_escalaDeGrises	<1267x843 uint8>
imagen_original	<1267x843x3 uint8>

Figura 26: Tipos de imagen y matrices asociadas

5.2.1.3 Eliminación de ruido impulsional

En el tipo de ruido conocido como ruido de sal y pimienta los píxeles de la imagen son muy diferentes en color o intensidad a los píxeles circundantes. El hecho que define este tipo de ruido es que el pixel ruidoso en cuestión no tiene relación alguna con los píxeles vecinos y por lo general, este tipo de ruido afectará a una pequeña cantidad de estos.

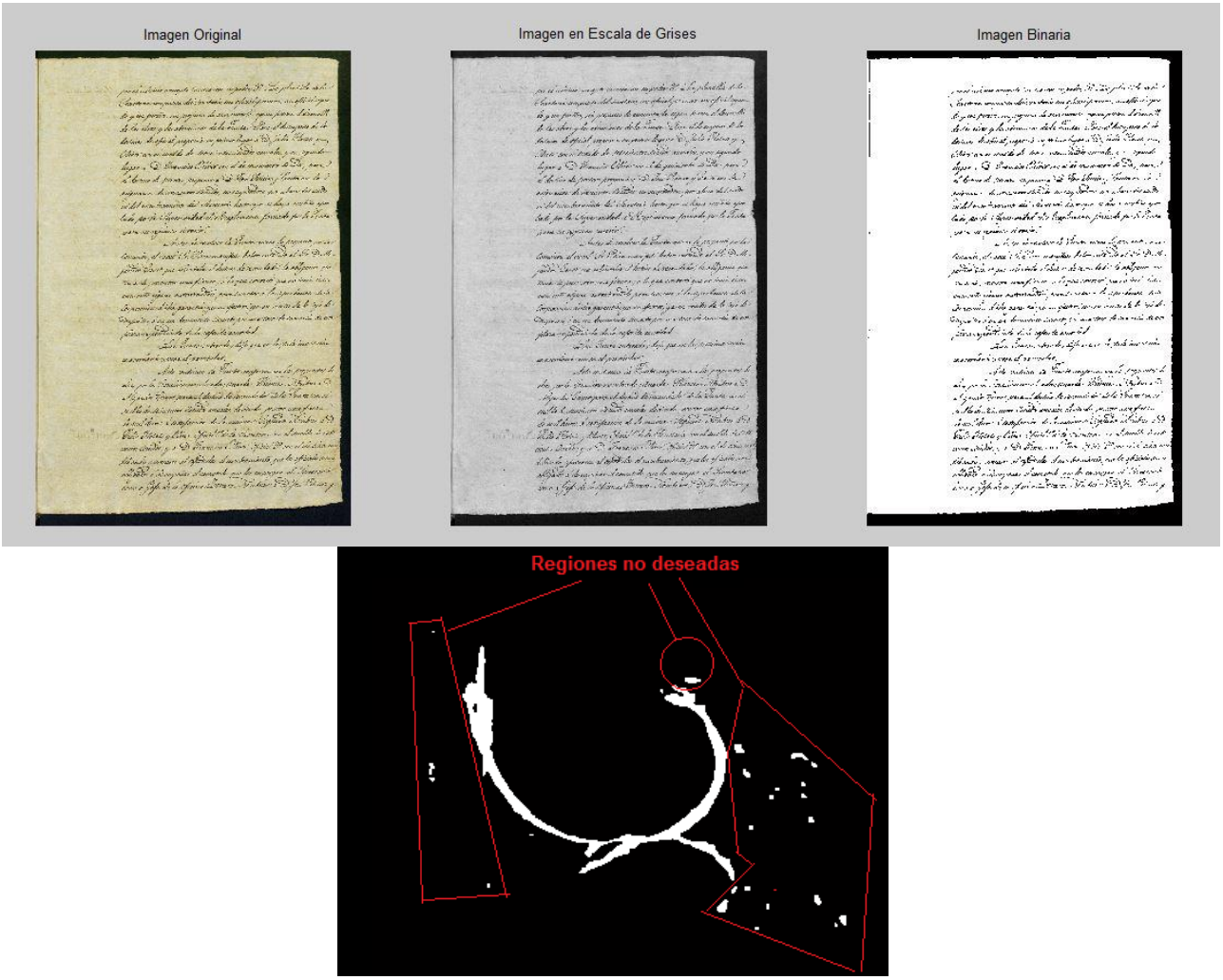


Figura 27: Ejemplo de ruido impulsional

Al ver la imagen, encontraremos puntos blancos sobre puntos negros o puntos negros sobre puntos blancos, de ahí el término sal y pimienta. Los defectos que contribuyen a este tipo de ruido son las manchas de polvo dentro de las ópticas de la cámara o escáner.

Para corregir estas imperfecciones se hace uso de la función bwareaopen de Matlab; dicha función elimina todos los componentes conectados (detectados automáticamente) que tienen menos de P píxeles; dicho umbral puede ser configurado desde la herramienta.

Figura 28: Ajuste de filtrado de ruido impulsional

5.2.1.4 Reducción del grosor del trazo.

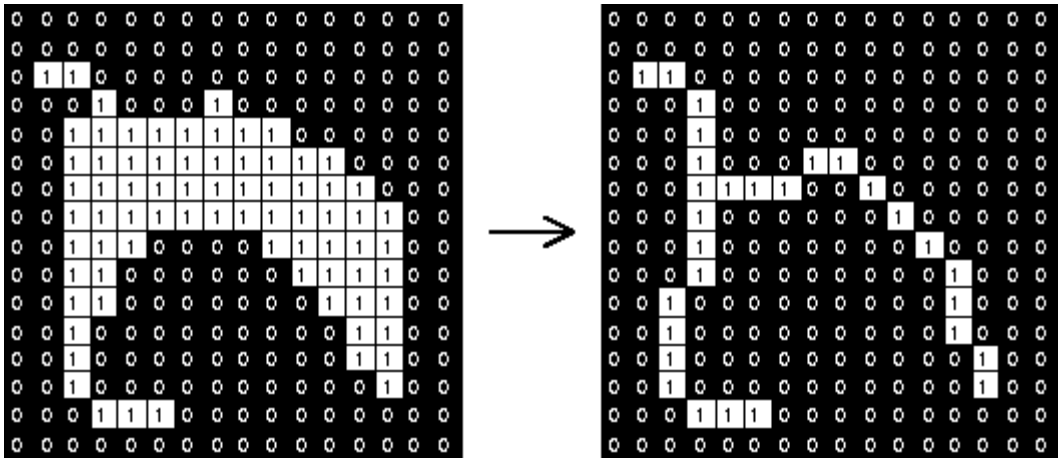


Figura 29: Afinado del trazo a nivel de píxel

Los métodos más comunes de adelgazamiento del trazo son:

-Número de conectividad

Es una medida de cuántos objetos están conectados con un píxel en particular siguiendo esta ecuación:

$$C_n = \sum_{k \in S} N_k - (N_k \cdot N_{k+1} \cdot N_{k+2})$$

Figura 5-11: Expresión utilizada para saber determinar los componentes conectados

Donde: Nk es el color de los 8 píxeles vecinos al analizado; N0 es el píxel central; N1 es el valor del píxel a la derecha del central y el resto son numerados en sentido antihorario al centro.

$$S = \{1, 3, 5, 7\}$$

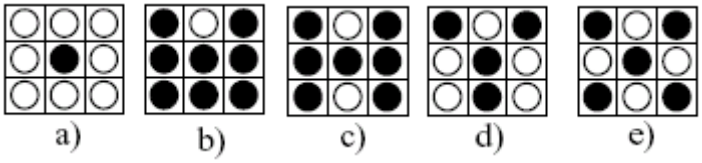
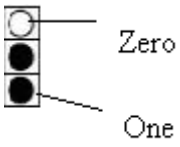


Figura 30: Explicación gráfica del funcionamiento del algoritmo



Los siguientes son los números de conectividad de la Figura 2.1:

a) Número de conectividad = 0.
b) Número de conectividad = 1.
c) Número de conectividad = 2.
d) Número de conectividad = 3.
e) Número de conectividad = 4.

- Algortimo de adelgazamiento de Zhang-Suen .
Este algoritmo esqueletización es un método en el que el nuevo valor obtenido solamente dependerá del valor iteración anterior. Es fácil y rápido de implementar. Este algoritmo se hace por dos sub-iteraciones. En la primera iteración, un pixel I (i, j) se elimina, si se cumplen las condiciones siguientes:

1. Su número de conectividad es uno.
2. Tiene al menos dos vecinos negros y no más de seis.
3. Al menos uno de I (i, j + 1), I (i-1, j), y I (i, j-1) son de color blanco.
4. Al menos uno de I (i-1, j), I (i + 1, j), y I (i, j-1) son de color blanco.

En la segunda sub-iteración las condiciones en los pasos 3 y 4 cambian.

1. Su número de conectividad es uno.
2. Tiene al menos dos vecinos negros y no más de seis.
3. Al menos uno de I (i-1, j), I (i, j + 1), y (i + 1, j) son de color blanco.
4. Al menos uno de I (i, j + 1), I (i + 1, j), y I (i, j-1) son de color blanco.

Finalmente, se eliminan los píxeles que cumplan estas condiciones. Si al final de cualquiera de las sub-iteraciones no hay píxeles que cumplan las condiciones para ser eliminados, entonces el algoritmo se detiene.

Un ejemplo de los resultados que se obtendrían con la aplicación de este algoritmo:

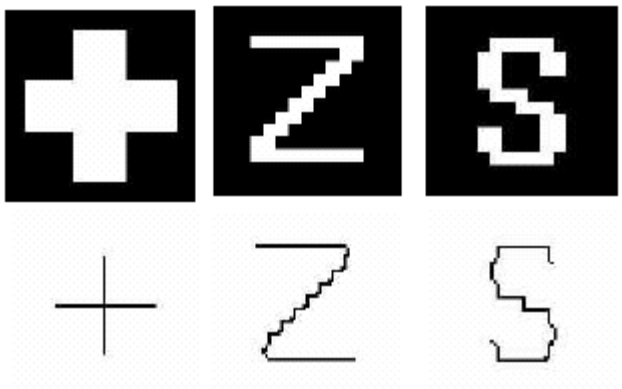


Figura 31: Resultado de la aplicación del algoritmo de afinado de trazo

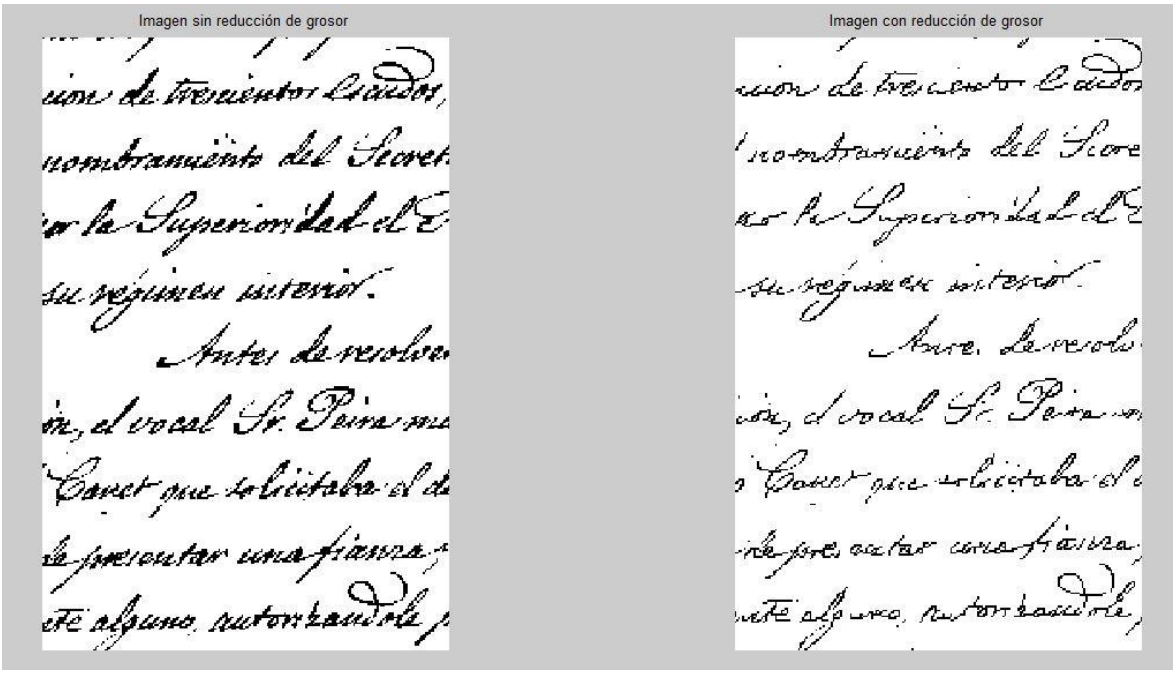


Figura 32: Aplicación del algoritmo de afinado de trazo sobre una porción de página de acta

5.2.1.5 Eliminación marco de escaneado

Para este caso particular en el que se trabaja con actas escaneadas de la misma forma se ha añadido una opción configurable desde la herramienta para eliminar los marcos negros de cada página productos del escaneo debido a que su aportación no es relevante a cualquier efecto y únicamente distorsionan cualquier etapa de preproceso y segmentación.

Para eliminar dichos marcos, de dimensiones similares a lo largo de toda la obra, cuando se clasifican las imágenes se puede obtener una medida de los píxeles negros en los extremos de la página; tras analizar todo el directorio de imágenes se puede obtener un promedio de tal modo que se elimine dicho marco adecuadamente en todas las imágenes, aunque puede ser conveniente alcanzar el compromiso de ajustarlo a la máxima dimensión, para asegurarse de que ninguna página tenga dicho marco a costa de estropear alguna palabra del texto que posteriormente será descartada por no alcanzar las dimensiones adecuadas en la segmentación o por no poder ser agrupada.

Una vez obtenidas las dimensiones de dichos marcos se procede a sustituir todos los píxeles de color negro que estén dentro de los mismos por píxeles de color blanco, coincidentes con el color blanco asignado al fondo de la página.

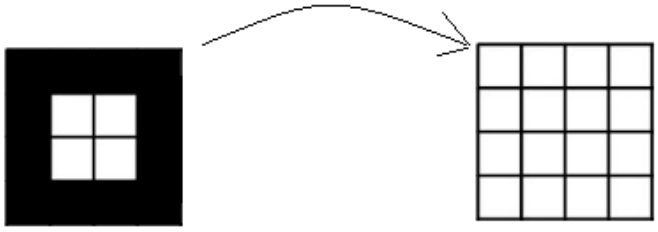


Figura 33: Explicación a nivel de píxel de la eliminación de marco procedente del escaneado, siendo cada cuadrado un píxel de una página de las Actas

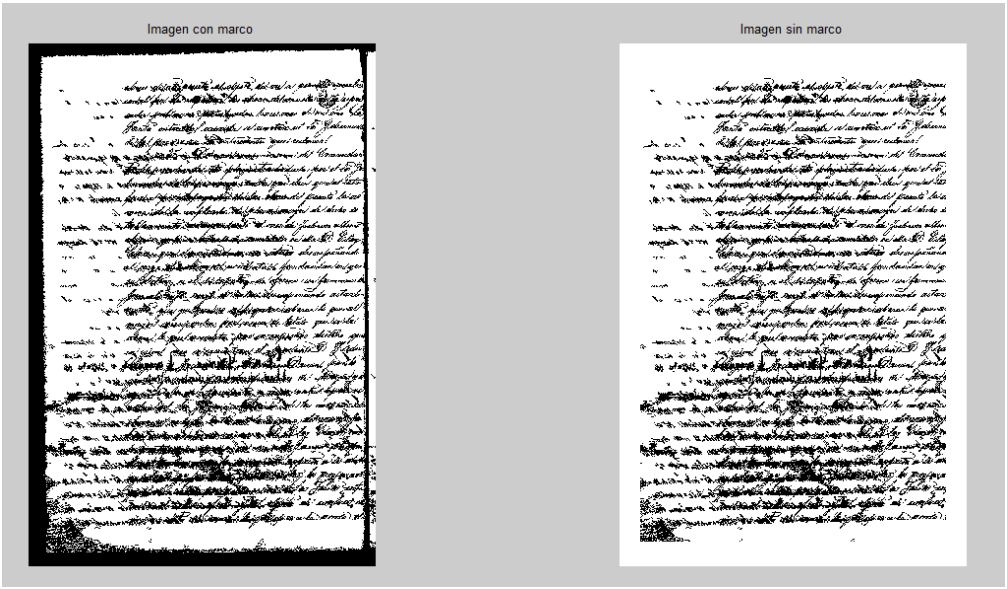


Figura 34: Resultado de la eliminación del marco procedente del escaneado

De cara a clasificar automáticamente las imágenes el marco negro procedente del escaneo estropeaba dicho proceso ya que al estar basada la clasificación en la comparación de píxeles de la página para determinar su calidad una página de gran calidad con un marco negro mayor al de la media introduciría un sesgo en la clasificación considerando esta página como de mala calidad.

5.2.1.6 Detección de slant y skew

En el caso de que la dirección de los ángulos de las palabras o la línea completa de texto no presente una dirección horizontal, es decir, supere el tamaño prefijado para la detección de una línea de texto dicha página se descartará y archivará en un directorio junto a las demás páginas que presenten una desviación respecto a la horizontal para ser tratadas posteriormente.

5.2.1.7 Filtrado del ruido entre líneas

Para la eliminación del ruido que no ha podido ser eliminado tras la aplicación de los filtrados más simples ofrecidos en la herramienta se recorre la imagen tanto vertical como horizontalmente para hallar la concentración de píxeles negros, que indican presencia de texto o ruido tras la binarización de la página; esto permite obtener la concentración de los mismos en sentido horizontal y vertical.

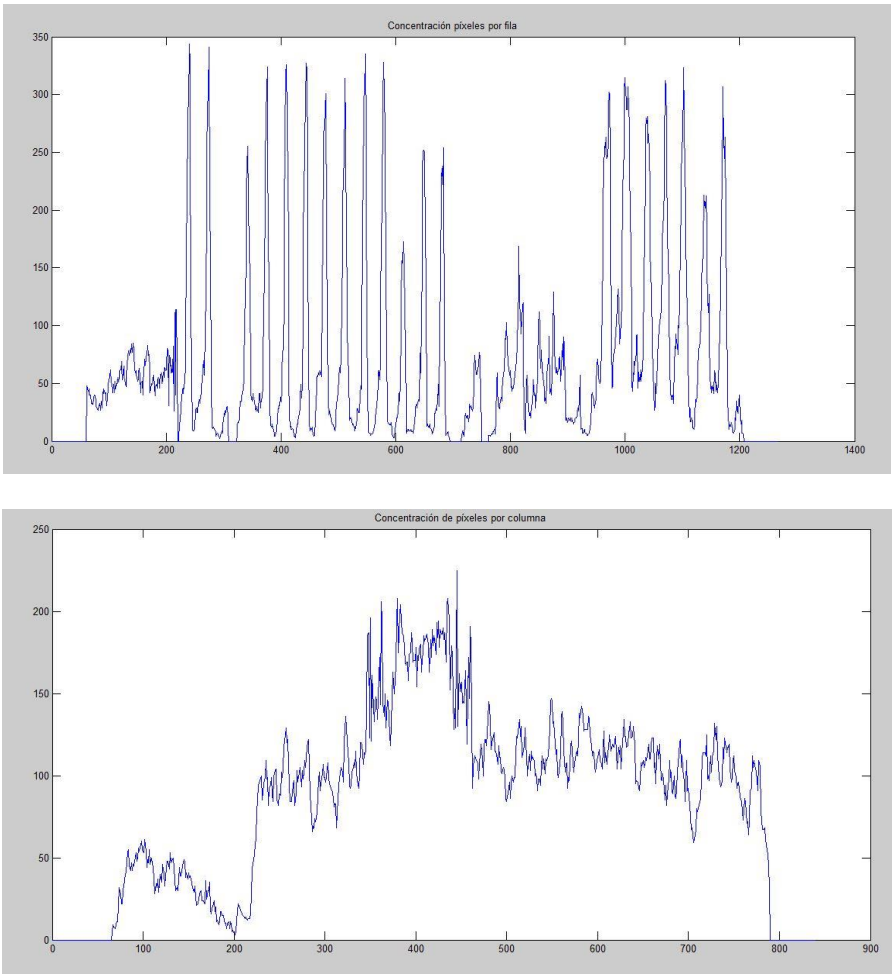


Figura 35: Representación de los histogramas obtenidos al recorrer la imagen vertical y horizontalmente

Con esta información se determina un umbral de tolerancia de borrado de aquellas líneas que no tengan la suficiente concentración de píxeles para que pueda ser considerada como una línea de texto para, de esta forma, eliminar todo aquel ruido que no se ha podido eliminar con los métodos más simples previamente aplicados. La tolerancia de borrado también puede ser configurada desde la herramienta por si es necesario ser más o menos permisivo y la importancia que tenga para las siguientes etapas el contar con la longitud de todos los caracteres íntegramente o no.

Figura 36: Configuración de la tolerancia de borrado dentro de la herramienta (GUI)

La operación de borrado de ruido entre líneas que se efectúa consiste en poner los píxeles presentes en las líneas que no entran dentro del umbral de concentración de píxeles negros suficientes para no ser considerados ruido al nivel de píxeles blancos de este modo se fusionan con el fondo de la página. Se trata de la misma operación efectuada al transformar el color negro del marco procedente del escaneado a color blanco, coincidente con el fondo de la página.

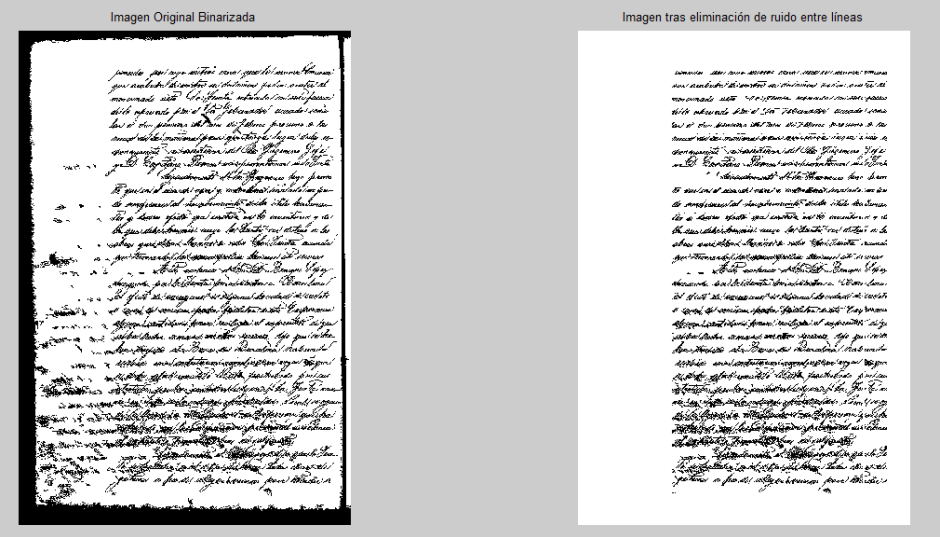


Figura 37: Resultado de la imagen tras el filtrado interlineal en comparación con la página del acta original

Finalmente, una vez implementadas todas estas opciones de preproceso de imagen se integran en el menú de preproceso de la herramienta.

Figura 38: Menú de preproceso dentro de la herramienta (GUI).

5.2.2 Segmentación

5.2.2.1 Segmentación a nivel de línea.

El primer tipo de segmentación se efectuará sobre las líneas de texto presentes en las imágenes de las actas, para ello el programa sobre una imagen filtrada de imperfecciones proveniente de la etapa anterior de preproceso se volverá a calcular la concentración de píxeles presentes en las filas de la imagen del acta para determinar dónde hay líneas de texto.

Este proceso se lleva a cabo debido a que los mejores resultados de segmentación a nivel de palabra se obtienen llevando a cabo este proceso primero con las líneas lo que evita introducir elementos difusos al algoritmo usado que detecta componentes conectados (en este caso palabras) mientras que la tasa de fallo al aplicar el mismo algortimo proporcionándole directamente la página es significativamente más elevado.

La eliminación de ruido entre líneas en la etapa anterior tiene una gran importancia para este proceso pues permite establecer que sean interpretadas como parte de las líneas de texto aquellas filas de la imagen (matriz) que estén entre dos filas de píxeles blancos.

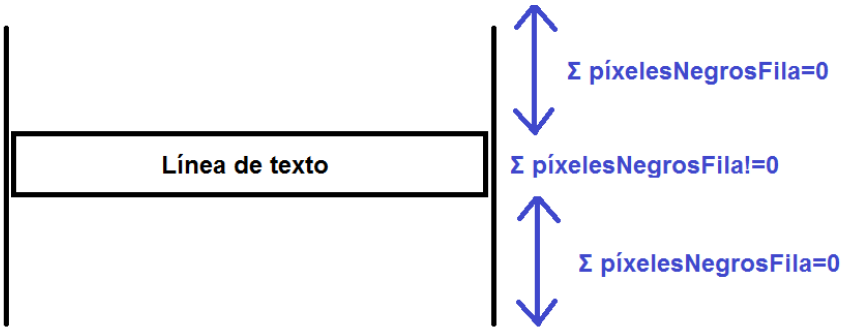


Figura 39: Esquema de detección de líneas de texto tras filtrado interlineal de ruido

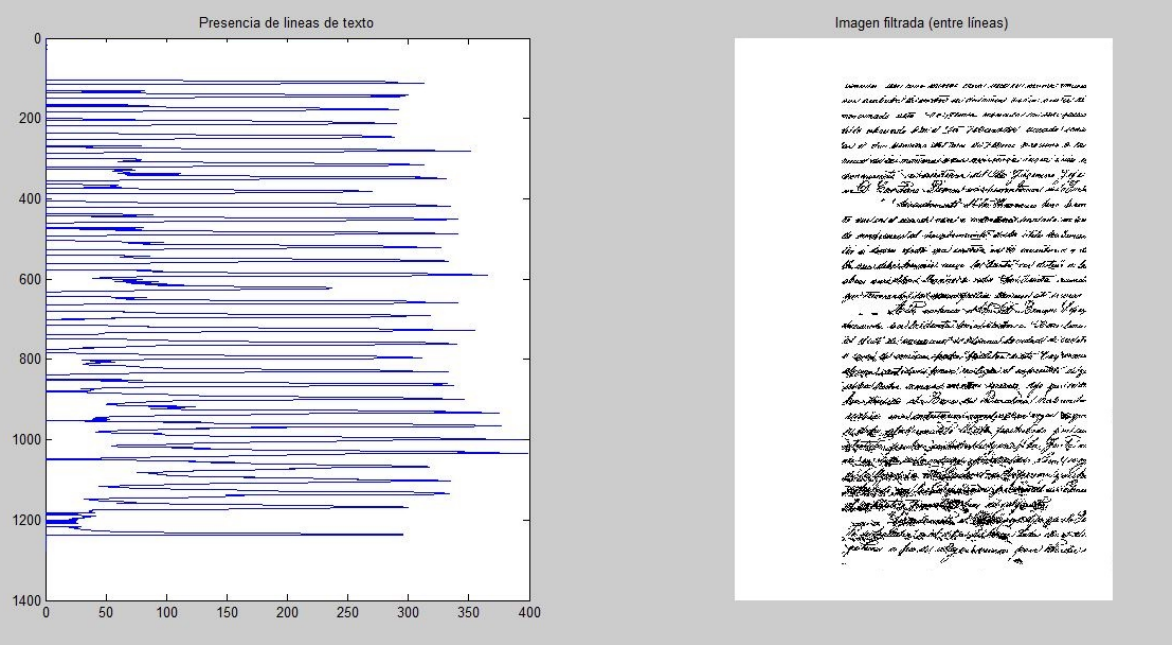


Figura 40: Representación de la acumulación de píxeles negros por fila de la imagen (matriz)

Para recortar cada línea se recorre la imagen interpretada como matriz por MATLAB verticalmente para, atendiendo a la información de la presencia de líneas de texto dada en la función representada a la izquierda de la imagen anterior detectar las coordenadas del inicio y el fin de la misma, es decir, mientras detecta que la línea tiene un valor distinto de cero y hasta llegar a una línea con valor nulo el contenido se va cargando dentro de una nueva imagen.

Por tanto, profundizando en este proceso, el principio y el fin de cada línea de texto se determinan del siguiente modo: en una imagen binaria en MATLAB el color 1 representa el blanco y el 0 el negro; por simplicidad se invierte el valor de cada color y se hace el sumatorio de cada fila de la matriz que representa la imagen; mientras que el sumatorio de la fila sea diferente de cero se van copiando las filas en una nueva imagen hasta que se detecte el final de la línea cuando el sumatorio de una fila vuelva a ser nulo.

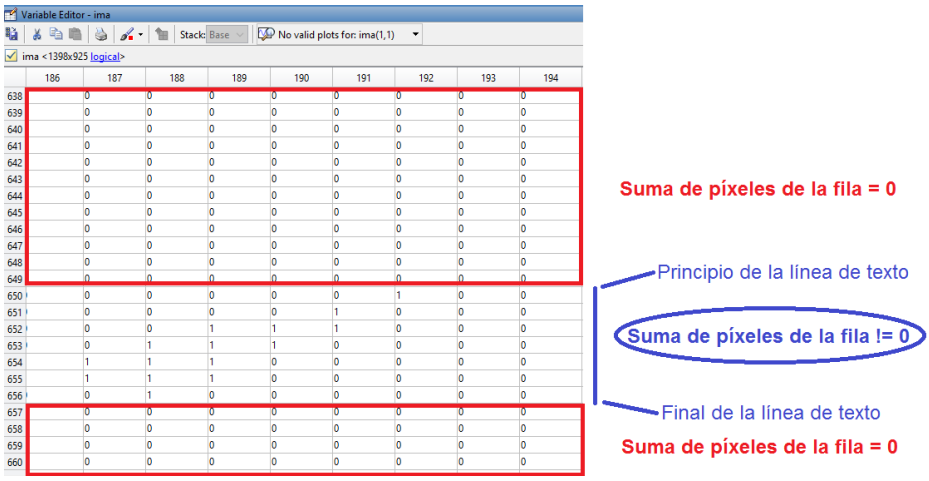


Figura 41: Proceso de detección de líneas de texto sobre la matriz

La línea segmentada se guardará como archivo en el directorio de trabajo.

Figura 42: Línea segmentada

Cada vez que una línea es recortada, dentro del programa principal se pasa localmente a una función llamada "ExtraerPalabras.m" que se encarga de segmentar aplicando el algortimo CCL las palabras presentes a lo largo de dicha línea. A continuación se expone la parte del código donde se recorre la imagen (matriz) ya preprocesada para obtener las líneas de texto que a su vez son procesadas a nivel de palabra individualmente.

```
while i<=filas
                %Si empieza una linea de texto (linea de pixeles distintos de 0).
                if (sum(imagen(i,:)~=0))
                    p=i;

                %Añadimos a la variable linea_texto tantas filas no nulas
                %de la matriz (imagen) como haya antes de la siguiente fila
                %nula (final de la linea de texto).

                        while (sum(imagen(p,:)~=0))
                        p=p+1;
                        r=r+1;
                        linea_texto(r,:)=imagen(p,:);
                        end
                    r=0;
                    imshow(~linea_texto);

                    Extraccion_palabra(linea_texto, i, num_linea); %Pasamos la linea de texto a la función para segmentar las palabras de la misma.
                    imwrite(~linea_texto, ['C:\Users\Luis\Desktop\lineas\linea',num2str(i),'.jpg']); %Guardamos las lineas en el directorio indicado.
                    clearvars linea_texto
                    i=p;
                %En caso de que la fila de la matriz (imagen) sea nula seguimos recorriendo verticalmente la misma hasta encontrar el
                %principio de una linea de texto.
                else
                        i=i+1;
                end

        num_linea=num_linea+1;
end
```

Figura 43: Extracto de código correspondiente a la segmentación de línea

Adicionalmente, el número de líneas es guardado en una variable que posteriormente será escrita en un archivo de texto para la evaluación del proceso

Puesto que el sumatorio de una fila puede ser distinto de cero en el caso de que haya uno o varios píxeles de ruido presentes en la misma, para evitar que se detecten erróneamente líneas de texto se ha fijado un tamaño mínimo de línea de texto, que puede ser modificado en el código del programa en caso de necesitar hacerlo porque se vaya a usar con una obra con diferente tamaño de líneas de texto.

Figura 44: Extracto de código correspondiente a la segmentación de línea

Ejemplo de línea no deseada cuyo tamaño no sobrepasa el umbral para ser considerada línea de texto.

5.2.2.2 Segmentación a nivel de palabra.

Enlazando con el apartado anterior, para llevar a cabo la segmentación a nivel de palabra de cada página del acta procedemos a tratar en la función "ExtraerPalabras.m" la línea de texto segmentada justo anteriormente, esto lo haremos haciendo uso de la función de MATLAB bwlabel(BW) que etiqueta componentes conectados (CCL) en una imagen binaria de dos dimensiones.

La técnica de CCL se aplica a la imagen tras el la etapa de preproceso donde es de gran importancia el proceso de binarización de la misma. El algoritmo se encarga de identificar todos los clusters de figuras espacialmente conectadas, de ahí la relevancia en la correcta aplicación de umbrales en el preproceso de la imagen ya que las regiones negras (píxeles a 1) deben estar bien aisladas del fondo blanco (píxeles a 0), ya que por simplicidad hemos invertido el valor de los píxeles blancos y negros de la matriz.

El uso de esta función permite que al algoritmo de etiquetado de componentes conectadas identificar los componentes conectados que se buscan en este caso, es decir, palabras presentes en cada línea de texto, escaneando secuencialmente la imagen binaria.

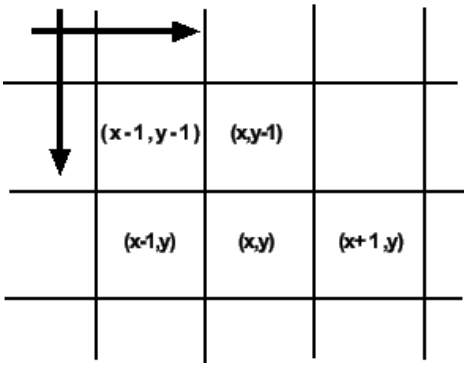


Figura 45: Escaneado de la imagen por el algoritmo de CCL

El algoritmo de etiquetado escanea una imagen binaria dos veces por orden de exploración de línea, por ejemplo, de izquierda a derecha y de arriba debajo de la imagen.

En el primer paso, el algoritmo etiqueta cada píxel con valor 1 con una etiqueta ya existente si está conectado con cualquier otro píxel o con una nueva en caso de ser detectado como el píxel inicial de una figura.

En el segundo paso el algoritmo combina las etiquetas equivalentes, es decir, diferentes etiquetas asignadas a los píxeles conectados y descubre nuevas conexiones posibles. Como resultado del algoritmo, cada región de la imagen perteneciente a un cluster se identifica mediante una etiqueta distintiva.

Este algoritmo de etiquetado de dos pasos podría ser poco eficiente tanto en espacio como en tiempo para imágenes muy grandes, de este modo se ha optado por tratar con este algortimo las imágenes de líneas de palabras a tratar directamente toda la página del acta puesto que tras la experimentación al elaborar el código para segmentar palabras dio un notable mejor resultado el procesamiento por líneas en vez de directamente hacerlo por actas. El pseudocódigo para una iteración del algoritmo de etiquetado es:

Etiquetar un determinado píxel (x,y)

si el píxel (x,y) tiene '0' entonces
    No hacer nada y analizar el siguiente píxel (x+1,y)
siNo el píxel (x-1,y-1) tiene una etiqueta entonces
    Asignar la etiqueta al píxel (x,y)
siNo si ningún píxel (x-1,y) o (x,y-1) no está etiquetado entonces
    Incrementar el número de etiqueta y asignar la última etiqueta al píxel (x,y)
siNo los píxeles (x-1,y) XOR (x,y-1) están etiquetados
    Asignar la etiqueta al píxel (x,y)
siNo si ambos píxeles (x-1,y) y (x,y-1) están etiquetados entonces
    Asignar la etiqueta al píxel (x-1,y) al píxel (x,y)
    Guardar la equivalencia si la etiqueta de los píxeles (x-1,y) y (x,y-1) no son idénticos.

Figura 46: Pseudocódigo del algoritmo de CCL

La función [L Ne]=bwlabel(imagen); devuelve la matriz L de elementos conectados reconocidos por el algortimo, cada palabra reconocida está asociada a una etiqueta (número), mientras que el segundo argumento retornado hace referencia al número total de objetos reconocidos en la imagen.

Un caso sencillo para observar el funcionamiento del proceso con la imagen de texto siguiente, correspondiente a una línea de tres palabras:

Figura 47: Línea de texto de ejemplo

Sobre esta imagen, interpretada por el programa principal como una matriz se aplica el algortimo de etiquetado de componentes conectados explicado anteriormente, obteniéndose:

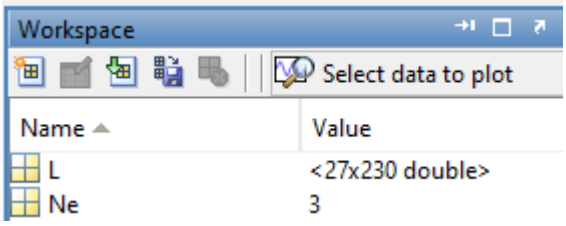


Figura 48: Argumentos devueltos por la función bwlabel

Siendo Ne el número de elementos encontrados y L la matriz de componentes conectados, de dimensiones semejantes a las de la imagen introducida con la información sobre la posición y cluster al que pertenecen los píxeles etiquetados; por ejemplo, una parte de la matriz L donde acaba el primer elemento correspondiente a la palabra “de” y empieza el segundo elemento correspondiente a la palabra “esta” tiene el siguiente aspecto:

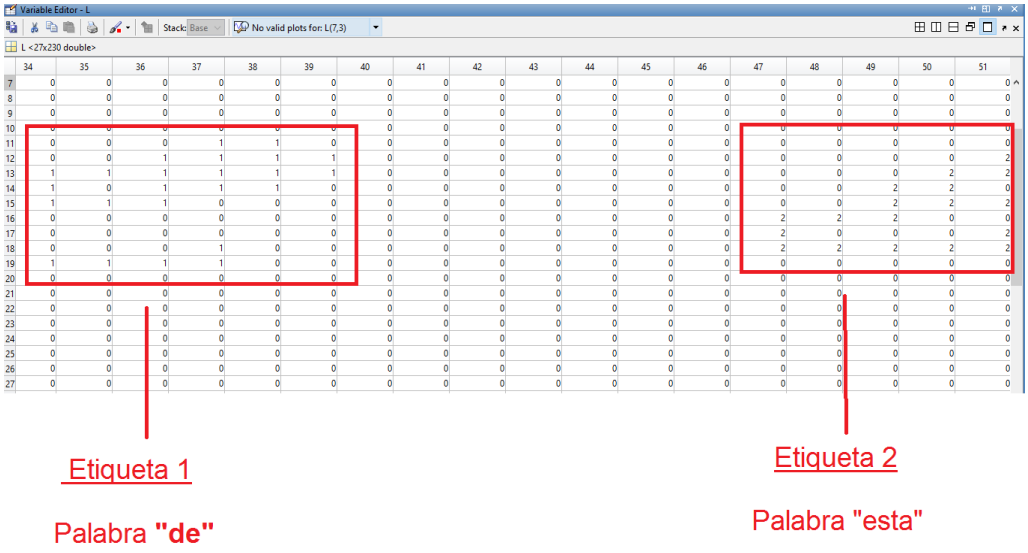


Figura 49: Etiquetado de componentes conectados en la matriz

Posteriormente, haciendo uso de la función propied=regionprops(L,'BoundingBox'); obtenemos una estructura de nombre “propied”. Esta estructura posee tantos elementos como objetos (palabras) encontrados durante la aplicación del algortimo de etiquetado de componentes conectados.

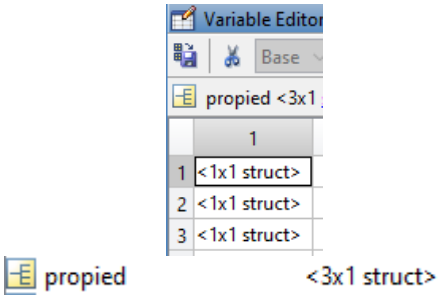


Figura 50: Estructura “propied”

La información contenida dentro de cada elemento de la estructura que se corresponde con una palabra detectada es un vector de cuatro elementos.

Los dos primeros valores del vector indican las coordenadas x e y que sirven de referencia para trazar el rectángulo (Bounding Box) de longitud de base igual a la magnitud indicada por el tercer elemento del vector y altura indicada por el cuarto elemento del mismo.

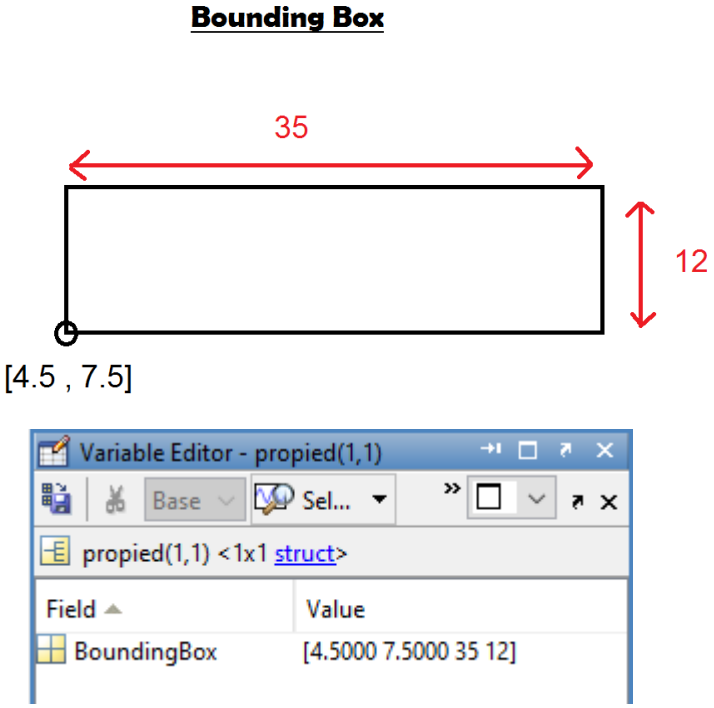


Figura 51: Descripción de Bounding Box

Se puede representar de forma gráfica haciendo uso de la función rectangle de MATLAB a la que se le debe propocionar las coordenadas de cada palabra.

```
%% Plot Bounding Box
for n=1:size(propied2,1)
    rectangle('Position',propied2(n).BoundingBox,'EdgeColor','g','LineWidth',1);
end
```

Figura 52: Extracto del código que representa los Bounding Box

Lo que da un resultado gráfico del desarrollo efectuado.

Figura 53: Línea de texto con los Bounding Boxes representados sobre los componentes conectados

Posteriormente, haciendo uso de la función find de MATLAB podemos guardar cada imagen de palabras detectadas en cada línea (o página) de texto en el caso de que superen los umbrales fijados (x e y).

```
%% Objects extraction

for n=1:Ne
    [r,c] = find(L==n); %find devuelve índices de los elementos no nulos.
    n1=imagen(min(r):max(r),min(c):max(c));
    [x y]=size(n1);
    if x>10 && y >6
        imwrite(~n1, ['C:\Users\Luis\Desktop\Recortadas\prueba',num2str(n),'.jpg']);
    end
end
```

Figura 54 : Extracto del código que extrae de la imagen del acta cada palabra (componentes conectados)

Análogamente a lo hecho al segmentar las líneas de texto se anotará en un archivo de texto la posición de la palabra en la línea para así cuantificar el número de palabras por línea y poder evaluar, tras la ejecución completa del programa, la efectividad del mismo.

Debido a que el estado de las imágenes de las actas no es óptimo, a lo largo del proyecto han surgido errores que hacían que el algortimo de etiquetado de componentes conectados no funcionase de manera óptima en los textos antiguos en los que se ha probado para realizar este TFG; entre ellos está el de el etiquetado erróneo de palabras que estaban una línea más arriba o abajo que se tomaban como parte del componente conectado, algo que se subsanó segmentando primero a nivel de línea para aislar la misma.

Posteriormente otro error común del CCL era la detección de varios componentes que hacían alusión a una misma palabra.

Por ejemplo, en la palabra esta aparecen dos rectángulos, es decir, el algoritmo ha identificado:

Figura 55: Detección errónea de componentes conectados sobre imagen

Para evitar que se solapen los bloques correspondientes a una misma palabra comparamos la coordenada de inicio del rectángulo que rodea la palabra; este motivo fue determinante a la hora de segmentar antes la línea de texto, de este modo únicamente se recorre de izquierda a derecha la palabra comparando el punto de inicio de los diferentes bloques viendo si se solapan y en función de un umbral fijado tras analizar una muestra relevante de casos se decide si se deben fusionar por pertenecer ambos a la misma palabra.

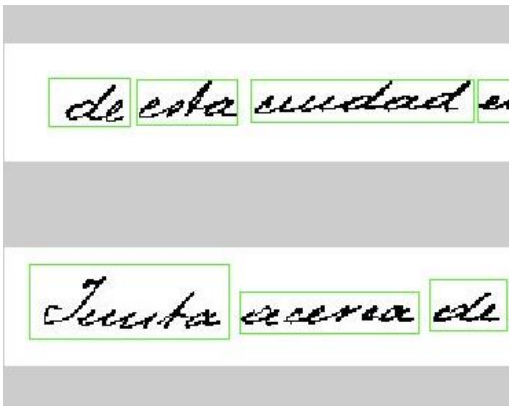


Figura 56: Líneas segmentadas a nivel de palabra

Para segmentar un conjunto de páginas las imágenes sobre el directorio en el que se trabaje usando la función dir de MATLAB obtenemos una estructura con el nombre y número de elementos presentes en el directorio que se va a tratar, posteriormente e un bucle se recorre dicho directorio procesando cada imagen individualmente y anotando las medidas del número de líneas y palabaras efectuadas para posteriormente evaluar la herramienta donde se ha integrado la opción de segmentar por solo por línea o por línea y palabra.

Figura 57: Menú de opciones de segmentación en la aplicación (GUI)

Para proceder a agrupar todas las palabras similares se compararán las imágenes midiendo la similitud entre ellas a través de los píxeles coincidentes en las mismas posiciones.

Para ello sobre las imágenes segmentadas se efectuará una renormalización de las imágenes de dimensiones diferentes para poder efectuar posteriormente la operación AND.

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

AND

Figura 58: Operación AND

De este modo se va a poder guardar en una variable el número de píxeles distintos de cero que quedan tras efectuar esta operación que indicará el número total de píxeles coincidentes entre imágenes; todas aquellas imágenes superen un umbral serán clasificadas como coincidencia y por tanto guardadas en un directorio con el resto de imágenes de palabras coincidentes y borrada del conjunto de imágenes de palabras para clasificar para no crear duplicados. Con este proceso se obtiene un primer paso para la clasificación de imágenes de forma automática.

5.2.1 Integración en la herramienta

Finalmente, se ha creado un menú en la aplicación para poder acceder de forma sencilla a la sección que le interese al usuario y configurar algún parámetro si necesita hacerlo sobre los que están prefijados por defecto, que han sido los que mejor resultado han dado para procesar las Actas de Tarragona.

Figura 59: Menú principal de la aplicación (GUI)

6 Pruebas y resultados

A continuación se detallarán las pruebas realizadas sobre la herramienta implementada.
Se ha efectuado el preprocesamiento y segmentación a nivel de línea y palabra sobre 6800 páginas de las Actas de Tarragona. Se clasificaron automáticamente por calidad y una posteriormente se ha elaborado un archivo de texto paralelamente a la ejecución del programa principal con el identificador de página del acta, número de líneas segmentadas por página, número de palabras segmentadas por página y palabras por línea segmentada.

	A	B	C	D	E	F	G	H	I	J
1	18691105_02.jpg	32	323	13,8,9,10,16,9,13,18,11,10,11,6,5,10,2,0,6,8,16,21,17,12,14,16,7,10,13,15,0,9,8						
2	18691112_01.jpg	32	323	13,8,9,10,16,9,13,18,11,10,11,6,5,10,2,0,6,8,16,21,17,12,14,16,7,10,13,15,0,9,8						
3	18691112_02.jpg	29	262	11,8,0,12,12,9,8,8,10,6,11,12,11,10,14,12,10,0,10,5,0,14,5,0,17,15,20,12,0,9,8						
4	18691115_01.jpg	29	262	11,8,0,12,12,9,8,8,10,6,11,12,11,10,14,12,10,0,10,5,0,14,5,0,17,15,20,12,0,9,8						

Figura 60: Archivo con medidas de resultados del proceso

Posteriormente estas medidas se han comparado con los resultados que se poseían de un proyecto de transcripción manual de los mismos textos manuscritos, particularmente con aquellos textos que presentaban un nivel mayor de complejidad debido a sus condiciones: aquellos del siglo XIX. Se elaboraron unas macros en Visual Basic, para obtener las medidas de los archivos de texto en formato digital del proyecto mencionado anteriormente y poder efectuar una comparación objetiva con todas las páginas de la obra calibrando la herramienta con 10 muestras aleatorias.

Tras efectuar la comparación sobre 4063 páginas de las Actas de Tarragona se ha obtenido que el número de líneas por página en la transcripción manual en promedio es de 22; tras proceder a segmentar las líneas de los documentos, en promedio se obtuvieron 17 líneas.

Respecto a las palabras por documento, las medidas obtenidas del proyecto de transcripción manual indican 101 palabras presentes por documento de media, mientras que con los resultados de la segmentación hemos obtenido 140 en promedio.

Finalmente, de forma cualitativa se evaluó el agrupamiento de palabras sobre una muestra de palabras segmentadas siendo de un 70% el acierto en la determinación de la similitud de imágenes de palabras elegidas aleatoriamente.

7 Conclusiones y trabajo futuro

7.1 Conclusiones

Tras haber integrado los módulos principales para la transcripción automática de textos manuscritos se ha determinado que, pese a la complejidad de la problemática, la automatización del proceso de transcripción es un gran avance en este campo de la investigación con respecto a los métodos tradicionales que se han llevado a cabo.

Los proyectos de transcripción de este tipo de documentos son por lo general a mano; particularmente en un proyecto de transcripción manual de los documentos históricos en el que trabajaron dieciséis personas se tardó tres meses y medio en interpretar y posteriormente transcribir a un formato digital la información presente en los textos, es decir, un proceso bastante más lento de lo que se puede conseguir poniendo al servicio de la transcripción la potencia de cómputo actual. Para tratar (preprocesar, segmentar a nivel de línea y palabra, y obtener la medida de distancia para posteriormente agrupar) las cerca de siete mil actas de la base de datos de la que se partió (la misma que la del proyecto de transcripción manual) se tardó dos horas, lo que permite acortar los plazos en el desarrollo de un proyecto de este tipo de forma notable.

La etapa de calibración de la herramienta sobre muestras de la obra completa debe efectuarse con un número lo más alto posible para tener en cuenta las diferentes singularidades que presenta cada página y que estas no afecten al proceso de segmentación. El preproceso de la imagen es la etapa más importante para obtener buenos resultados de segmentación por lo que es recomendable hacer numerosas pruebas para segmentar grandes volúmenes de documentos.

En definitiva, la eficiencia del proceso de transcripción se aumenta con la creación de la herramienta ya que un proceso que de forma manual tarda meses con la presencia de dieciséis personas, poniendo al servicio el procesamiento informático de dichos documentos tarda horas sin la necesidad de tener tantas personas en el equipo de transcripción; una única persona es suficiente. Esta digitalización del proceso permite efectuar en paralelo varias transcripciones, con diferentes opciones de calibración para ver cuál es la que arroja un mejor resultado por lo que en definitiva esta herramienta supone el ahorro de tiempo y recursos respecto a cómo se ha venido haciendo la transcripción de textos en los últimos años.

7.2 Trabajo futuro

Con los resultados obtenidos a lo largo del trabajo se puede concluir por dónde sería necesario seguir para mejorar globalmente el proceso, para ello, en la parte inicial del tratamiento de las imágenes de texto, es decir, en el preproceso de las mismas se podría efectuar un estudio automático por zonas de la página para poder obtener unas análisis estadístico automático de toda la obra que indique las zonas con más presencia de ruido en las imágenes de páginas que la compone para así no tener que parametrizar manualmente todas las opciones de la herramienta. Análogamente, en el clasificador por calidad de imagen se utilizaría este análisis para hacer más precisa esta clasificación.

En lo relativo a la segmentación se podría probar otras técnicas diferentes al etiquetado de componentes conectados e ir profundizando en mejorar la calidad de la segmentación según la época del texto, integrando para cada una de ellas las singularidades de la misma extendidas entre los escribanos.

Finalmente, en la parte de agrupamiento de palabras se podría probar más técnicas introduciendo programación dinámica y elaborando nuevos algoritmos que agrupen correctamente por similitud las imágenes correspondientes a una misma palabra además de, utilizando las medidas de distancia obtenidas, proceder a ejecutar algoritmos de clustering no supervisado sobre las mismas.

De forma complementaria se podría elaborar una nueva presentación del software en otro lenguaje de programación que permita una mejora del aspecto de la herramienta final.

Referencias

[1] G. Louloudis, B.Gatos, I.Pratikakis, C.Halatsis, “Text line and word segmentation of handwritten documents”

[2] J.Vives,. “Digitalización del patrimonio: archivos, bibliotecas y museos en la red” Barcelona, 2009.

[A] El-Yacoubi, R. Sabourin, M. Gilloux, C.Y. Suen “Off-Line Handrittwn word recognition using Markov Models”

[3] V. Romero, N. Serrano, A. Toselli, J. Andreu , E. Vidal, “Handwritten Text Recognition for Historical Documents”

[4] I. Bar-Yosef, N. Hagbi, K. Kedem “Line segmentation for degraded handwritten historical documents”

[5] CS169 - Software Engineering OCRchie,
“http://people.eecs.berkeley.edu/~fateman/kathey/skew.html”

[6] T. Rath, R. Manmatha “Word Image Matching Using Dynamic Time Warping”

[7] S.Saha, S. Basu, M. Nasipuri, D. Kr. Basu “A Hough Transform based Technique for Text Segmentation”

[8] I. Bar-Yosef, N. Hagbi, K. Kedem “Line segmentation for degraded handwritten historical documents”