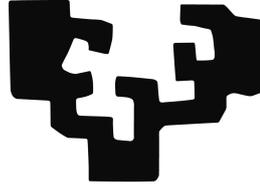


eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Facultad de Informática / Informatika Fakultatea

TITULACIÓN: Ingeniería Informática

Identificación y clasificación de perfiles
en un entorno controlado.

Alumno/a: D./Dña. Gabriel Baquedano Santarén

Directores/as: D./Dña. Abdelmalik Moujahid Moujahid,

D./Dña. José María Rodríguez

Proyecto Fin de Carrera, Febrero de 2014

Resumen

La evolución en los sistemas de información y campos como la visión por computador ha tenido un impacto notable en los procesos de producción de las empresas durante los últimos años.

El presente trabajo trata la aplicación de técnicas de visión por computador y desarrollo de software con el objetivo de simplificar, mejorar y agilizar una parte del proceso de producción de una empresa de extrusión de materiales de caucho y derivados.

La empresa en para la que se ha desarrollado éste trabajo se llama Standard Profil¹ y está localizada en el polígono industrial de la Portalada, en Logroño. Su principal actividad consiste en la producción y manipulado de piezas de caucho utilizadas en el sector de la automoción.

Las capas de presentación finales de éste proyecto están enfocadas a un perfil de operario que no tiene porqué tener conocimientos avanzados de informática.

¹Página web oficial (inglés/turco):<http://www.standardprofil.com>

Agradecimientos

En primer lugar, me gustaría agradecer la dedicación de las dos personas que han supervisado la realización de éste proyecto desde el principio hasta el final: a José María Rodríguez por depositar su plena confianza en mí, y a Abdelmalik Moujahid por guiarme en cada momento que lo he necesitado.

A mis padres, por su continuo apoyo en los momentos más difíciles que éste periodo de tiempo ciertamente ha tenido.

Por último, a todas éstas personas, estén lo lejos que estén, que se han interesado por el progreso del proyecto y que han conseguido animarme en cada momento.

Índice general

Resumen	I
Índice general	V
Índice de figuras	IX
Indice de tablas	XI
1. Introducción	1
2. Documento Objetivos del proyecto	3
2.1. Motivación	3
2.2. Objetivos	3
2.3. Metodología	4
2.4. Gestión del proyecto	5
2.5. Estructura de Descomposición de Tareas (EDT)	5
2.6. Riesgos y planes de contingencia	8
2.7. Métodos de trabajo	9
2.8. Resto de la memoria	10
	v

3. Introducción	13
3.1. El problema	13
3.2. Estado del arte	14
3.3. Herramientas	15
3.3.1. OpenCV	15
3.3.2. Python	15
3.3.3. C++/CLI	15
3.3.4. HTML5 y Javascript	16
4. Segmentación	17
4.1. Descripción y objetivos	17
4.2. Primer procesamiento	17
4.3. Eliminando ruido	19
5. Descriptores de formas	21
5.1. Conceptos generales	21
5.2. Descriptores de Fourier	21
5.3. Firmas unidimensionales	22
6. Formación de una firma con el contorno	25
6.1. Simplificación del contorno	26
6.2. Aplicación de la transformada de Fourier	27
6.3. Invarianzas	28
6.4. Diseño	30
6.5. Implementación	31
7. Comparación con el plano	35
7.1. Diseño	36
7.2. Herramientas	37
7.3. Funcionalidades	38

8. Conexión entre ambas herramientas	41
8.1. Pruebas	47
Anexos	51
A. Desviaciones en la planificación	53
B. Almacenamiento de datos en la intranet	57
C. Normalización de la iluminación	59
D. Conclusiones	61
D.1. Futuros puntos de trabajo	61
Bibliografía	63

Índice de figuras

2.1. Diagrama <i>Gantt</i>	6
2.2. Diagrama <i>EDT</i>	7
4.1. Sección de las <i>hileras</i> H421 y H468.	18
4.2. Plano de las secciones H421 y H468.	18
4.3. Operación de <i>threshold</i> gráficamente.	19
4.4. <i>Hileras</i> H421 y H468 binarizadas tras el <i>threshold</i>	19
4.5. <i>Hileras</i> H421 y H468 suavizadas y binarizadas tras el <i>threshold</i>	20
6.1. Contorno de la <i>hilera</i> H421 con 4, 8, 16, 32, 64 y 128 puntos	26
6.2. Valor de los descriptores de la <i>hilera</i> H421 con 128 puntos de contorno.	27
6.3. Valor de los descriptores del plano H421 con 128 puntos de contorno.	28
6.4. Valor de los descriptores la <i>hilera</i> H421 y su plano comparados.	30
6.5. Vista de la interfaz de usuario	33
8.1. Contornos con misma magnitud en todos los descriptores	42
8.2. Comparación de los descriptores de la <i>hilera</i> 561 con su plano conservando la información de fase.	44
8.3. Comparación de los descriptores de la <i>hilera</i> 973 con su plano conservando la información de fase.	44
8.4. Desplazamiento en el <i>starting point</i> provocado por la rotación entre la imagen 561 y su plano.	45

8.5. Desplazamiento en el starting point provocado por la rotación entre la imagen 973 y su plano.	45
8.6. Comparación de los descriptores de la <i>hilera</i> 561 con su plano conservando la información de fase y habiéndolos rotado.	46
8.7. Comparación de los descriptores de la <i>hilera</i> 973 con su plano conservando la información de fase y habiéndolos rotado.	46
8.8. Rotación de la <i>hilera</i> 561 con respecto a su plano. La estimación del algoritmo es de -76.51°	47
8.9. Rotación de la <i>hilera</i> 973 con respecto a su plano. La estimación del algoritmo es de -176.09°	47
8.10. Comparación entre los planos 973 y 994, cuyos descriptores han identificado de forma incorrecta.	49
8.11. Visualizador de plano abierto tras identificar correctamente la <i>hilera</i> 463 .	50
A.1. Diagrama <i>Gantt</i> final.	54
A.2. Diagrama <i>EDT</i> con las tareas formativas completadas.	55
B.1. Portal de digitalización de hileras	57
B.2. Vista de calendario con las hileras digitalizadas durante los días marcados	58

Indice de tablas

8.1. Desplazamientos y ángulos de las secciones con respecto a sus planos. . .	43
8.2. Clasificación de las piezas según los distintos algoritmos	48

1. CAPÍTULO

Introducción

Situándonos en el contexto de casi cualquier empresa que cuente con un conjunto de procesos de producción, podemos entender éste proyecto como un paso más en la mejora y evolución de una parte de sus procesos de producción.

En la actualidad, el trabajo correspondiente al control de calidad así como el almacenaje del histórico de piezas fabricadas se hace mediante cortes de secciones físicas archivadas en carpetas. Dada la cantidad de producto que se debe almacenar y la frecuencia con la que se efectúan cortes de control, el espacio requerido para almacenar dicha información no es ni mucho menos, despreciable. Para pasar de cortes físicos de piezas a información digital hemos utilizado imágenes tomadas por un escáner. Las imágenes se han sometido a un procesamiento y almacenaje con el objetivo de sustituir y mejorar notablemente el espacio y velocidad del sistema actual.

Nuestro objetivo ha sido facilitar una herramienta que nos permita la identificación de la sección con la que estamos trabajado y compararla con el plano referencia que refleja su producción óptima.

Del procesamiento de la imagen hemos extraído la información que nos ha permitido llevar a cabo la identificación agilizando así la interacción persona-sistema.

Una vez identificada la sección, hemos preparado un entorno web que nos ha permitido comparar la sección obtenida con el plano, pudiendo detectar posibles errores en el proceso de producción, así como identificar los parámetros a modificar para corregir dichos errores en futuras fabricaciones.

Siempre siguiendo el enfoque corporativo en el que nos encontramos, nuestro objetivo ha sido también adaptarnos a los sistemas actuales y evitar todo tipo de cambios mayores que puedan provocar la pérdida de productividad. Éste pensamiento ha sido determinante en las herramientas utilizadas para llevar a cabo el trabajo.

2. CAPÍTULO

Documento Objetivos del proyecto

2.1. Motivación

El presente proyecto forma parte de la modificación del proceso de fabricación de materiales de caucho extruídos. Las herramientas utilizadas hasta ahora tanto para almacenar el historial de piezas fabricadas como para compararlas con sus respectivos planos carecen de un sistema inteligente capaz de agilizar este proceso.

En este sentido, el presente proyecto presenta una solución inteligente a la fase final del proceso de fabricación correspondiente a las tareas de identificación, almacenamiento y consulta de piezas fabricadas.

2.2. Objetivos

El objetivo principal será la identificación y almacenamiento de imágenes de perfiles de caucho procesadas mediante un escáner, así como facilitar una herramienta de comparación del perfil obtenido con el plano-ideal.

Siguiendo el enfoque corporativo en el que nos encontramos, la aplicación informática deberá disponer de una interfaz sencilla e intuitiva capaz de preprocesar una imagen de una pieza determinada, extraer de la misma una firma que la identifique de manera inequívoca y reconocer el modelo de la misma.

El proceso de identificación de las piezas consta de dos fases. Primero, las imágenes pasan por un proceso de segmentación en el cual se identifican las diferentes regiones que las componen; luego, se procede a la extracción de los vectores de características o firmas digitales.

El algoritmo deberá a su vez, ser invariante a los cambios de escala pieza-plano (podríamos trabajar en diferentes escalas), a las rotaciones (no tenemos por qué saber cómo está dibujado el plano o forzar una posición de escaneo) y a la traslación. Adicionalmente, deberá identificar correctamente aquellos perfiles que no sean exactamente iguales al plano, ya que existen ciertas tolerancias y errores dentro de los márgenes de fabricación que están permitidos.

Por otra parte, se facilitará un visor de plano que superpondrá la imagen del perfil con el plano-ideal y además tendrá unas herramientas que ayudarán con la toma de cotas y distancias con el objeto de determinar posibles errores o desfases en el proceso de fabricación.

2.3. Metodología

El progreso del proyecto irá determinado por las conclusiones que se extraigan de las reuniones que se llevarán a cabo junto con el responsable del proyecto de la empresa. El proyecto queda dividido en dos partes, la primera abarca el proceso de identificación de la sección fabricada. Constará de una aplicación que dada una imagen nos facilite el modelo de pieza que aparece en la misma. En un futuro dicha aplicación deberá ser capaz de integrarse a modo de componente para poder formar parte de un proyecto más amplio. La segunda parte del proyecto es el siguiente paso a la identificación: una vez determinada la pieza, abrir un plano interactivo que nos permita comparar la pieza identificada con la imagen que hemos procesado. Además de la comparación visual se pide un pequeño conjunto de herramientas que nos permitan manejar la imagen dinámicamente y acotar partes de la sección con la que estamos trabajando.

El desglose general del proyecto tiene varias partes:

- **Procesos Tácticos:** Abarca la parte administrativa y determina los pasos a llevar a cabo para poder completar el proyecto, así como determinar si en cada momento se están cumpliendo los objetivos.

- **Procesos Formativos:** Son el conjunto de tareas formativas que se llevarán a cabo para poder utilizar las herramientas necesarias para efectuar con éxito las tareas del proyecto, además de la propia integración en el ambiente de trabajo empresarial. Ésto es, aprender su propia metodología de trabajo y adaptarse a ella.
- **Procesos Operativos:** Son las diferentes partes que forman los entregables del proyecto. Cada parte se desarrollará en un prototipo independiente que determinará tras una reunión si cumple los objetivos esperados o si requiere alguna modificación.

2.4. Gestión del proyecto

Si bien en cada uno de los apartados llevados a cabo para la realización del proyecto comentamos posibles futuros puntos de trabajo y mejoras pendientes, se han dejado en un ámbito secundario para evitar el vencimiento de los plazos establecidos inicialmente. El conjunto de reuniones nos permitirá a lo largo del proyecto no sólo conocer en qué situación se encuentra el proyecto en cada momento, sino proyectar rutas de trabajo para un futuro que nos permitan mejorar y perfeccionar el proyecto base.

2.5. Estructura de Descomposición de Tareas (EDT)

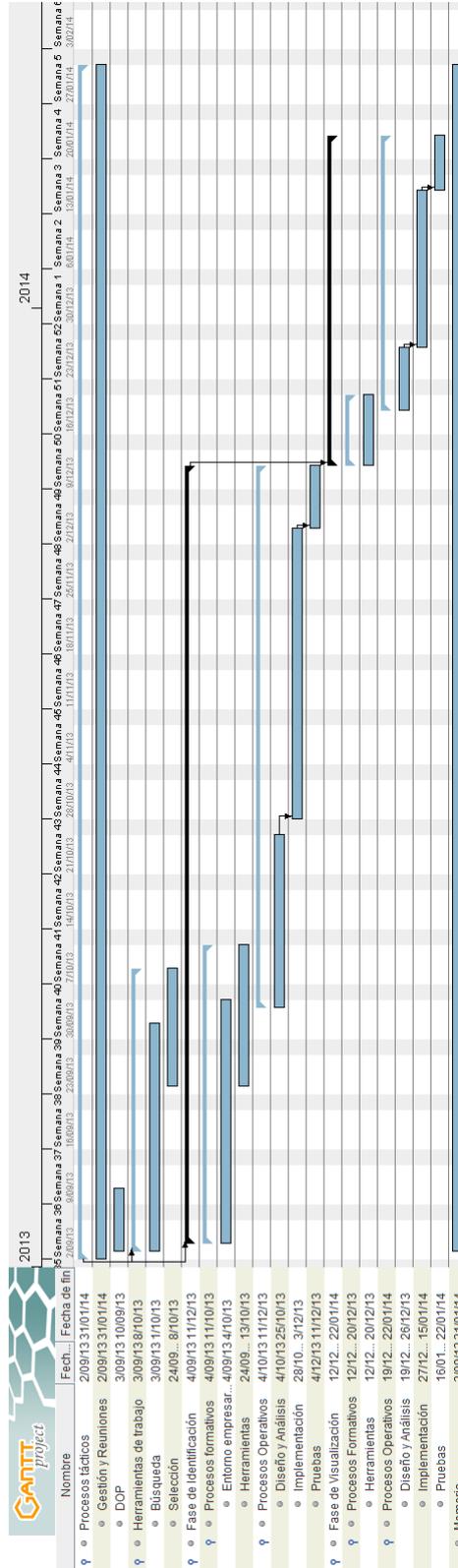


Figura 2.1: Diagrama Gantt.

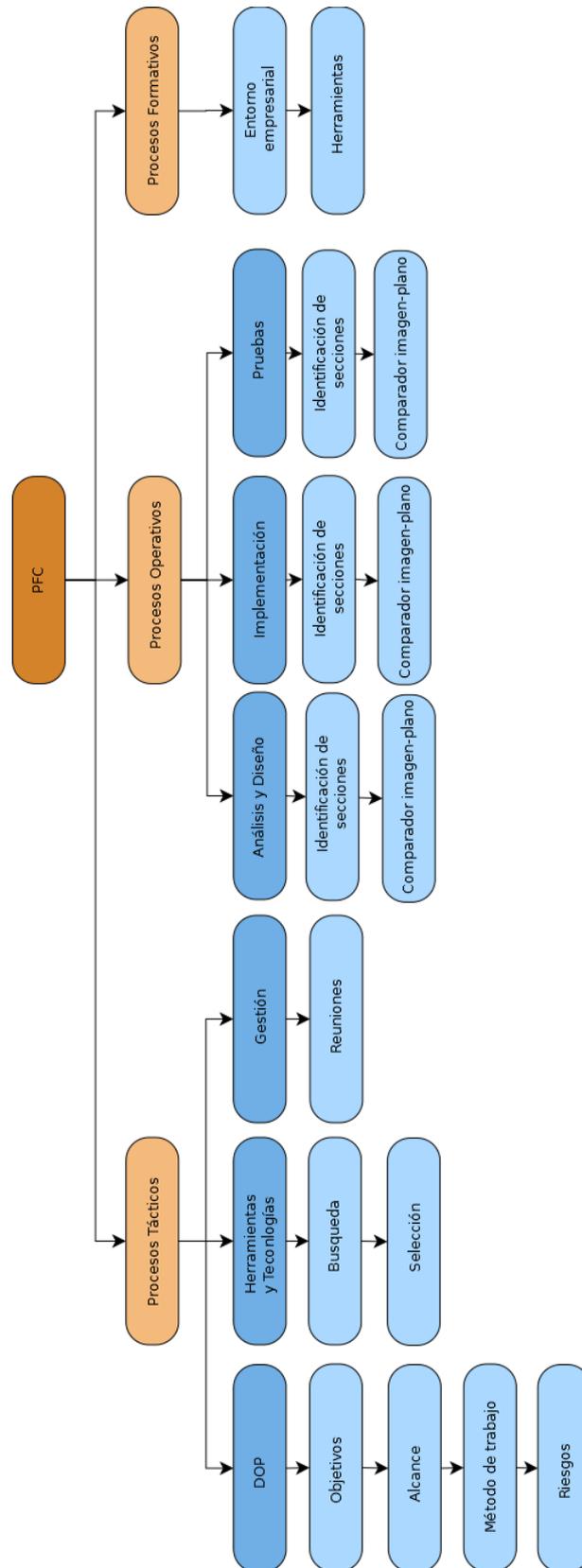


Figura 2.2: Diagrama EDT.

2.6. Riesgos y planes de contingencia

- Equipo estropeado
 - Descripción: un fallo en el hardware podría dejar inutilizado el equipo en el que se está desarrollando el proyecto, y, dependiendo del componente estropeado, podría conllevar una pérdida del trabajo ya realizado.
 - Probabilidad: baja
 - Plan de contingencia: realizaremos copias de seguridad al finalizar cada sesión de trabajo en Dropbox y Google Drive que nos permitan continuar trabajando o minimizar la pérdida a las últimas horas trabajadas.
- Fallo del servicio Google Drive
 - Descripción: El servicio de almacenamiento de ficheros de Google deja de funcionar.
 - Probabilidad: baja
 - Plan de contingencia: tenemos otra copia del trabajo en la nube mediante Dropbox.
- Fallo del servicio Dropbox
 - Descripción: El servicio de ficheros en la nube Dropbox, deja de funcionar.
 - Probabilidad: baja
 - Plan de contingencia: al igual que en el caso anterior, disponemos de la copia de seguridad en éste caso en Google Drive.
- Enfermedad
 - Descripción: imposibilidad de avanzar en el proyecto durante unos días por enfermedad.
 - Probabilidad: media
 - Plan de contingencia: habrá que modificar la planificación inicial e informar al director del proyecto para adaptarse al cambio, modificando si fuera necesario la fecha de entrega de esa parte del trabajo.
- Cambios en los requisitos

- Descripción: si bien los grandes cambios no están contemplados, los pequeños son comunes en un proyecto de éste tipo. Podrían ser desde especificaciones para la futura integración hasta modificaciones en la interfaz o aspectos internos de diseño.
 - Probabilidad: alta
 - Plan de contingencia: adaptar los cambios a la planificación inicial y actualizar la estimación de horas necesarias.
- Indisponibilidad para reunirse
 - Descripción: el director del proyecto o el alumno no pueden acudir a una reunión que estaba establecida.
 - Probabilidad: media
 - Plan de contingencia: comunicarse vía email y modificar la fecha de la reunión, además de comentar aspectos importantes en el propio correo que de no ser tratados impedirían el desarrollo de las siguientes tareas.

2.7. Métodos de trabajo

El proyecto quedará dividido en dos partes principales, la de identificación y la de visualización. Ambas partes tendrán en común los procesos tácticos y parte de los formativos (tendrán herramientas diferentes), pero sus procesos operativos tendrán una dependencia final-inicial.

- Procesos tácticos
 - Gestión del proyecto: se llevará a cabo realizando un seguimiento del diagrama Gantt, así como en conjunto con reuniones con el director del proyecto.
 - Reuniones con el director del proyecto: una reunión cada dos semanas para detallar el estado del proyecto en cada momento y poner en común las decisiones a tomar.
 - Reuniones con el tutor del proyecto: se llevará a cabo una reunión mensual para explicar la situación del proyecto y comentar aspectos relativos al mismo, así como pedir opinión en la toma de decisiones.
 - Memoria: la redacción se llevará a cabo en Texmaker utilizando una plantilla.

- Procesos formativos
 - Se llevará a cabo una búsqueda de las herramientas que pueden ser utilizadas para llevar a cabo el proyecto. La búsqueda se realizará a través de internet y se preguntará a personas conocidas sobre su experiencia en proyectos similares o relacionados.
 - De los datos recopilados, se procederá a un proceso formativo estimado de 2 semanas, en las cuales nos familiarizaremos con los conceptos generales de éstas herramientas y su utilización básica para poder comenzar a trabajar.

- Procesos operativos
 - Se buscará información relativa al estado del arte actual y se consultarán diferentes métodos y formas para identificar figuras.
 - Se procederá a realizar un análisis en profundidad de la información que se nos facilita para poder abarcar el problema.
 - Se procederá a buscar librerías para web que nos permitan mostrar y manipular gráficos vectoriales.
 - Se buscará la forma de conectar ambos proyectos y fomentará la cooperación entre ambos para mejorar la integración.

2.8. Resto de la memoria

En el capítulo 3 del presente documento se realiza una introducción que del estado del arte de los temas tratados en el proyecto. El capítulo 4 presenta la primera problemática a la que nos enfrentamos, la segmentación de la pieza del fondo y la reducción de ruido.

Posteriormente comentaremos en rasgos generales las características de las firmas de formas y en especial los descriptores de Fourier, materia clave en la resolución del problema principal. Explicaremos algunas de las firmas que se utilizan con los descriptores y más adelante daremos algunas recetas que las dotarán de características deseables, junto con la aplicación de las mismas al contorno de la imagen ya segmentada, en el capítulo 6.

En el capítulo 7 mostraremos el diseño que hemos seleccionado para el visor de planos y comentaremos algunas características así como sus funcionalidades más importantes para emular el trabajo que los operarios hacen a día de hoy a mano.

En el siguiente capítulo mostraremos el enlace entre ambas partes del proyecto, y explicaremos una importante mejora realizada en el análisis de Fourier que nos facilitará la colocación de la pieza en nuestro plano.

Por último, plasmaremos las conclusiones que hemos ido adquiriendo durante la realización del presente proyecto y sugeriremos las líneas de mejora que nos han aparecido y que consideramos más importantes.

3. CAPÍTULO

Introducción

3.1. El problema

El conjunto de características deseables que debe tener nuestro algoritmo se deducen de las primeras pruebas realizadas con el sistema de digitalización propuesto (escáner) sobre un conjunto inicial de piezas:

- **Adaptación:** nuestro algoritmo debe ser adaptable. Las imágenes tendrán diferentes brillos y contrastes (nos limitemos a escáneres planos pero no todos tienen el mismo sistema de digitalización y lentes). Una pieza procesada en un mismo escáner en momentos distintos no tiene por qué provocar la misma salida. En caso de un escáner distinto, la salida será presumiblemente más dispar. Más adelante trataremos de paliar los éstos efectos con un sistema de iluminación LED.
- **Invarianza:** la firma que extraigamos de una pieza debe ser invariante a la traslación, escalado y rotación. El motivo es eliminar restricciones en el proceso de producción. No nos debe importar la orientación que tenga la imagen ni la posición en la que se encuentre la pieza dentro de la misma, y el tamaño tampoco ya que es algo que podemos cambiar con la configuración del escaneado. Dos piezas que sean de distinto tamaño u orientación pero con similares propiedades geométricas representarán para nosotros la misma clase.
- **Tolerancia** a pequeñas desviaciones: dentro del proceso de producción en el que nos encontramos, existen márgenes de error "permitidos", y la propia elasticidad de

algunos componentes de las piezas puede hacer que en el momento de la digitalización de la imagen se produzca alguna pequeña deformación. La propia naturaleza de algunos componentes finales a lo largo del proceso hace que existan deformaciones controladas que no son corregidas hasta encajar las piezas en el producto final. Nuestro algoritmo debe tener en cuenta dichas desviaciones y no alterar el resultado producido.

3.2. Estado del arte

La rápida evolución en los sistemas de visión artificial y la bajada de precios en los dispositivos de digitalización como cámaras o escáneres ha provocado un acercamiento de éstos sistemas a los entornos industriales. Si bien la visión por computador es considerada un subcampo de la inteligencia artificial, resulta difícil entenderla como un campo independiente, ya que la gran mayoría de las veces los datos obtenidos tras el procesado de imágenes terminan desembocando en un proceso de aprendizaje. En otras ocasiones, el algoritmo que procesa los datos de la imagen ha recibido un entrenamiento previo, como en los algoritmos de reconocimiento facial.

En el campo de visión artificial han emergido empresas en los últimos años que se han centrado en la especialización. Si bien antes los fabricantes de cámaras y los diseñadores de sistemas de visión tenían una relación de fabricante-proveedor, hoy en día no es difícil encontrar empresas que sean fabricantes-proveedoras y a su vez distribuidoras de sistemas completos de visión, compuestos por hardware y software, formando un producto atractivo (y generalmente muy asesorado) para empresas que quieran introducir éstos nuevos sistemas en sus procesos de producción.

En la parte software, existen multitud de librerías y alternativas que nos permiten trabajar con cualquier tipo de imagen. Asimismo dichas librerías tienen diferentes licencias y sistemas soportados: desde licencias gratuitas (a su vez más o menos restrictivas) a licencias privativas (cuyas condiciones dependen del fabricante y suelen ir asociadas a un componente hardware en particular, aunque no tiene porqué ser así) con soporte para plataformas móviles, equipos de escritorio o hardware más específico. Es decir, existen muchas alternativas para implementar introducir la visión artificial en nuestros sistemas y tenemos un amplio abanico de posibilidades en el que elegir lo que más se ajuste a nuestras necesidades.

En nuestro caso, las referencias a la librería OpenCV en multitud de documentos

que trataban diferentes temas de visión artificial junto con su licencia BSD y el soporte multiplataforma, la convirtió en un primer punto de partida como librería de desarrollo para el presente proyecto.

3.3. Herramientas

3.3.1. OpenCV

De la página web de OpenCV: OpenCV [4] es una librería de código abierto con licencia BSD que incluye varios cientos de funciones de visión por computador. La utilización de OpenCV se ha extendido a multitud de aplicaciones que van desde sistemas control de movimiento hasta el control de procesos con reconocimiento de objetos.

OpenCV dispone además de bindings para Python y Java, que facilitan las llamadas a sus funciones desde dichos entornos.

En el presente proyecto se ha utilizado la librería en dos partes: una con el binding de python para facilitar el diseño y las primeras pruebas y otra en la versión final del sistema de identificación de perfiles, en su versión compilada en C++.

3.3.2. Python

Python [5] ha ido ganando terreno como lenguaje de programación de alto nivel. Utilizando el binding de OpenCV y la cantidad de librerías ya disponibles (Numpy y matplotlib entre otras) tenemos un conjunto de herramientas que nos permiten llevar a cabo pruebas de manera muy rápida y que nos ayudarán a abordar el diseño, permitiéndonos tener una primera impresión de los resultados que vamos obteniendo.

3.3.3. C++/CLI

C++/CLI es una especificación de lenguaje que extiende C/C++. Fue creada por Microsoft y nos permite trabajar tanto con fragmentos de código administrado como no administrado. El intérprete está disponible desde hace años en la versiones Express de Visual Studio. Utilizando C++/CLI tendremos acceso a los componentes de .Net Framework (4.5.1) y podremos diseñar una interfaz actualizada de forma más sencilla ayudándonos

del editor. Gran parte de la elección de utilizar éste lenguaje y las herramientas de Microsoft ha venido determinada por la integración que se realizará en el proyecto en un futuro, pasando a formar parte (como componente) de otros proyectos.

3.3.4. HTML5 y Javascript

La segunda parte del proyecto consiste en crear una herramienta web que nos permita comparar la imagen de la sección obtenida con el plano. Los planos que se nos han facilitado están en formato SVG (Scalable Vector Graphics), formato que puede ser visionado en prácticamente cualquier navegador actualizado. Combinaremos entonces, diferentes librerías Javascript y el canvas HTML5 para crear una interacción imagen-plano que nos permita realizar comparaciones y tomar cotas.

4. CAPÍTULO

Segmentación

La segmentación es el proceso por el cual aislamos, agrupamos o determinamos conjuntos de píxeles de una imagen que son de interés para su análisis o tratamiento. En nuestro caso, un primer análisis visual de imágenes de las secciones que tenemos que identificar nos da las primeras pistas sobre la información que nos puede resultar útil:

- Todas las piezas tienen un color bajo en la escala de grises. A pesar de que en algunas imágenes podemos distinguir diferentes componentes por su color, no existe un margen muy amplio que permita diferenciarlos fácilmente.
- Los propios planos que se nos han facilitado no contienen información del color. No obstante esto no significa que no podamos aprovechar la información del color: podemos guardarla para en un futuro realizar un análisis más exhaustivo.
- La sección se se puede segmentar fácilmente del fondo utilizando un algoritmo threshold que trabaje a nivel de grises.

4.1. Descripción y objetivos

4.2. Primer procesamiento

Realizando un análisis de las imágenes de las secciones [4.1](#) y de los planos que las representan [4.2](#), vemos que, a priori, la única información que podemos comparar es el

contorno exterior de la sección y las características geométricas del mismo.

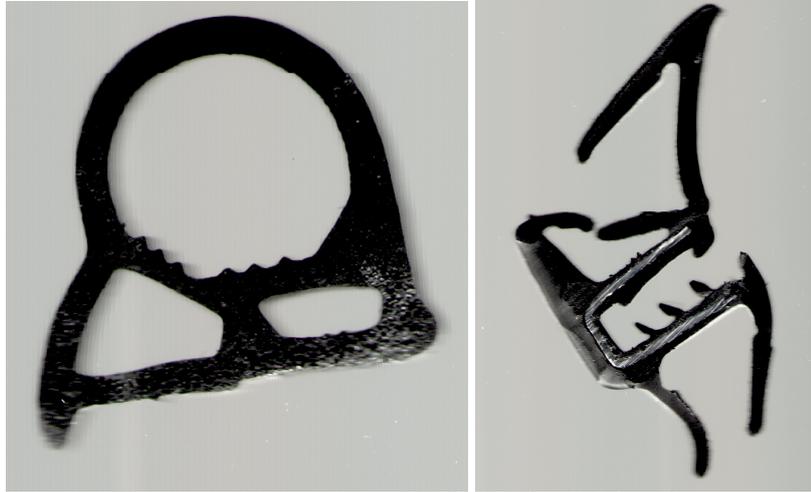


Figura 4.1: Sección de las *hileras* H421 y H468.

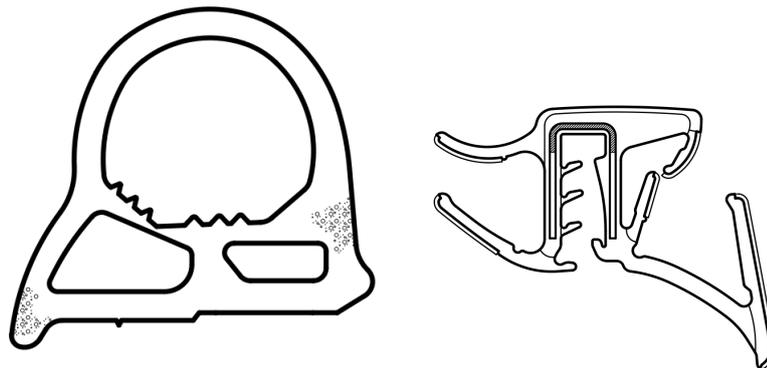


Figura 4.2: Plano de las secciones H421 y H468.

Dada la composición de las secciones de caucho facilitadas y el entorno en el que las analizamos, nuestra primera aproximación para conseguir el contorno se basará en una función `threshold`¹, que nos permita determinar qué píxeles de la imagen pertenecen a la sección y cuáles al fondo de la imagen, quedando descartados 4.3:

$$dst(x,y) = \begin{cases} \maxVal & \text{si } src(x,y) > thresh \\ 0 & \text{e.o.c.} \end{cases}$$

¹Documentación de OpenCV sobre sus funciones de `threshold`: <http://docs.opencv.org/doc/tutorials/imgproc/threshold/threshold.html>

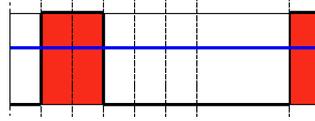


Figura 4.3: Operación de threshold gráficamente.

Aplicando la función al mapa de píxeles en escala de grises de la imagen, los que no superen la frontera especificada son puestos a 0. De ésta forma podemos describir ahora el contorno de la sección. Los puntos que no han superado la frontera se marcan como 0's y los puntos que tengan un valor superior al *threshold*, como 1's: la imagen queda binarizada y extraer los contornos ahora es buscar los puntos que tienen un valor opuesto como vecinos.

En la figura 4.4 podemos ver el resultado de aplicar el proceso de *threshold*.



Figura 4.4: Hileras H421 y H468 binarizadas tras el threshold

4.3. Eliminando ruido

La primera diferencia que encontramos al comparar los contornos de los planos con el contorno extraído por nuestra primera aproximación de segmentación está en la falta de suavidad de los bordes. El escaner nos ha hecho perder calidad (información) al no estar

normalizado completamente (diferentes posiciones de la pieza proporcionan diferentes resultados, debido a la iluminación). Una solución para paliar éste problema consiste, precisamente, en aplicar ruido² a la propia imagen antes de tratarla. El resultado se aprecia en 4.5

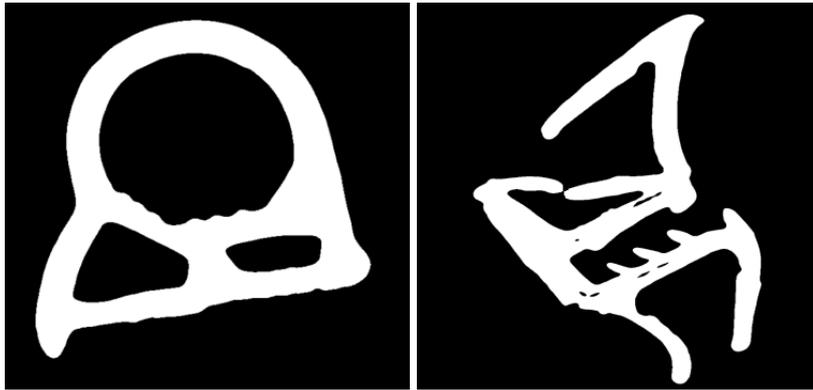


Figura 4.5: *Hileras H421 y H468 suavizadas y binarizadas tras el threshold*

²Documentación de OpenCV sobre procesos de filtrado: <http://docs.opencv.org/modules/imgproc/doc/filtering.html>

5. CAPÍTULO

Descriptores de formas

5.1. Conceptos generales

Como ya se ha comentado anteriormente, el problema del reconocimiento de formas es un campo en el que se ha trabajado desde hace tiempo. Es en 1972 cuando Granlund introduce la representación compleja en los descriptores [2] para identificar escritura a mano mediante descriptores, y un mes después Charles T. Zahn y Ralph Z. Roskies [6] aplican el método para describir una curva cerrada.

5.2. Descriptores de Fourier

Para comprender los descriptores de Fourier es preciso enunciar la principal idea del análisis de Fourier: podemos describir una función $f(x)$ como un sumatorio de series de senos y cosenos cuya frecuencia va aumentando. El resultado obtenido está en el que llamamos dominio de la frecuencia:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad n = 0, \dots, N-1 \quad (5.1)$$

Los descriptores se componen del conjunto de coeficientes que forman el resultado de aplicar la transformada de Fourier a los datos de entrada. En nuestro caso, la señal

de entrada será una función unidimensional que describirá el contorno del objeto que queremos identificar.

Una de las ventajas que nos ofrecen los descriptores de Fourier frente a otros descriptores de formas, es la facilidad de normalización que tienen, además de la capacidad de recuperar los datos de entrada. Si a un conjunto de descriptores les aplicamos la transformada inversa, volveríamos al conjunto de datos inicial:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \quad n = 0, \dots, N-1 \quad (5.2)$$

En nuestro caso, la transformada de Fourier inversa nos permitirá recuperar la descripción de un contorno desde sus descriptores.

5.3. Firmas unidimensionales

Una firma de forma unidimensional es cualquier función 1-D que represente o bien áreas o contornos de 2-D. En [7] se proponen cuatro firmas para representar una curva cerrada:

- **Coordenadas complejas:** conjunto de números complejos que describen los puntos de un contorno, siendo la coordenada x la parte real, y la coordenada y la parte imaginaria.

$$z(t) = x(t) + iy(t) \quad (5.3)$$

- **Distancia al centroide:** serie que representa las distancias de cada punto al centro de la curva.

$$x_c = \frac{1}{L} \sum_{t=0}^{N-1} x(t), y_c = \frac{1}{L} \sum_{t=0}^{N-1} y(t) \quad (5.4)$$

$$r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{\frac{1}{2}} \quad (5.5)$$

- **Firma de la curvatura:** está representada por la segunda derivada del contorno y la primera derivada de la tangente al contorno.

$$K(t) = \theta(t) - \theta(t-1) \quad (5.6)$$

donde:

$$\theta(t) = \arctan \frac{y(t) - y(t-w)}{x(t) - x(t-w)} \quad (5.7)$$

función que tiene discontinuidades cada 2π en el contorno, quedando reemplazada por la función:

$$K(t) = \varphi(t) - \varphi(t-1) \quad (5.8)$$

siendo $\varphi(t)$:

$$\varphi(t) = [\theta(t) - \theta(0)] \bmod(2\pi) \quad (5.9)$$

- Variación angular: reutilizando la descripción de la curva de las ecuaciones anteriores, las sumas angulares van recorriendo los puntos en sentido anti-horario, quedando la función normalizada:

$$\psi(t) = \varphi\left(\frac{L}{2\pi}t\right) - t \quad (5.10)$$

La principal característica de las cuatro firmas anteriores es que preservan la información de la curva: se puede reconstruir el contorno que representan a partir de una firma. En [7] analizan la utilización y precisión de éstas y otras firmas en el reconocimiento de contornos. En nuestro caso hemos realizado pruebas con las dos firmas que obtenían mejores resultados en [7], y hemos determinado que para nuestro caso la firma que más precisión nos ha dado es la representación compleja de coordenadas.

6. CAPÍTULO

Formación de una firma con el contorno

Una vez realizado el primer procesamiento de la imagen, habiendo filtrado parte del ruido y teniendo una segmentación de la sección que queremos identificar, el siguiente paso es extraer el contorno y formar una firma 1-D con el mismo. En primer lugar tendremos que tener en cuenta que una sección no tiene porqué estar formada por un único contorno, ya que puede contener tubulares interiores o pequeños contornos derivados de ruido o diferentes materiales que reflejen la luz en el proceso de digitalización. En nuestro caso buscaremos siempre el contorno más grande de la imagen (que no tiene porqué ser el exterior), e ignoraremos los otros contornos por el momento¹.

A pesar del primer filtrado que hemos realizado (desenfoque), el contorno con el que trabajemos puede estar todavía afectado por el ruido de la digitalización o por alteraciones del contorno al cortar la sección de la pieza.

Otro paso para intentar disminuir todavía más las alteraciones en el contorno es el aplicarle una reducción. Si un contorno está formado por 2000 puntos, es muy probable que las partes más características del mismo se puedan describir con un subconjunto pequeño y que la parte restante se centre en detalles muy específicos que realmente no contribuyan a su identificación. Por ello, del primer contorno que hemos extraído, llevamos a cabo una selección: cogeremos una serie de puntos del contorno realizando saltos, para quedarnos con la parte más general del contorno.

¹En el apartado de futuros trabajos a realizar y líneas de mejora se adelantan posibles puntos a trabajar teniendo en cuenta más contornos y las dificultades que conllevaría.

6.1. Simplificación del contorno

En la figura 6.1 podemos ver el contorno resaltado en rojo, de aplicar la reducción de puntos al contorno de la *hilera* H421. Vemos como según vamos ampliando el número de puntos a incluir en el contorno, se van incluyendo zonas de la sección que han quedado deformadas en el proceso de corte y segmentación, como su parte inferior.

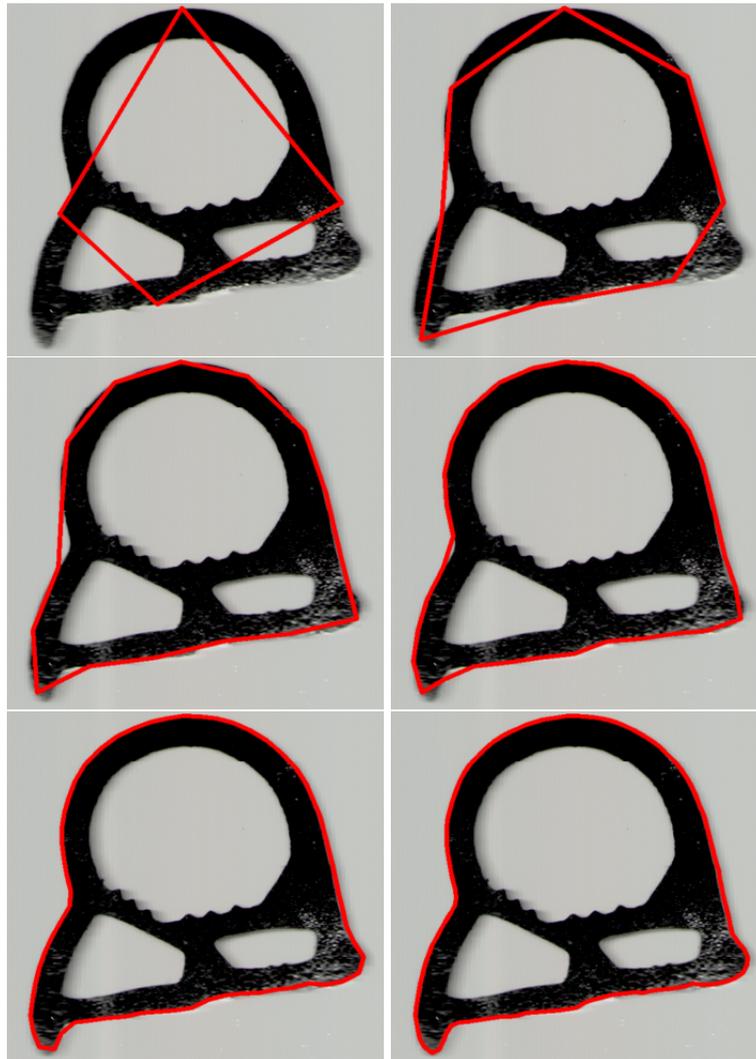


Figura 6.1: Contorno de la *hilera* H421 con 4, 8, 16, 32, 64 y 128 puntos

En las imágenes con las que hemos trabajado (a una resolución de 600 DPI), un número de puntos entre 128 y 256² nos da una descripción del contorno más que suficiente,

²La razón por la que se utilizan múltiplos de 2 es para más adelante poder ejecutar los algoritmos FFT (Fast Fourier Transform), pasando de una complejidad de $O(N^2)$ a $O(N \log N)$ en una muestra de N puntos

y nos permite reducir los efectos del ruido que hemos introducido.

6.2. Aplicación de la transformada de Fourier

Una vez extraído y simplificado el contorno, tenemos que formar una sucesión numérica unidimensional (firma) a la que aplicar la transformada de Fourier para poder extraer los descriptores. En nuestro caso, hemos utilizado la firma compleja, es decir, procesaremos las coordenadas de cada punto $P(x, y)$ del contorno para formar una sucesión de la forma:

$$s(n) = x(n) + iy(n) \text{ para } n = 1..N \text{ siendo } N \text{ el número de puntos del contorno} \quad (6.1)$$

Gráficamente, podemos ver el resultado de aplicar la transformada al contorno de 128 elementos extraído de la *hilera* H421 en 6.2 y los correspondientes a su plano en 6.3.

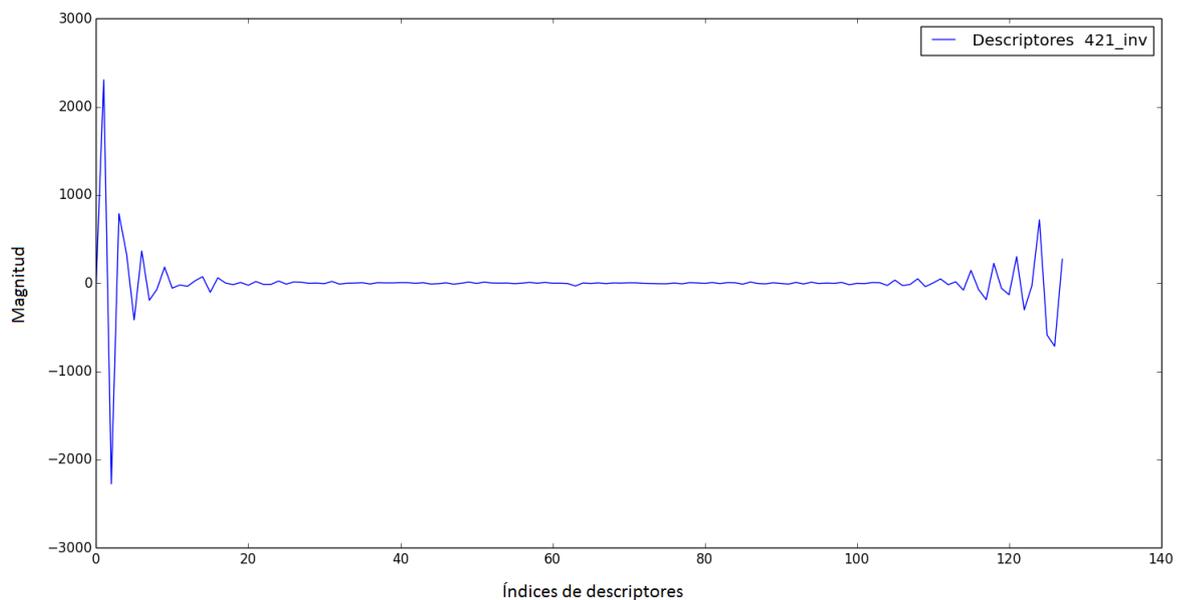


Figura 6.2: Valor de los descriptores de la *hilera* H421 con 128 puntos de contorno.

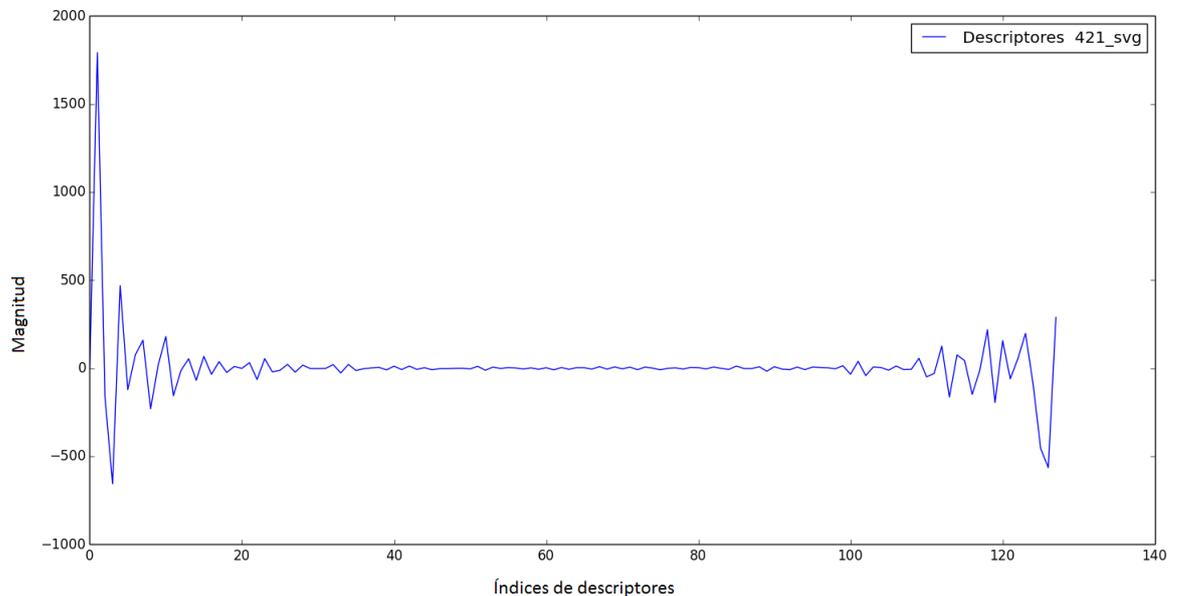


Figura 6.3: Valor de los descriptores del plano H421 con 128 puntos de contorno.

6.3. Invarianzas

Hasta el momento hemos extraído el contorno de las secciones, hemos aplicado una reducción al número total de puntos y con éste subconjunto hemos creado una firma compleja a la que hemos aplicado la transformada de Fourier. Pero los descriptores que hemos obtenido no guardan mucha similitud con los correspondientes del plano. Algunas de las razones son las siguientes:

- El tamaño de la imagen de la sección no se corresponde con el tamaño de su plano correspondiente. Hay una diferencia en el **escalado** de los mismos.
- La sección y el plano no tienen la misma orientación, es decir existe una **rotación** entre la sección digitalizada y el plano proporcionado.
- El plano y la sección no se encuentran en las mismas coordenadas, hay una **traslación** en cada una de las imágenes que afecta a cada coordenada.

Continuando la sección de firmas en [7], se nos explican los pasos para aplicar invarianza al escalado, rotación³ y traslación de los descriptores.

La invarianza a éstas tres características geométricas resulta difícil de conseguir en el dominio del espacio. La mayoría de las técnicas emplean gran cantidad de tiempo y cómputo para conseguirlas. Sin embargo, con los descriptores de Fourier la situación es distinta.

Las firmas presentadas anteriormente son invariantes a la traslación, por lo tanto los propios descriptores heredan ésta característica y son también invariantes a la traslación⁴. La invarianza a la rotación la conseguimos modificando los resultados obtenidos en el dominio de la frecuencia, es decir, alterando los descriptores para ignorar su información de fase y quedarnos sólo con la magnitud.

En nuestro caso, esto es, utilizando la firma de números complejos, el primer descriptor recibe el nombre de componente **DC**, y sólo aporta información relativa a la posición del contorno, con lo cual lo descartaremos. Para conseguir invarianza al escalado, dividiremos todo el conjunto de descriptores por el segundo descriptor. El vector de descriptores con las invarianzas a la traslación, escalado y rotación para la firma compleja, sería entonces:

$$f = \left[\frac{|FD_n|}{|FD_1|}, \dots \right] \text{ para } n = 2..N-1 \quad (6.2)$$

Para conseguir la invarianza en las otras firmas se puede consultar la referencia a [7], en la cual se citan las firmas de distancia al centroide, de curvatura y de variación angular con sus respectivos descriptores normalizados.

Aplicando las invarianza descrita anteriormente a los descriptores que teníamos calculados, vemos cómo las gráficas 6.4 de ambos descriptores ahora si tienen una tendencia similar.

³Aunque en un principio descartamos la información de rotación, más adelante veremos que nos puede resultar interesante preservarla para la conexión con la siguiente parte del proyecto: la comparación de la pieza con el plano

⁴Aunque anteriormente hemos mencionado que la traslación era una de las responsables de la poca similitud entre ambos conjuntos de descriptores, la naturaleza de la firma que hemos elegido nos hace evadir éste problema.

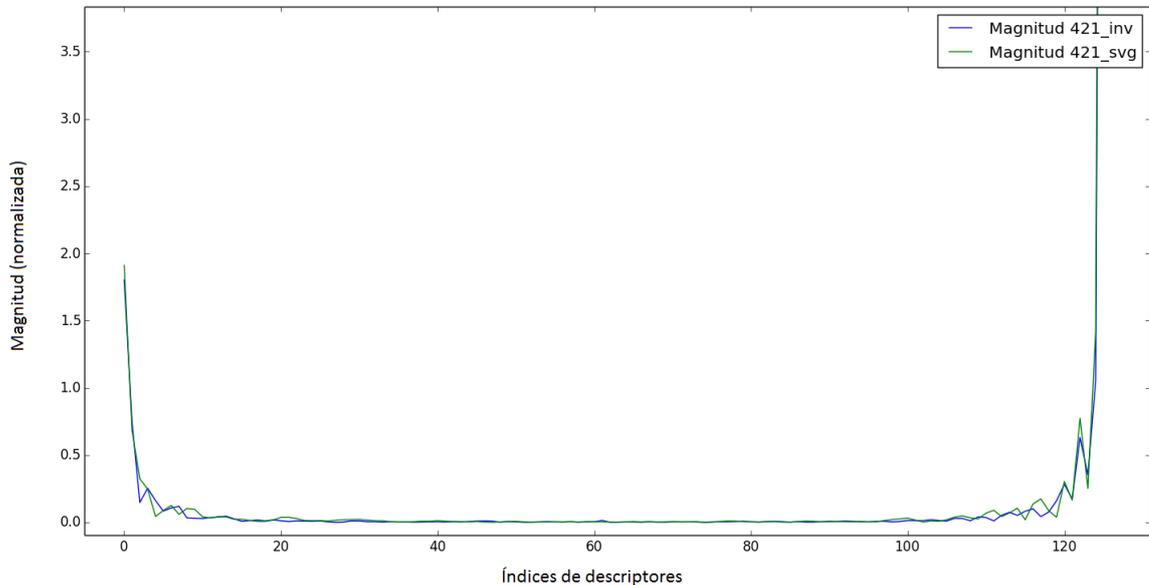


Figura 6.4: Valor de los descriptores la *hilera* H421 y su plano comparados.

6.4. Diseño

A continuación se detallarán los aspectos más importantes del diseño de la herramienta de identificación de perfiles:

- La interfaz de usuario debe ser sencilla, es posible que sea utilizada por empleados que no tengan elevados conocimientos de informática.
- El algoritmo debe ser configurable: si cambiamos el método de digitalización, los brillos o colores de las imágenes, se deben de poder modificar los parámetros para adecuarlos a las nuevas características.
- El conjunto de planos se ha facilitado inicialmente en ficheros, y es posible que se añadan o modifiquen los existentes de acuerdo con los cambios que se estimen en ingeniería.
- Se debe preparar la salida para facilitar la conexión entre ésta herramienta y la futura herramienta a desarrollar: el comparador entre la imagen y el plano.

Teniendo en cuenta los puntos anteriormente descritos, podemos dividir la herramienta en varias partes. Por un lado tendríamos la interfaz del usuario, desde la cual

podremos reconfigurar el proceso para adaptarlo a la entrada que estamos recibiendo. Por otra parte tendríamos los dos elementos principales del tratamiento de la imagen: la segmentación y el tratamiento y comparación de las firmas del contorno extraídas con las de los planos.

Para poder configurar la herramienta, se ha decidido crear un sistema de perfiles que nos permite determinar qué algoritmos se van a ejecutar y en qué orden. Inicialmente hemos incluido un juego de algoritmos básico que podrá ser ampliado fácilmente en un futuro según sea necesario. Para ello hemos creado una interfaz **Filtro** que pueden extender los diferentes tipos de filtros que queramos ir implementando.

Los cuatro filtros básicos iniciales que tendremos serán *grayscale*, *desenfoque Gaussiano* y *threshold*. A su vez, el desenfoque y el *threshold* tendrán unos parámetros de configuración que determinaran el *kernel*⁵ del desenfoque y el factor *frontera*⁶ del *threshold*.

El siguiente paso consiste en determinar la longitud del contorno y el número de descriptores de Fourier⁷ que vamos a utilizar. Para dar una idea del contorno simplificado que estamos obteniendo, dibujaremos sobre la imagen original el resultado del contorno extraído de la imagen segmentada. De ésta forma podremos estimar un número óptimo de puntos de contorno.

Para tener una visión en tiempo real de los efectos que provocan los procesos de segmentación con los filtros seleccionados y de extracción de contorno, se guardan tres buffers de cada imagen, que representarán la entrada original, la imagen segmentada y la imagen original con el contorno extraído. Las dos últimas se actualizarán al modificar alguno de los parámetros de segmentación o del contorno.

6.5. Implementación

De acuerdo a lo establecido en el diseño, se ha implementado la lógica del algoritmo por una parte y la interfaz de usuario por otra, quedando ambas combinadas en una aplicación C++/CLI.

Las partes de la interfaz de usuario están creadas en los fragmentos de código decla-

⁵El kernel determina el número de pixels que se samplean en cada pasada.

⁶Anteriormente, en la descripción del algoritmo *threshold*, lo hemos denominado *thresh*.

⁷En nuestro caso, enlazaremos el número de descriptores al número de puntos, pero más adelante comentaremos otras opciones

rados como "*managed*"⁸, por lo tanto el GC⁹ de .Net Framework se encargará de limpiar las referencias a objetos en memoria que ya no utilizemos. Nuestro algoritmo por otra parte estará programado en C++ en un fragmento de código "*unmanaged*"¹⁰ sin la influencia del entorno CLR¹¹ que afecta a la parte de interfaz de usuario de nuestro software.

Para una imagen dada, se aplicará un proceso de segmentación que irá determinado por las características del escáner y se buscará entre todos los planos disponibles, calculando sus descriptores normalizados. Después se computará la distancia euclídea entre cada descriptor de la imagen y del plano, y aquél que esté a la menor distancia determinará la clase asignada a la imagen de entrada. En la figura 6.5 se puede ver la separación entre las diferentes partes comentadas en el diseño:

- Segmentación: el TreeView de la parte derecha guarda la lista de filtros que hemos aplicado a nuestra imagen base (a la izquierda) para producir la salida segmentada (a la derecha).
- Análisis de contornos: la segunda parte de la interfaz incluye dos TrackBars que indican el número de puntos del contorno y el número de descriptores¹² así como la opción de dibujar el cuadrilátero mínimo que contiene a nuestro contorno.
- Clasificación: En ésta parte disponemos de un botón que iniciará la búsqueda entre todos los planos para determinar el más parejo a la imagen introducida de acuerdo con los descriptores de Fourier del contorno.

⁸Código administrado.

⁹Garbage Collector

¹⁰Código no administrado.

¹¹Common Language Runtime

¹²Actualmente están auto-configurados para elegir un número compatible con los algoritmos FFT. No obstante, puede resultar muy interesante seleccionar un número arbitrario de descriptores independientemente del número de puntos del contorno.

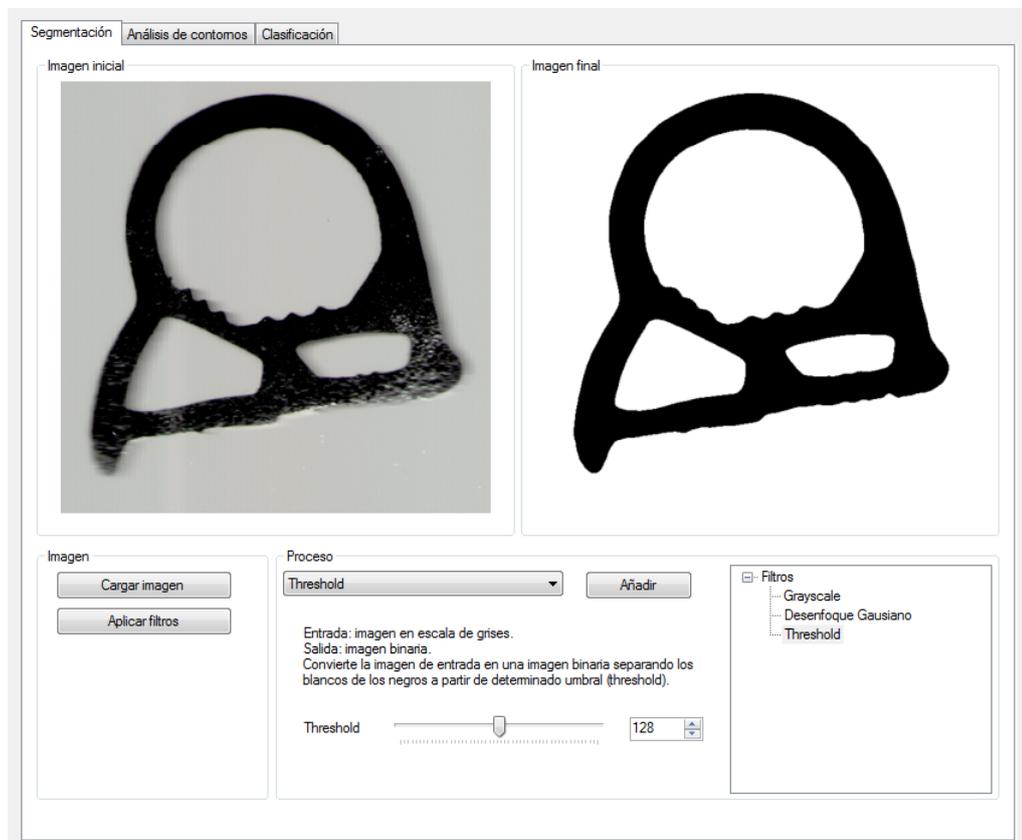


Figura 6.5: Vista de la interfaz de usuario

7. CAPÍTULO

Comparación con el plano

Una vez hemos identificado la pieza con la que estamos trabajando podemos pasar al siguiente paso del control de calidad: comparar la imagen que hemos obtenido con su plano original.

Éste proceso se lleva a cabo actualmente manualmente: se tienen unos planos impresos a escala y se compara a trasluz la desviación entre la pieza y su plano. Algunos de los beneficios que pretendemos ofrecer con éste nuevo sistema son:

- Aumento de **precisión**: eliminaremos la dependencia a la escala de las piezas fabricadas, de forma que podremos digitalizarlas en imágenes más grandes con mayor calidad (más DPIs) y mostrar un plano con el tamaño correspondiente sin perder calidad¹
- **Velocidad** en el proceso: aplicando las ampliaciones necesarias podremos determinar rápidamente si una parte de la sección coincide exactamente con los requerimientos de la especificación.
- **Almacenaje** de históricos: el almacenamiento digital de las secciones ofrecerá una ventaja principal frente al sistema actual, y es que no ocupará el espacio físico que se requiere para guardar el histórico de cada día de producción.
- **Acceso** a históricos: acceder a una pieza determinada con sus datos de fabricación (como su fecha) resultará una tarea rápida al estar guardada en la base de datos.

¹La característica principal de los ficheros SVG es que pueden escalarse la cantidad que se desee, ya que su rasterización es un proceso dinámico.

No será necesario buscar manualmente entre el histórico de piezas para encontrar la fabricación de una fecha determinada.

- Toma de **cotas**: con el sistema digital añadiremos la opción de tomar cotas que facilitarán la validación de secciones así como de las zonas en las que hay que modificar el proceso de producción para ajustar los parámetros de fabricación.
- **Portabilidad**: al diseñar el sistema en HTML5 y Javascript, podremos visualizar la comparación de una pieza con su plano con prácticamente cualquier navegador.

7.1. Diseño

Los parámetros de entrada se nos facilitarán en la URL del navegador, ya la especificación que se nos ha dado determina los siguientes parámetros:

- *imagen*: determina la imagen que vamos a cargar y que representa la pieza que vamos a visualizar.
- *hilara*: es el nombre del plano en formato SVG que representa la imagen que se nos ha facilitado.
- *anchura*: la anchura de la imagen referenciada en cm.
- *altura*: altura de la imagen en cm.
- *posiciones*: conjunto de datos separados por el carácter "|" que determinan la posición en la que se debe de cargar la imagen, así como si está rotada o volteada horizontalmente. Tiene campos que no son utilizados en la actualidad pero que se han dejado como *placeholders* para un futuro.
- *operador*: número de operario (interno a la empresa) que identifica a quién está almacenando una posición.
- *fechareal*: campo de fecha que nos permite almacenar una fecha diferente a la del momento en que se realiza la digitalización (en ocasiones hay algún retraso o se pospone una digitalización).

Para dotar del máximo dinamismo a la herramienta y facilitar tareas como la carga de la imagen y del plano, se han utilizado las librerías **jQuery** y **jQuery.svg**. Aunque

la mayoría de navegadores acepta la carga nativa de documentos SVG, Internet Explorer 10 mostró algún problema con los mismos, y dado que las máquinas de la empresa actualmente tienen instalado dicho navegador, se utilizó la librería **jQuery.svg**² para evitar problemas de compatibilidad. Para el sistema de cotas y manipulado de la imagen, se ha utilizado la librería **Paper.js**³. Más adelante describiremos el papel que cumple cada librería.

El diseño del "banco de trabajo" constará entonces de dos capas:

1. Canvas *imgload*: formado por la imagen cargada y sobre el que se capturarán los eventos principales del ratón, tanto para arrastrar la imagen como para rotarla y poder situarla en el plano.
2. Canvas *svgload*: estará sobrepuesto al canvas de la imagen y en él cargaremos el plano de la imagen. Para poder visualizar la imagen que se encuentra por debajo, dotaremos de transparencia al plano SVG.

7.2. Herramientas

- jQuery (de su página oficial⁴): jQuery es una rápida y ligera librería de Javascript con muchas características. Hace que la manipulación de elementos HTML, el manejo de eventos, la animación, y las llamadas Ajax sean mucho más sencillas con una API que funciona en multitud de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que mucha gente escribe Javascript.
- jQuery.svg: Es un plugin de jQuery que nos permite manipular documentos SVG desde Javascript. Requiere una versión modificada de jQuery facilitada en la página oficial del plugin.
- Paper.js: es un framework de scripting para gráficos vectoriales que funciona sobre un canvas HTML5. Ofrece una escena y modelo de objetos así como muchas funcionalidades para trabajar con gráficos vectoriales y curvas bezier, todo ello programado en una interfaz consistente y limpia.

²Enlace a la dirección de jQuery.svg: <http://keith-wood.name/svg.html>

³Enlace a la página oficial de Paper.js: <http://paperjs.org/>

⁴Página de JQuery: <http://jquery.com/>

7.3. Funcionalidades

Para facilitar la utilización de la herramienta y poder realizar las tareas de comparación de la sección con el plano de la forma más similar a la que utilizan en la actualidad, se han añadido una serie de funcionalidades:

- Rotación relativa a un **punto**: utilizando la combinación Control + mouse derecho, se coloca un puntero amarillo que indica el punto central de la rotación de la imagen. La razón de añadir ésta función va ligada a la rotación que se efectúa en ocasiones con los planos impresos: interesa tener un punto fijo y realizar pequeñas rotaciones relativas a éste punto.
- Modo de **precisión** en la rotación: pulsando la tecla ALT + la rueda del mouse, podemos realizar rotaciones más pequeñas ideales para los giros finales que requieren ya cuadrar la imagen en el plano.
- Sistema de cotas: al presionar el botón derecho del mouse podremos trazar líneas de acotación que nos permiten medir en directo partes de la sección. Adicionalmente, si se pulsa la tecla shift, se puede trazar una polilínea que marca la cota total de las líneas trazadas, y que puede resultar útil para medir secciones curvadas. Con la tecla d se eliminan las cotas trazadas en el plano.
- Cambios en el color del plano: en algunas circunstancias se ha encontrado que el color inicial establecido para trazar el plano no se distingue bien de la imagen que estamos tratando. Para ello se ha añadido una paleta que nos permite repintar al momento el plano y evitar problemas de contraste.
- Sistema de **almacenamiento** en intranet: el botón guardar nos lanza un formulario que permite al operario identificarse e introducir comentarios relativos a la pieza que se está tratando y que serán analizados posteriormente. El sistema de almacenamiento de ésta información está implementado en la intranet mediante una página ASP.net a la cual realizamos un post con la información relativa a la posición de la pieza y los datos introducidos en el formulario anteriormente comentado.
- Botones de zoom: para poder ampliar el detalle de las partes que nos interesen de la sección y el plano, se han añadido dos botones que realizan un zoom y reescalado de la imagen y plano.

- Volteo horizontal: puede suceder que la imagen digitalizada esté invertida horizontalmente con respecto al plano, ya que el operario no tiene porqué conocer su orientación original. Éste botón reorienta la pieza y guarda un flag en el vector *posiciones* para que cuando carguemos un histórico la imagen siga volteada.
- Rotar 180°: para evitar la rotación completa con la rueda del mouse en caso de una mala orientación inicial, se ha añadido un botón que nos permite rotar instantáneamente 180° la pieza.
- Enlace plano-imagen: para poder desplazar la pieza y el plano de forma simultánea se ha añadido un botón que nos permite enlazarlos y aplicar las transformaciones a ambos objetos a la vez. Resulta útil en el caso de planos que contienen más de una pieza o más de una vista, ya que puede ser que la imagen que estamos tratando esté en una parte inferior del plano.

8. CAPÍTULO

Conexión entre ambas herramientas

Con el visualizador de planos SVG ya preparado, nos surge una primera idea a la hora de conectar ambos proyectos. La tarea que más está costando durante las pruebas es la de orientar la pieza con respecto al plano. El proceso de rotación es lo que más tiempo requiere a pesar de las herramientas que hemos facilitado para acelerarlo.

A la hora de normalizar los descriptores de Fourier, estamos descartado la información de fase para conseguir la invarianza a la rotación. A pesar de que actualmente estamos obteniendo buenos resultados, estamos provocando unos efectos laterales que más adelante nos pueden llevar a falsos positivos, ya que la mayoría de la información que almacenan los descriptores se encuentra, precisamente, en su fase. La figura 8.1 aparece en [1] y muestra uno de los efectos laterales antes comentados: ambos contornos tienen exactamente la misma magnitud en cada uno de los descriptores, lo único que los diferencia es la información de fase.

Si utilizásemos el algoritmo con la normalización que estamos usando hasta ahora en los descriptores, nuestro algoritmo devolvería un matching perfecto. Es decir la distancia euclídea entre las magnitudes de ambos contornos sería 0. La propuesta de [1] difiere de los métodos que se habían utilizado hasta entonces en que no intenta des-rotar los descriptores, sino calcular la rotación relativa entre ambos juegos de descriptores. Comenzando con el teorema de la correlación cruzada:

$$\int_{-\infty}^{\infty} \bar{c}_1(l)c_2(\Delta l + l)dl = \int_{-\infty}^{\infty} \bar{C}_1(n)C_2(n)^{i2\pi n\Delta l} dn \quad (8.1)$$

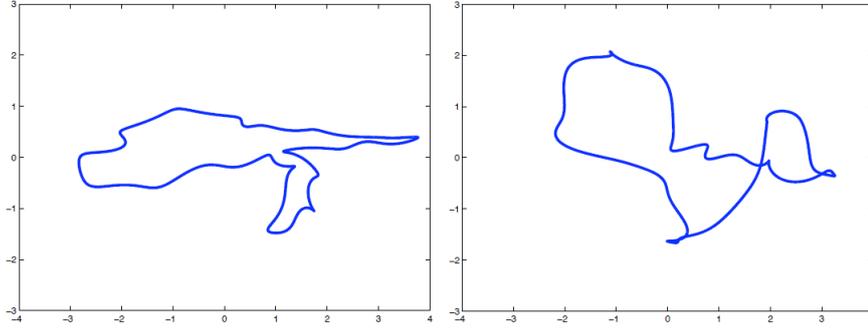


Figura 8.1: Contornos con misma magnitud en todos los descriptores

Cambiando la integral infinita por una discreta:

$$(c_1 \star c_2)(\Delta l) = \int_0^L \bar{c}_1(l)c_2(\Delta l + l)dl = \sum_{n=0}^{\infty} \bar{C}_1(n)C_2(n)e^{\frac{i2\pi n\Delta l}{L}} \quad (8.2)$$

Utilizando ahora el teorema de la correlación calculamos los mínimos cuadrados entre dos juegos de descriptores sin estimar explícitamente los parámetros. Empezamos suponiendo que $c_2(l) = e^{i\phi}c_1(l - \Delta l)$. Calculamos ahora, una aproximación de la correlación $r_{12} = (c_1 \star c_2)$ como la transformada inversa finita de Fourier:

$$r_{12} \approx F^{-1}\{\bar{C}_1 \cdot C_2\} = \sum_{n=0}^N \bar{C}_1(n)C_2(n)e^{\frac{i2\pi n\Delta l}{L}} \quad (8.3)$$

El mínimo error cuadrático que relaciona c_1 y c_2 bajo rotaciones y desplazamiento del origen, que denotaremos como τ viene dado por:

$$\min_{\tau} \|c_1 - \tau c_2\|^2 \approx 2 - 2\max_l |r_{12}(l)| \quad (8.4)$$

Para extraer los parámetros de τ , se puede estimar la posición y el ángulo de la fase maximizando:

$$\begin{aligned} \Delta l &\approx \operatorname{argmax}_l |r_{12}(l)| \\ \phi &\approx \operatorname{arg}(r_{12}(\Delta l)) \end{aligned} \quad (8.5)$$

Para el caso de $N = \infty$ todas las aproximaciones son igualdades. Supondremos para

Hilera	Desplazamiento Δ	Rotación ϕ
561	55	-76.51
973	61	-176.09

Tabla 8.1: Desplazamientos y ángulos de las secciones con respecto a sus planos.

las siguientes ecuaciones que estamos trabajando con $N = \infty$. Empezaremos calculando la correlación cruzada entre c_1 y c_2 a través de sus descriptores de Fourier:

$$\begin{aligned}
r_{12} &= F^{-1}\{\bar{C}_1 \cdot C_2\} \\
&= F^{-1}\{\bar{C}_1(n)e^{i\phi C_1(n)}e^{-\frac{i2\pi n\Delta}{L}}\} \\
&= e^{i\phi F^{-1}\{|C_1(n)|^2 e^{-\frac{i2\pi n\Delta}{L}}\}} \\
&= e^{i\phi r_{11}(l-\Delta l)}
\end{aligned} \tag{8.6}$$

Al tener la función de auto-correlación r_{11} valores reales y su máximo en 0. La rotación ϕ y desplazamiento Δl de los descriptores se puede calcular con las ecuaciones anteriores.

Si c_2 no es una c_1 transformada, la correlación cruzada máxima será inferior a 1, y en caso de que sea un una transformación "perfecta", la correlación será de 1.

Añadiendo las modificaciones al algoritmo para tener en cuenta la información de fase de los descriptores podemos calcular la rotación de un objeto A con respecto a un objeto B realizando un análisis de la correlación cruzada de su descriptores. Si superponemos el valor de los descriptores conservando la información de fase obtenemos, para las figuras 561 y 973 los resultados de 8.2 y 8.3.

En ambas figuras se aprecian los efectos de la fase en el valor de los descriptores. Si ahora calculamos los parámetros Δ y ϕ con (8.5), podemos calcular el ángulo y el desplazamiento (cambio de starting point).

La tabla 8.1 muestra los valores obtenidos para los parámetros Δ y ϕ al calcular la transformación de los descriptores de la imagen con respecto a su plano utilizando (8.5).

Éstos nos permitirán orientar automáticamente la imagen de la sección identificada con respecto al plano que representa de una manera bastante precisa (estaremos maximizando la correlación), como se puede ver en las figuras 8.6 y 8.7.

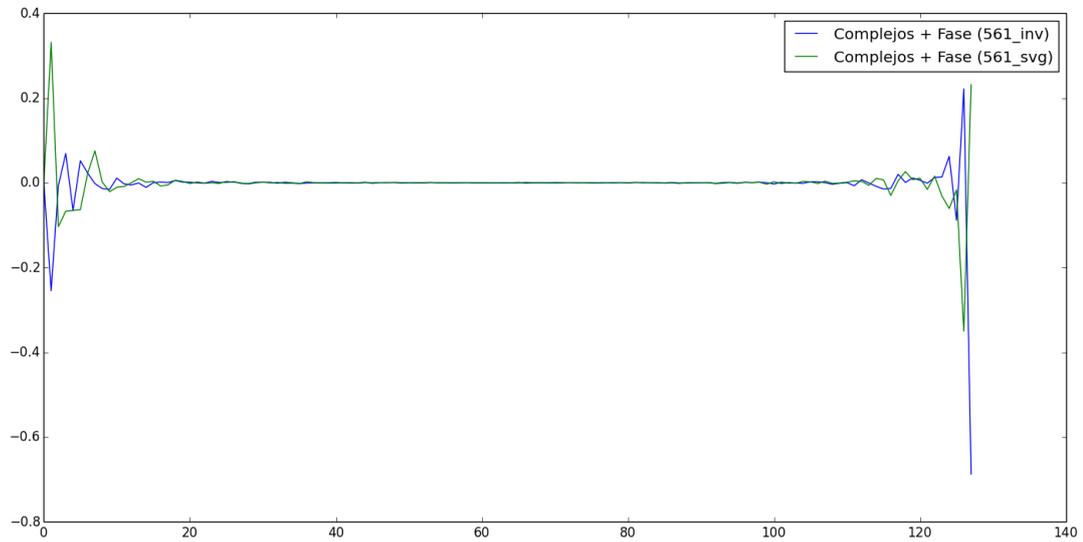


Figura 8.2: Comparación de los descriptores de la *hilera* 561 con su plano conservando la información de fase.

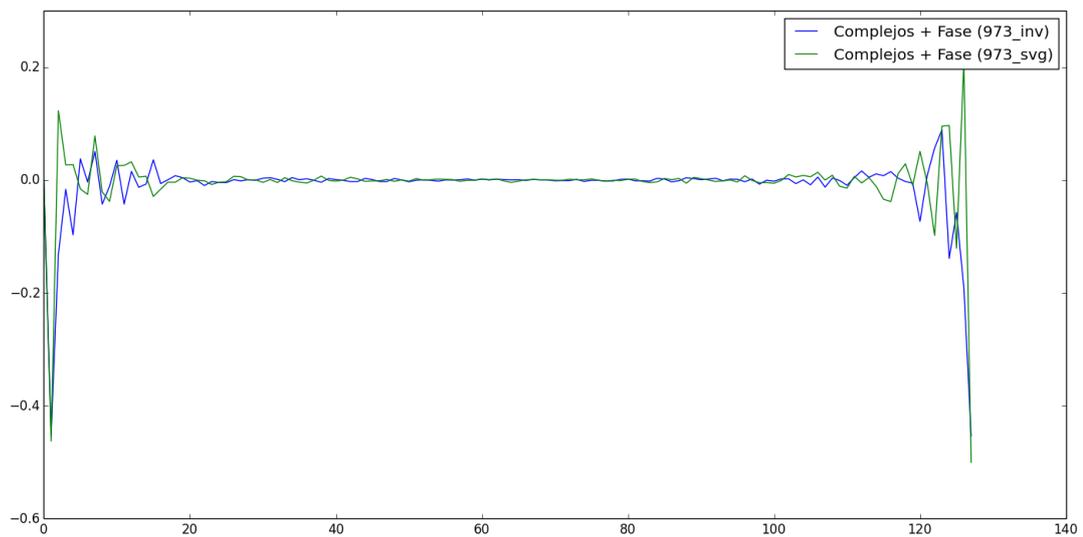


Figura 8.3: Comparación de los descriptores de la *hilera* 973 con su plano conservando la información de fase.

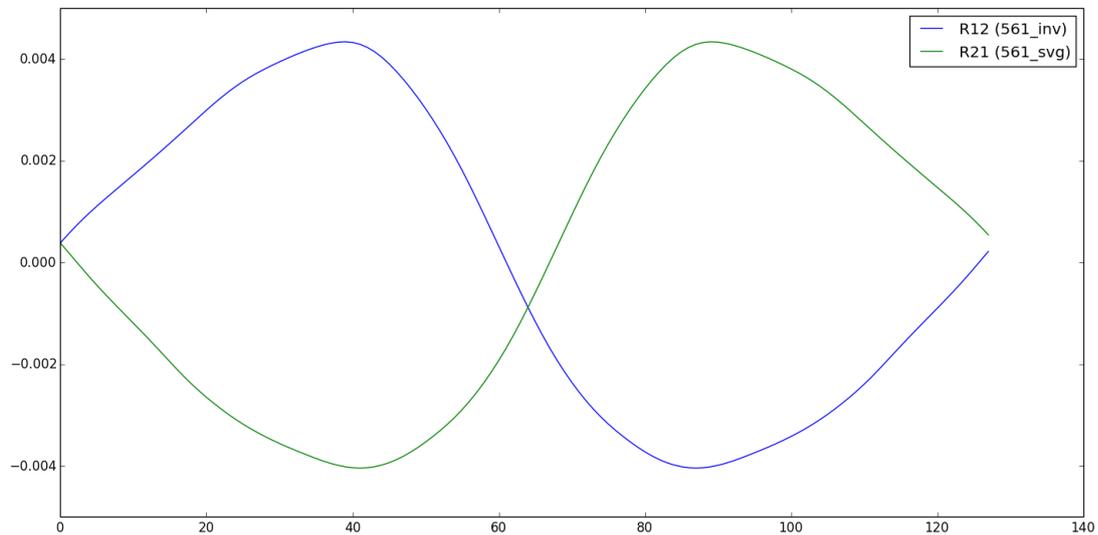


Figura 8.4: Desplazamiento en el starting point provocado por la rotación entre la imagen 561 y su plano.

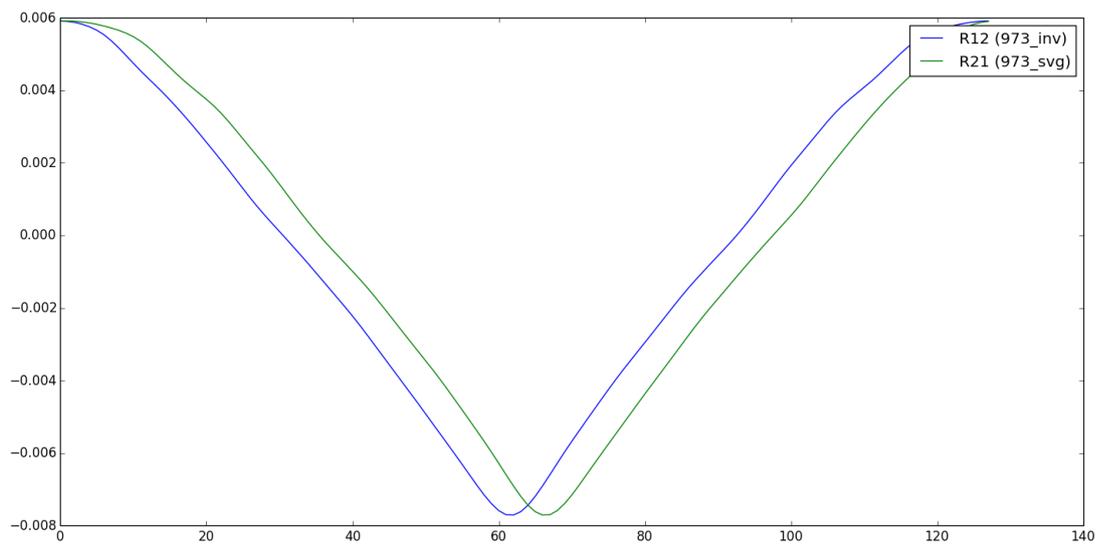


Figura 8.5: Desplazamiento en el starting point provocado por la rotación entre la imagen 973 y su plano.

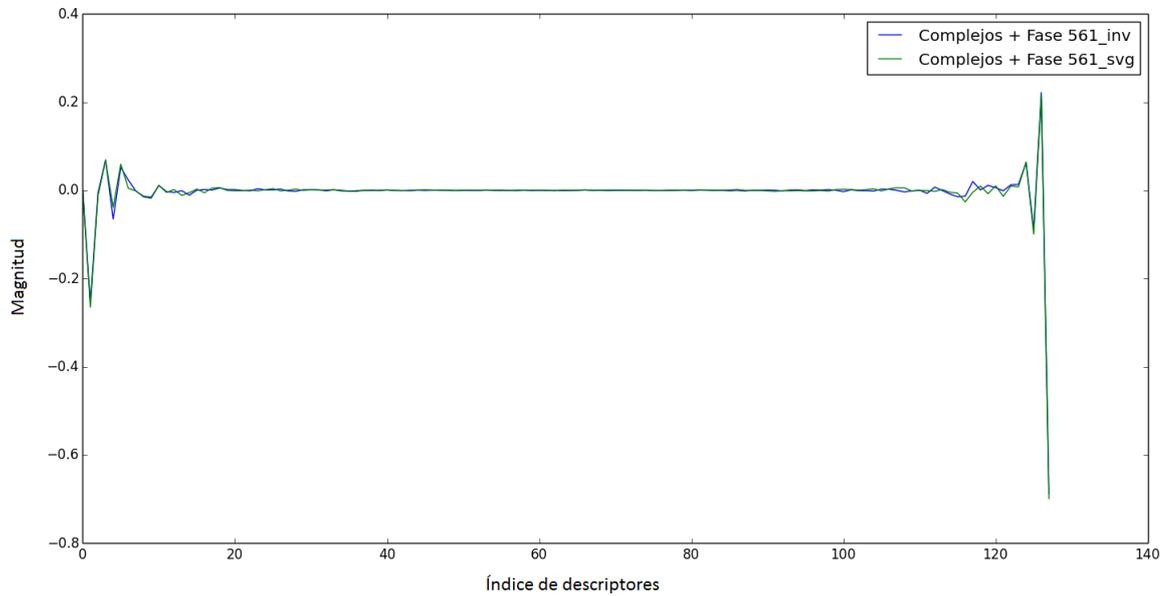


Figura 8.6: Comparación de los descriptores de la *hilera 561* con su plano conservando la información de fase y habiéndolos rotado.

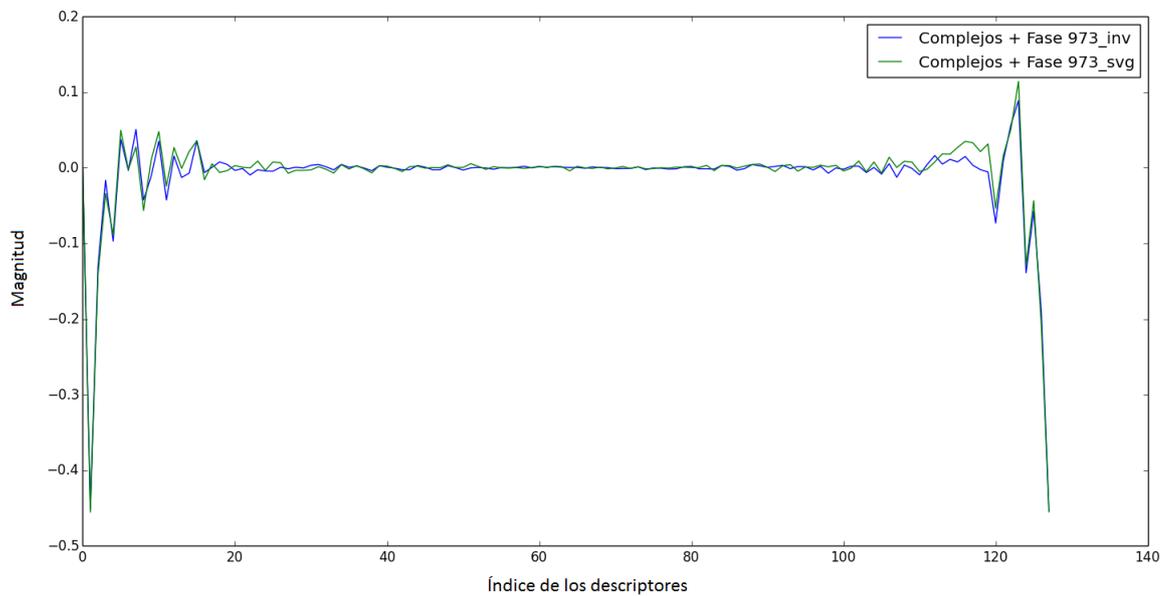


Figura 8.7: Comparación de los descriptores de la *hilera 973* con su plano conservando la información de fase y habiéndolos rotado.

En las figuras 8.8 se aprecia que los ángulos devueltos por el algoritmo se aproximan bastante bien a la rotación relativa del plano de la sección.

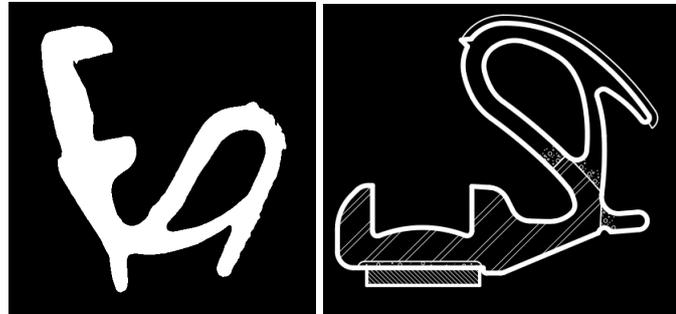


Figura 8.8: Rotación de la *hilera* 561 con respecto a su plano. La estimación del algoritmo es de -76.51°

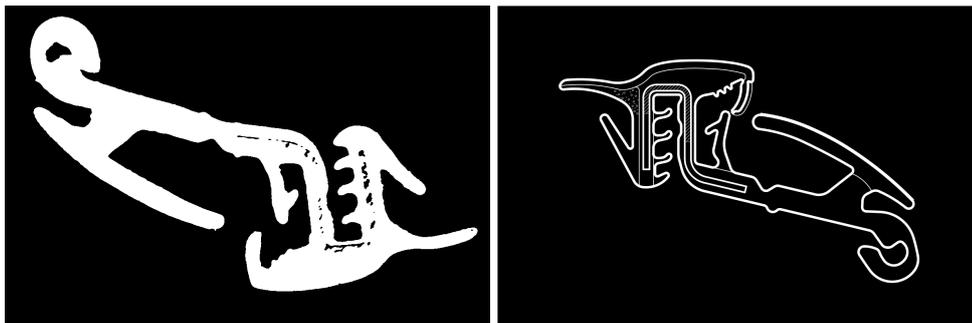


Figura 8.9: Rotación de la *hilera* 973 con respecto a su plano. La estimación del algoritmo es de -176.09°

Preservando la información de fase de los descriptores hemos conseguido dos cosas:

- Evitar los falsos positivos al utilizar únicamente la información de magnitud, como hubiese sucedido con el caso de la figura 8.1.
- Recuperar la rotación relativa de los descriptores de la sección con respecto al plano, para poder pasarlo como parámetro al visor de planos y llevar a cabo una orientación inicial.

8.1. Pruebas

De cara a la realización de pruebas se han incluido los siguientes métodos de clasificación:

Hilera	Complejos + Magnitud	Distacia (Mag)	Complejos + Fase	Correlación	Hu
014	014	0,063	014	0,986	367
254	254	0,1157	254	0,9975	254
421	421	1,6233	421	0,9996	421
463	463	7,3708	463	0,9634	241
468	468	0,1891	468	0,9044	242
561	561	0,0436	561	0,9954	750
749	959	0,7678	959	0,9768	787
944	944	0,0367	944	0,9966	944
973	994	0,1063	994	0,9818	593
993	993	0,0176	993	0,9960	966
996	996	2,8852	996	0,9983	871

Tabla 8.2: Clasificación de las piezas según los distintos algoritmos

- Descriptores de Fourier con firma compleja normalizados para utilizar sólo su información de magnitud (fase descartada) y comparados mediante distancia euclídea.
- Descriptores de Fourier con firma compleja normalizados para preservar la información de fase y comparados mediante correlación cruzada.
- Momentos de imagen Hu[3] basados en el contorno.

Todas las imágenes de las secciones nombradas a continuación han sido tomadas con el mismo escáner, *Epson Stylus SX100*, a 600 DPI. Asimismo, todas están sometidas al mismo proceso de segmentación consistente en los siguientes pasos:

1. **Grayscale:** La imagen se ha transformado a escala de grises inicialmente.
2. **Desenfoque:** Se ha aplicado un desenfoque utilizando un *kernel* de 21x21 con el objetivo de reducir el ruido en los contornos.
3. **Threshold:** Se ha binarizado la imagen con un valor frontera de 127.

En cuanto al contorno, se han seleccionado 256 puntos distribuidos uniformemente por el contorno extraído y 256 descriptores de Fourier. Los resultados de los tres algoritmos se puede ver en la tabla 8.2.

De la tabla se desprenden las conclusiones: los descriptores de Fourier resultan mucho más precisos identificando contornos que los momentos Hu, al menos en las pruebas realizadas: ambos juegos de descriptores han tenido un 81,8% de precisión mientras que

los momentos Hu sólo han alcanzado un 27,2%. Es importante mencionar que las *hileras* en las que ha fallado la clasificación a través de los descriptores tienen unos planos muy similares 8.10 a la pieza que estábamos identificando. Ésto nos abre futuros caminos a investigar a mejorar el algoritmo actual.

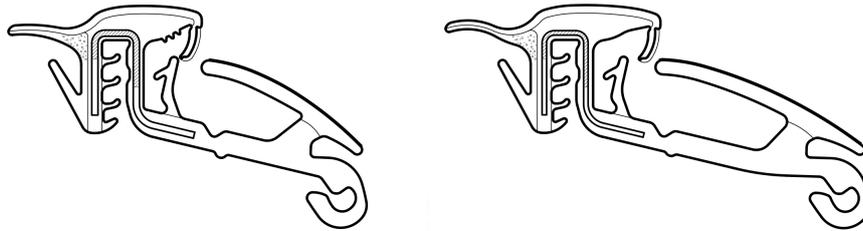


Figura 8.10: Comparación entre los planos 973 y 994, cuyos descriptores han identificado de forma incorrecta.

Al abrir el enlace que se nos muestra con los planos candidatos sugeridos se nos abrirá la página con la imagen 8.11 que hemos procesado, con una rotación ya aplicada que trata de igualarla a la del plano, y en el que podemos tomar mediciones de la figura a escala (en el ejemplo se ha tomado una cota de 8,035mm).

Anexos

A. ANEXO

Desviaciones en la planificación

La planificación inicial no ha sufrido grandes cambios con respecto a la real. No obstante, las herramientas determinadas para llevar a cabo el proyecto sí requirieron un tiempo formativo superior al estimado.

El correspondiente diagrama gantt final sería el de la figura [A.1](#) y el EDT completo con los procesos formativos se vería en la figura [A.2](#).

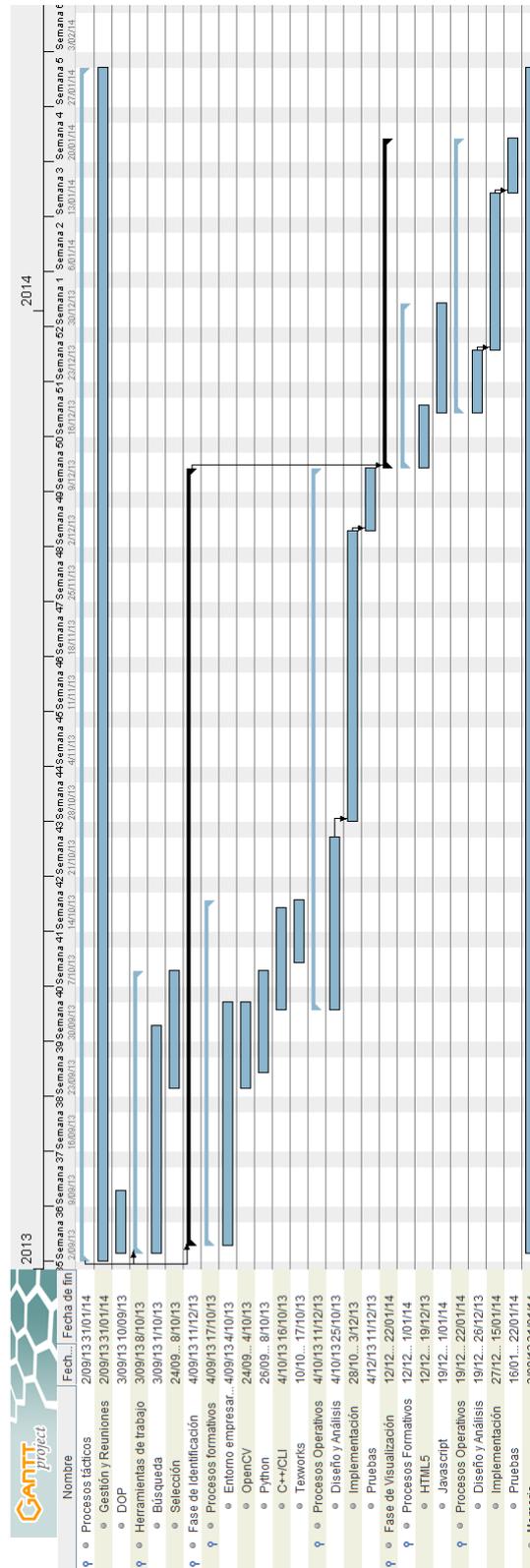


Figura A.1: Diagrama Gantt final.

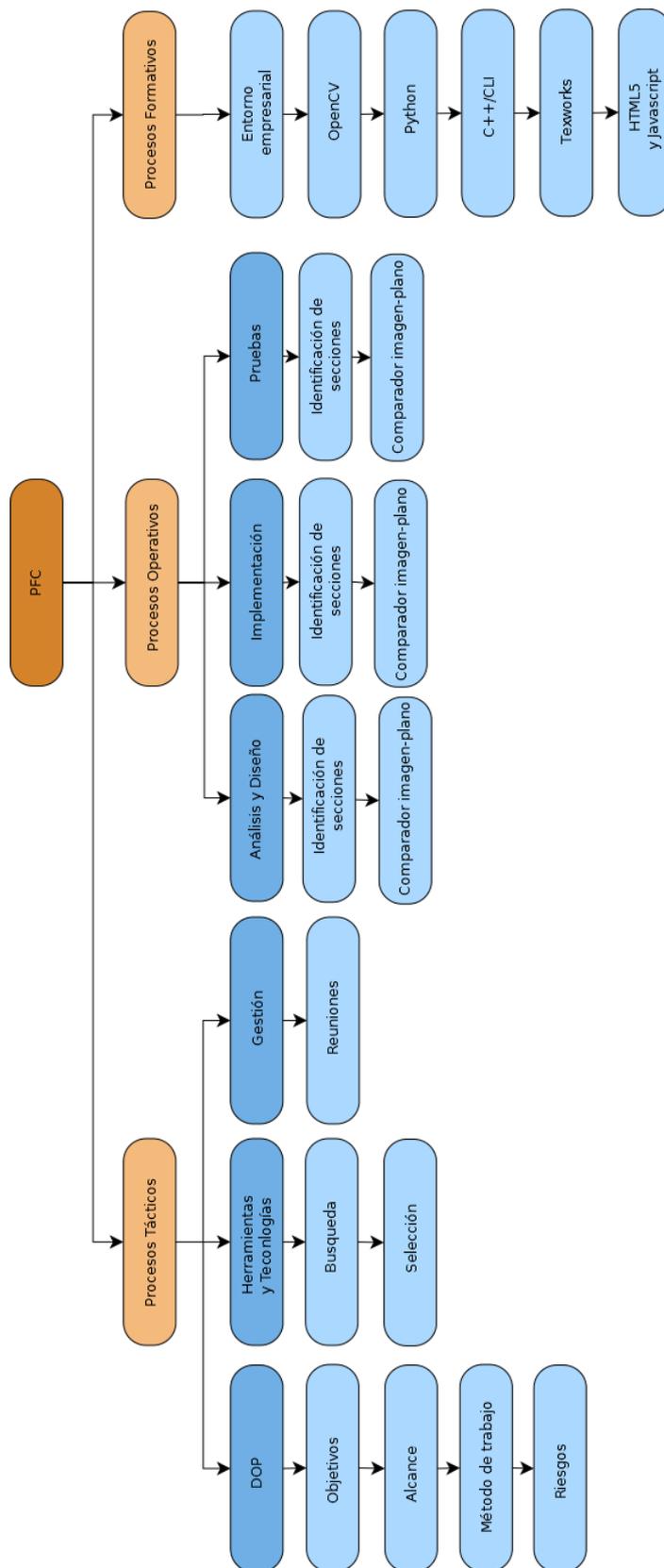


Figura A.2: Diagrama *EDT* con las tareas formativas completadas.

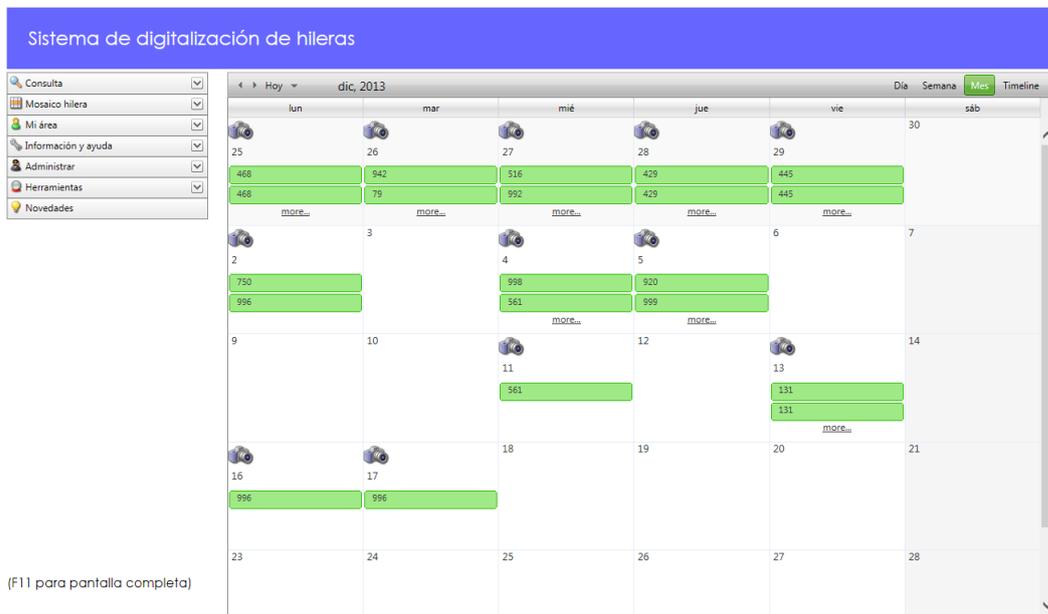


Figura B.2: Vista de calendario con las hileras digitalizadas durante los días marcados

C. ANEXO

Normalización de la iluminación

En algunas de las *hileras* escaneadas, notamos que al no hacer un contacto completo con la superficie (en parte por no haberse realizado un corte perfecto) se formaban sombras en algunas partes y excesos de luz en otros.

Para atajar el problema de las sombras (cuestión que dificultaba la visión en el visor de planos) se adquirió un sistema de iluminación LED¹, que será adaptado como tapa del escáner con una extensión fabricada por una impresora 3D.

¹Página de referencia de sistema de iluminación LED:<http://www.infaimon.com/catalogo-industria/iluminacion/led/iluminacion-frontal-lateral-directa/paneles-leds-701-p16233.html>

Conclusiones

En éste proyecto se ha intentado dar forma a una primera aproximación en la clasificación de imágenes con una información inicial muy limitada. Si bien el número de clases distintas es del orden de 1000 (H000 a H999), la disponibilidad del plano de cada sección no nos aporta mucha información a parte de la que podemos extraer de su contorno.

El color de las piezas si bien ha provocado que la segmentación sea más sencilla o por lo menos no tan vulnerable al entorno de escaneado, no nos ha proporcionado mucha información adicional que nos permita distinguir los diferentes modelos de pieza, aunque si hemos apreciado componentes metálicos en algunos modelos.

Los descriptores de Fourier tienen unas propiedades que los convierten en herramientas idóneas para la descripción de señales continuas, como los contornos cerrados que hemos descrito. La invarianza a escalado, rotación y traslación resulta fácil de obtener, e incluso utilizando el método de la correlación cruzada recuperar transformaciones relativas entre dos conjuntos de descriptores nos proporcionará información que nos puede resultar de utilidad.

D.1. Futuros puntos de trabajo

- Modificar el número de descriptores: la idea es eliminar aquellos de mayor frecuencia que aportan información sobre detalles.

Supongamos que tenemos un juego de 64 descriptores de un contorno: los prime-

ros nos aportarían una idea general del contorno mientras que los últimos detalles precisos sobre las características de la curva. Éstos últimos descriptores pueden ser descartados para eliminar ruidos introducidos durante el proceso de digitalización. Éste proceso sería interesante en conjunto con el sistema de segmentación actual.

- Incorporar información sobre contornos adicionales: actualmente sólo se utiliza el contorno exterior de la imagen. En algunos casos (aquellos en los que el clasificador falla) es posible que haya características internas en la pieza (como tubulares) que nos permitan discriminar los planos incorrectos. El desafío que plantea es que en los planos que se nos han proporcionado hay trazos y líneas que dificultan la detección "limpia" de éstos tubulares y demás regiones internas.

Una posible forma de trabajo sería utilizar las imágenes segmentadas para almacenar información de contornos internos, ya que en éstas imágenes se aprecian mejor que en los planos.

- Extraer más características del análisis de Fourier: al igual que hemos calculado el desplazamiento del punto inicial y la rotación de la imagen con respecto al plano, podría resultar interesante calcular el escalado una vez hayamos alcanzado un nivel de precisión más alto para realizar un análisis geométrico completo de la pieza.

Así tendríamos una comparación directa entre el plano y el contorno representado nada más realizar el primer análisis.

- Migrar los datos a una base de datos: ésta parte de mejora ya se ha empezado a hacer, así evitamos abrir todos los ficheros cada vez que queremos identificar una figura. De hecho, en la base de datos actual se guardan los descriptores de Fourier de cada plano, de forma que únicamente hay que calcular los del contorno extraído de la pieza tratada.
- Tratamiento de las piezas que son identificadas incorrectamente: realizar algún tipo de aprendizaje o correlación entre la identificación incorrecta de una pieza y su plano real. Es posible además que una pieza se haya sometido a deformaciones que están dentro de los parámetros de producción pero que provoquen una discrepancia lo suficientemente grande como para ser identificada incorrectamente con respecto a su plano.

Bibliografía

- [1] Michael Felsberg Fredrik Larsson and Per-Erik Forssén. Patch contour matching by correlating fourier descriptors. *Digital Image Computing: Techniques and Applications (DICTA)*, 2009.
- [2] G. H. Granlund. Fourier processing for hand print character recognition. *IEEE Transactions on Computers*, c-21:195–201, 1972.
- [3] Ming-Kuei Hu. *Visual Pattern Recognition by Moment Invariants - IRE Transactions on Information Theory*. 1962.
- [4] Intel. Open computer vision library. <http://www.opencv.org>.
- [5] Guido van Rossum. Python programming language. <http://www.python.org>.
- [6] Charles T. Zahn and Raplh Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, c-21:269–281, 1972.
- [7] Dengsheng Zhang and Guojun Lu. A comparative study on shape retrieval using fourier descriptors with different shape signatures. pages 2–4.