UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

título del TFG

Presentado por Nombre del alumno
en Universidad de Burgos — 29 de septiembre
de 2016
Tutor: nombre tutor

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Nombre del alumno, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 29 de septiembre de 2016

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Introducción

1.1. Gestores de tareas:

Trello

Es una pizarra virtual también conocida como canvas en la cual podemos organizar nuestros proyectos a través de su aplicación web de forma fácil e intuitiva desde cualquier equipo en el que introduzcamos nuestra cuenta ya que al estar en la nube no se pierde nuestra información ni aunque se degrade el sistema.

Ventajas:

- Es muy rápido su aprendizaje y su uso así como su simplicidad.
- La hemos usado en el transcurso de la carrera por lo que ya estaríamos familiarizados con el entorno.

Desventajas:

- No esta integrado dentro de nuestro repositorio por lo que lo tendríamos que usar como una herramienta mas en la que al final duplicaríamos trabajo.

Version One

Es un gestor de tareas online en el cual podemos gestionar todas las tareas de nuestros proyectos así como realizar un seguimiento de forma visual del estado del proyecto y de las características del mismo.

Ventajas:

- Podemos incluir código por lo que es mas completo que otras herramientas de gestión de tareas.

Desventajas:

- Como antes hemos indicado en este caso también seria algo que nos duplicaría trabajo al no estar integrado en nuestro repositorio seria también una herramienta a parte que no se comunicaría con nuestro repositorio.

zenhub

Es una herramienta similar a Trello que también es a modo de pizarra donde ver los cambios y el estado de un vistazo pero con algunas diferencias.

Ventajas:

- La ventaja principal es que podemos integrarlo desde GitHub por lo que ya no seria necesario duplicar trabajo con el uso de una aplicación externa.

Desventajas:

- No podremos añadir código pero tampoco es algo catastrófico ya que justo en el repositorio donde se integra la herramienta podemos visualizar dicho código.
- Por este motivo he decidido usar esta herramienta.

1.2. Gestores de versiones:

GitHub

Es un repositorio de versiones donde el código queda organizado por tareas(issues)y las versiones cada vez que hacemos un commit se actualiza las clases de código mostrando lo que ha cambiado. El software esta escrito en Ruby usando el framework Ruby on Rails.

Ventajas:

- Es de los repositorios mas usados y esta basado en git.

- El código es publico y cualquiera que le interese te puede proponer cambios en el mismo seguirte y ver el proyecto.
- Las distintas versiones del código están en la nube por lo que si se nos borra el contenido del disco duro aun así podremos recuperar lo.

Desventajas:

- También puedes tener proyectos privados pero al no influir sobre este proyecto no pasa nada.
- Hemos decidido usar este repositorio al ser uno de los mas utilizados y guardar mucha similitud con sus competidores por lo que sabiendo usar este podriamos usar los demas sin demasiados problemas.

Bitbucket

Este servicio es muy similar al anterior tambien esta basado en Git y ademas en Mercurial. Este repositorio web tambien guarda nuestro codigo y nuestras iteraciones sobre el proyecto para asi tener una vision mas completa sobre nuestro trabajo. Este software esta escrito en Python.

Ventajas:

- Es un repositorio muy usado y que esta basado en git que es casi la referencia en este tipo de proyectos.
- Tiene cuentas gratuitas para proyectos privados y publicos.

Desventajas:

- No se puede incluir mas de 5 personas en los proyectos gratuitos.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

Conceptos teóricos

3.1. Espacios de color

El color que percibimos en los objetos que nos rodean depende de la radiación reflejada en ellos.Según los estudios nosotros como humanos tenemos un rango "de luz visible" ese rango son en verdad tres frecuencias diferentes dentro del rango 769THz a 384THz.
Por lo que en verdad una imagen que percibimos es la unión de las tres frecuencias diferentes y para poder simular este hecho las maquinas simulan esta capacidad innata de los humanos creando los espacios de color que son modelos matemáticos para representar en una maquina lo que veríamos.3.1

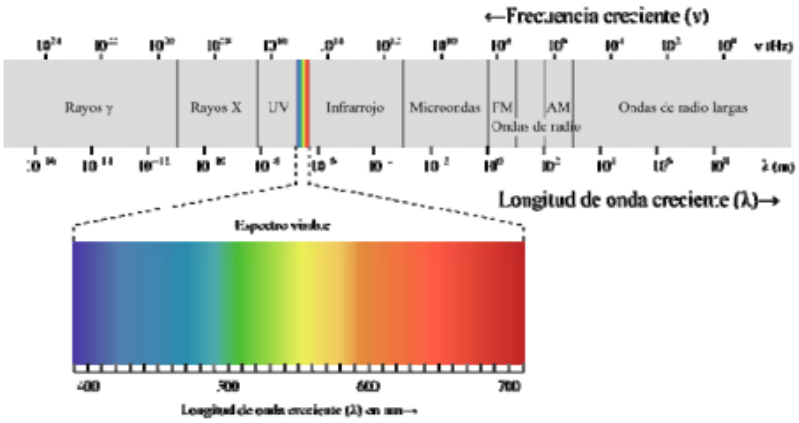


Figura 3.1: Frecuencias de luz visible [4]

RGB

El modelo RGB es usado por todos los sistemas digitales para la representación y captura de imágenes.

Se divide en tres canales:3.2

- R: canal del rojo (RED) contiene la intensidad de rojo de cada pixel
- G: canal del verde (GREEN)contiene la intensidad de verde de cada pixel
- B: canal del azul (BLUE)contiene la intensidad de azul de cada pixel

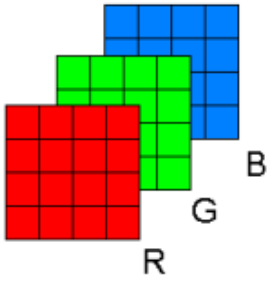


Figura 3.2: Los tres canales del espacio RGB [4]

La combinación de estos colores crea todas la gama de colores representable. El valor de la intensidad de cada canal depende de la codificación usada para su representación (8bits dan 16millones de colores)3.3

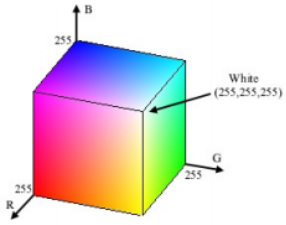


Figura 3.3: Representación del modelo RGB[4]

HSV

El modelo HSV [6] esta orientado a la descripción de los colores en términos mas prácticos para el ser humano que el RGB.
Ya que los canales significan algo por si solos no solo la cantidad de un color que no nos informa de mucho.3.4

- H: (Matiz) que representa el tono o color.
- S: (Saturación) representa el nivel de saturación de un color.
- V: (Brillo) representa la intensidad lumínica.

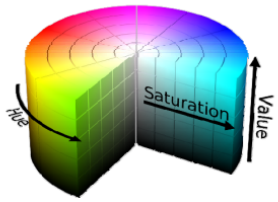


Figura 3.4: Representación del modelo HSV[4]

Una ventaja con otros espacios de color parecidos es que este permite representar todas las combinaciones del espacio RGB.

3.2. Transformada de hough

Introducción

Uno de los puntos relevantes del proyecto es la detección de las lineas pintadas o detectadas por el algoritmo (modo automático) para ello vamos a usar una técnica que sirve para detectar formas dentro de imágenes en visión artificial siempre que puedan ser expresadas de forma matemática.

Esta tecnica fue inventada por Richard Duda y Peter Hart en 1972 pero 10 años antes Paul Hough propuso y patento [2] la idea inicial de detectar lineas en la imagen. Mas tarde se generalizo para detectar cualquier figura.

Teoría

Normalmente para detectar figuras sencillas en una imagen primero hay que usar algún algoritmo de detección de bordes o una binarización de la imagen, quedándonos con la región de interés apropiada (los pixeles que forman las rectas) pero normalmente faltan pixeles por el ruido en la imagen.

Para ello el metodo de hough propone solucionar el problema detectando grupos de puntos que forman los bordes de la misma figura y asi conseguir unirlos creando la recta real a la que pertenecen.

Psudocódigo Transformada de Hough

7

Input: Imagen
Output: (list) de segmentos encontrados
foreach *punto en la imagen* **do**
  if *punto* ***(x,y)*** *esta en un borde:* **then**
    foreach *angulo en angulos* Θ **do**
      Calcular ρ para el punto (x,y) con angulo Θ
      Incrementar la posicion (ρ , Θ) en el acumulador
Buscar las posiciones con mayores valores en el acumulador
return *Lasrectascuyosvaloressonlosmayoresenelacumulador*

Limitaciones

Para que este proceso sea exitoso los bordes del objeto deben ser detectados bien con un buen pre-procesado de la imagen y aparecer claramente las nubes de puntos que forman las rectas.3.5

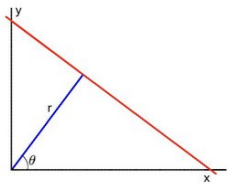
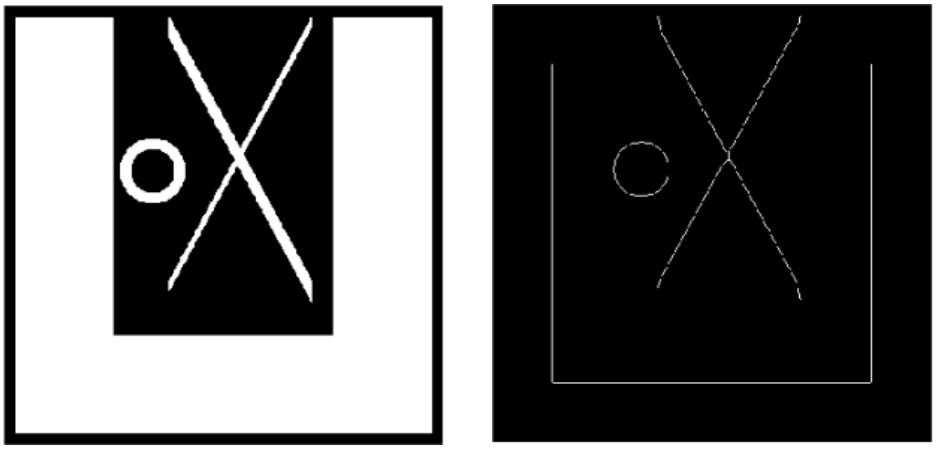


Figura 3.5: Lineas de hough[1]

Transformada probabilística de Hough

[3]Es una version que se basa en que la detección de bordes o la producción de la imagen binaria que contiene el objeto, podría tener ruido y por lo tanto los pixeles que corresponden al ruido con la transformada normal podrían ser considerados como una recta, cuando en verdad es ruido.
Para que unos puntos sean considerados recta en la probabilística el numero necesario de ellos es menor que en la normal. Pero penalizando a los puntos que se consideran la nube de puntos aleatoria a aquellos que estén en duda de donde se colocan.
Un exceso de ruido en la imagen también haría este método inservible pero para pequeñas cantidades lo hacen mas preciso que el método normal. Otra ventaja es que con este método obtenemos el segmento que necesitamos no la prolongación de el hasta el infinito.



(a) Original (b) Skeletonizada

Figura 3.6: Ejemplo de eskeletonize.

3.3. Skeletonize

[7] Dentro del pre-procesado de la imagen uno de los puntos clave para que nuestro método funcione es que después de su binarización y la detección de los bordes de la imagen a procesar debemos reducir la región sobre la que aplicar la transformada de hough para que esta sea mas rápida y detecte menos número de lineas imaginarias por cada linea real.

Esto lo conseguiremos usando una función de esqueletonizado que nos devuelve lo que su nombre indica el esqueleto de los bordes de la imagen reducidos a 1 pixel.3.6

3.4. Grafos

Introducción

K-componentes

Técnicas y herramientas

4.1. Interfaz gráfica de usuario:

Tkinter

Es una librería que proporciona el diseño y visualización de interfaces de usuario en Python esta a su vez esta basada en librerías de TK/TCL que están incluidas por en la propia instalación.

Ventajas:

- Es fácil de usar y es recomendable para el aprendizaje del lenguaje.
- Viene preinstalado con la distribución por lo que su uso es inmediato
- Al servir para aprendices y venir preinstalado podemos encontrar multitud de tutoriales y de documentación sobre ello.

Desventajas:

- Pocos elementos gráficos., escaso control de las ventanas y bastante lento.

WxPython

Es una librería basada en otra importante que veremos mas adelante , también es multiplataforma y esta programada en C/C++, es mas nueva que Tkinter.

Ventajas:

- Es más difícil de usar que Tkinter pero aun así hay mucha documentación sobre ella.
- Dispone de gran cantidad de elementos gráficos por lo que es bastante potente aunque con alguna limitación respecto a otras.
- Permite hacer una barrera o separación entre el código Python y lo que es la interfaz.
- Cuenta con una gran comunidad de gente que lo usa y postea ejemplos y tutoriales.

Desventajas:

- La principal desventaja es que se actualizan las versiones mucho y para mantener una aplicación durante largo tiempo perdemos tiempo de.
- Es más complejo de usar que el anterior.

Pyqt

Es más difícil de usar que Wxpython y wxwidget pero da más control sobre los elementos gráficos y muchas librerías se basan en ello por lo que se puede encontrar bastante información sobre esta librería.

Ventajas:

- Al ser tan usada si instalamos Python desde anaconda que es un cojunto de librerías y aplicaciones de Python ya tendríamos pyqt4 por defecto instalado.

Desventajas:

- Es mas difícil de entender y comprender que los anteriores pero queda mas limpias las interfaces.
- Si somos puristas e instalamos Python solo sin ide ni nada no vendría instalado pero si usamos anaconda si que vendría instalado

WxWidgets:

Como hemos mencionado anteriormente es muy parecido a wxpython por lo que no vamos a entrar en detalle también está programado en C/C++ y es multiplataforma da un aspecto de comportamiento nativo.

Aspectos relevantes del desarrollo del proyecto

5.1. Entorno de desarrollo

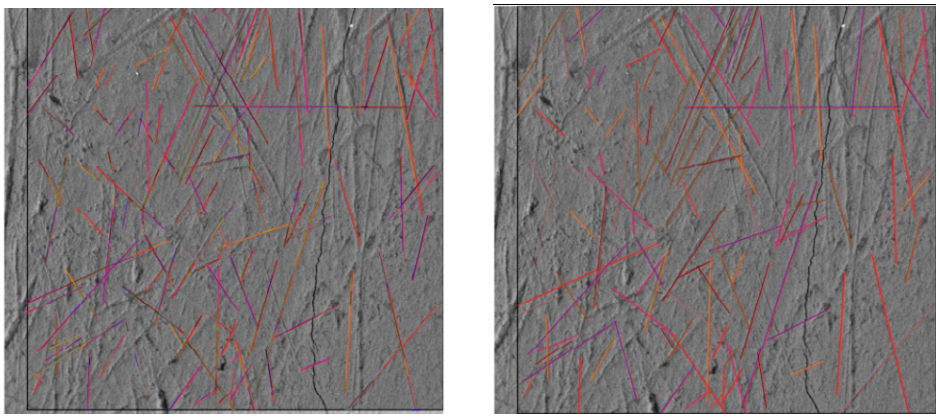
Como entorno de desarrollo de los prototipos hemos designado Jupyter ya que en sus notebooks interactivos puedes ejecutar directamente código python como si fuese un interprete.

Ventajas

- He podido añadir widguets para calibrar en buen grado las funciones que hemos utilizado.
  Gracias a estos widguets podemos dar valores e ir viendo como cambia la salida de la función de forma interactiva.5.7

Figura 5.7: Ejemplo de un widguet sobre la función de hough

- Su rápida visualización sin tener grandes conocimientos de interfaz gráfica ha sido un gran apoyo para poder visualizar desde el principio las imágenes procesadas y como quedaban.5.8
- Desde el propio entorno puedes ejecutar no solo código estructurado en script sino también código estructurado en clases y llamadas a métodos es como un IDE pero con limitaciones.5.9



(a) Lineas antes de unir (b) Lineas después de unir

Figura 5.8: Ejemplo de una visualización del resultado intermedio de las funciones.

```
In [32]: img=leerImagen()
         distance_red=distanciaAlRojo(img)
         imgBin=binarizar(distance_red)
         imgBinCrop,imgCrop=cropImg(imgBin,img)
         sinRuido=reducirGrosor(imgBinCrop)
         lines=proHough(10,5,11,sinRuido)
         G=nx.Graph()
         G=combina(8,4,lines,G)
         k_components = apxa.k_components(G)
         segmentosDeVerdad=segmentosVerdad(k_components,lines)
         segmentosDeVerdad

Out[32]: [((434, 12), (310, 323)),
          ((485, 467), (434, 643)),
          ((469, 251), (337, 15)),
          ((531, 293), (513, 79)),
          ((722, 182), (337, 184)),
          ((463, 340), (294, 19)),
```

Figura 5.9: Ejemplo de una Ejecución.

- Multitud de librerías y funciones que en entornos parecidos como matlab serian de pago y aquí al ser software libre el ejemplo anterior lo resume en una librería numpy [5].

5.2. Procesado imagen

Para llegar a conseguir calcular las lineas que había pintadas en las imágenes tube que realizar una serie de pasos que vamos a resumir en tres etapas.

Binarización

Partiendo de una imagen que solo tenia lineas en rojo pintadas encima de las estrías producidas por el desgaste y lo demás de la imagen en escala de grises, lo primero fue leer la imagen a trabes de las funciones ya programadas

en la librería de Scikit-Image(skimage).
Una vez que tenemos la imagen guardada en el espacio de color RGB podemos empezar el procesado quedándonos con el canal Rojo.
Calculamos la distancia de cada pixel de la imagen al color rojo restando, uno menos el valor absoluto del pixel en el canal S (saturación) del espacio de color HSV , (restando el valor absoluto a la unidad conseguimos normalizar entre [0-1])y pasamos la imagen de distancias a blanco y negro y así tendremos un valor entre 0 y 256 en cada pixel correspondiente a la distancia al rojo cuanto mas alejado mas negro y los que sean rojos en blanco.
Para que la diferencia sea blanco o negro binarizamos la imagen con un valor umbral calculado como threshold otsu y así la imagen los valores de la distancia que sean mayores que el umbral pasaran a valer máximo y los que no consigan pasar el umbral serán los bordes(blanco).5.10

Obtener segmentos

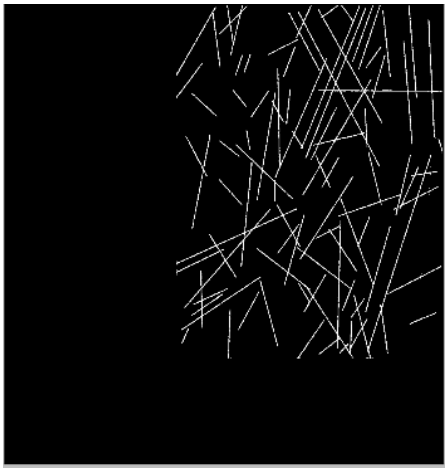
Partimos de la imagen binarizada y lo primero es reducir el grosor de las lineas detectadas a un pixel eso lo conseguimos llamando a la función skeletonize que nos devuelve la imagen con las lineas de un pixel (así no acumulamos errores y es mas rápida la búsqueda de rectas).
Seguidamente llamamos a la función "probabilistic hough line"que nos va a encontrar segmentos que formaran las lineas el funcionamiento ha sido explicado en el apartado (conceptos teóricos).5.11

Procesado de segmentos

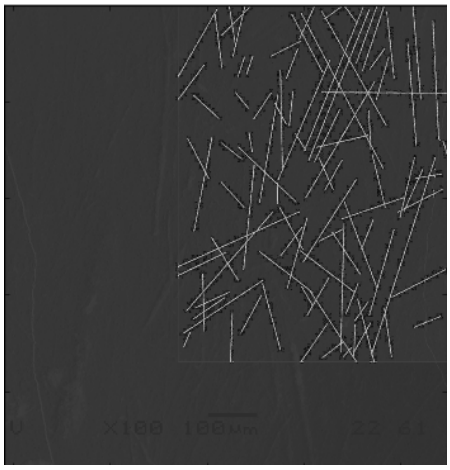
Llegados a este punto lo que tenemos son muchos segmentos que forman las lineas reales y tenemos que unirlos.
Para ello vamos a usar la teoría de grafos añadiendo los segmentos a un grafo. Para unir dos segmentos tiene que cumplirse que la distancia mínima entre sus extremos sea menor que un umbral y si pasa este punto comprobaremos que el angulo que forman entre ellas sea menor a otro umbral y si cumplen las dos condiciones añadiremos un camino al grafo desde la recta uno a la recta dos.5.12

Recuperación de lineas

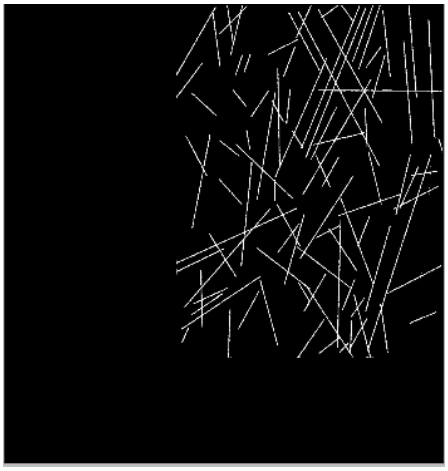
Ahora lo que tenemos es un grafo con clusters ya que cada cluster se identifica con únicamente una recta y tendremos tantos como rectas. Un problema de grafos es el problema de las k-componentes pero a nosotros solo nos interesan las 1-componentes del grafo ya que cada grupo de estos segmentos çercanos"se corresponde con una recta real. devolvemos la combinación de los segmentos mas relevantes de cada cluster y estos se convierten en nuestra buscada linea real.5.13



(a) Original



(b) Distancia al rojo

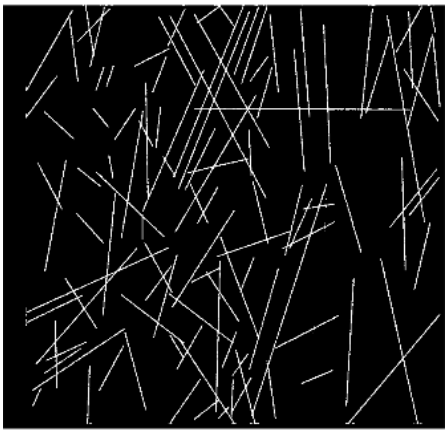


(c) Imagen binarizada

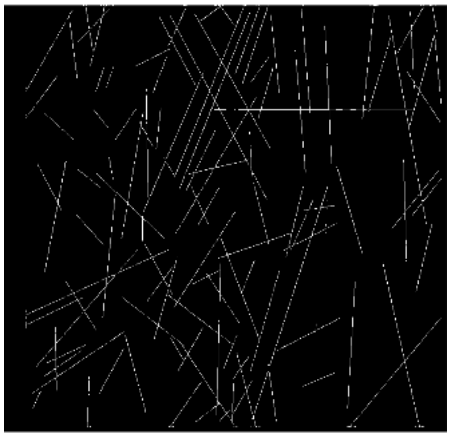
Figura 5.10: Resumen visual binarización.

Resumen pasos

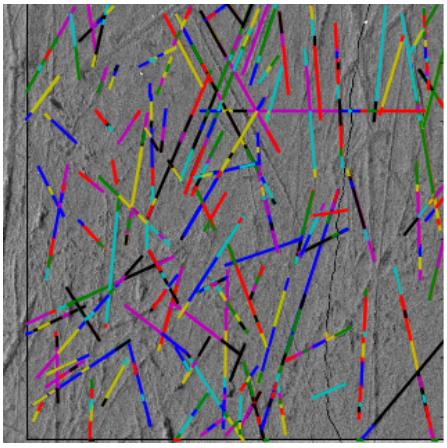
- Binarizar la imagen para solo quedarnos con los objetos de interés
- Obtener segmentos que forman trozos de las lineas.
- Añadir caminos entre las lineas cercanas en un grafo
- obtener los grupos de lineas próximas y devolver la recta que las una.



(a) Binarizada

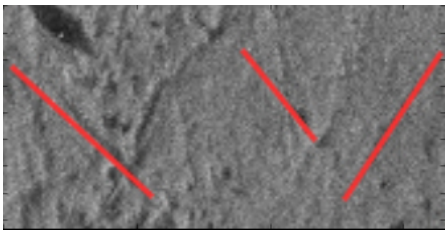


(b) Lineas a 1 pixel

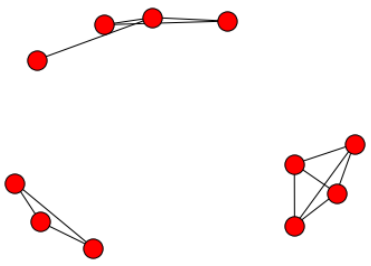


(c) Imagen original con segmentos

Figura 5.11: Resumen visual Obtener Segmentos.



(a) Imagen con segmentos en rojo.



(b) Grafo de clusters de segmentos.

Figura 5.12: Resumen visual procesado de Segmentos.

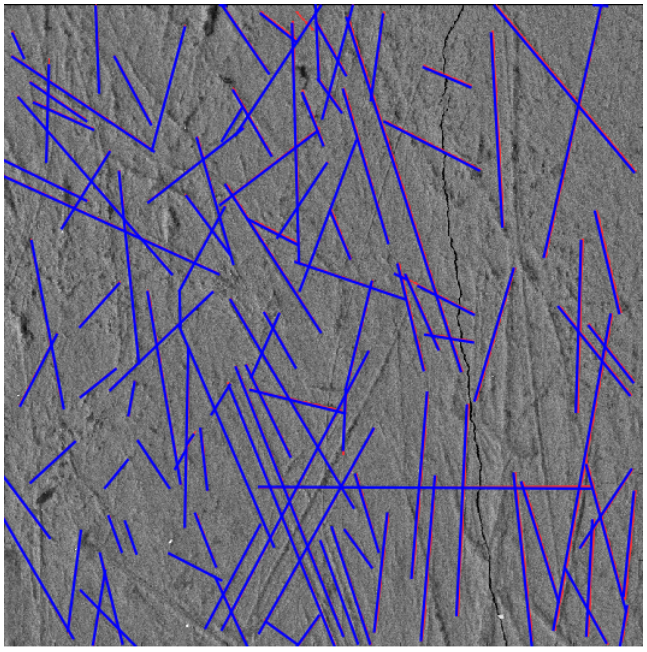


Figura 5.13: Lineas obtenidas después de procesar el grafo

5.3. Interfaz

Primera versión

Para el desarrollo de la interfaz gráfica pensé en una una planificación espacial acorde con los elementos que esta contendría por lo que un layout que seria muy adecuado podría ser un border layout pero como no existe en Pyqt4 lo he tenido que simular gracias a apilar layouts de otros tipos y consiguiendo tener una botonería arriba del todo y dos columnas debajo claramente diferenciadas en la que en una estuviera la imagen que vamos a mostrar y en la otra las funcionalidades.5.14

Versión a evaluar

En otro Sprint del proyecto lo que he usado han sido pestañas para así tener en la zona de funcionalidades los modos de trabajo de la aplicación claramente separados.5.15

- El modo uno.
- El modo dos.
- El modo tres.

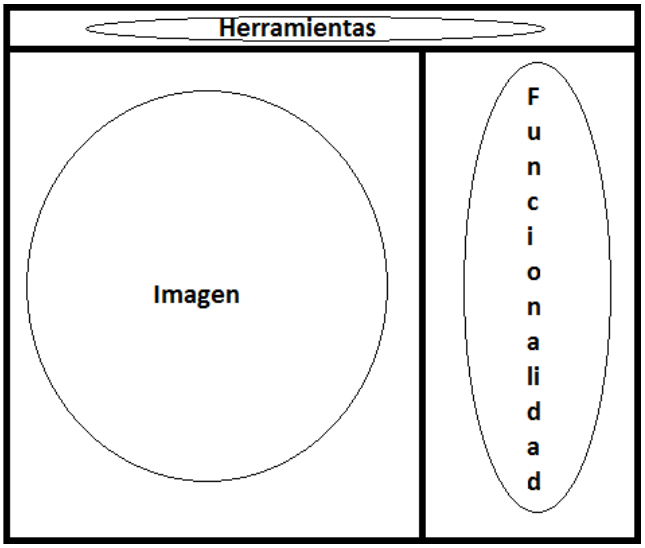


Figura 5.14: Diseño de la interfaz de usuario

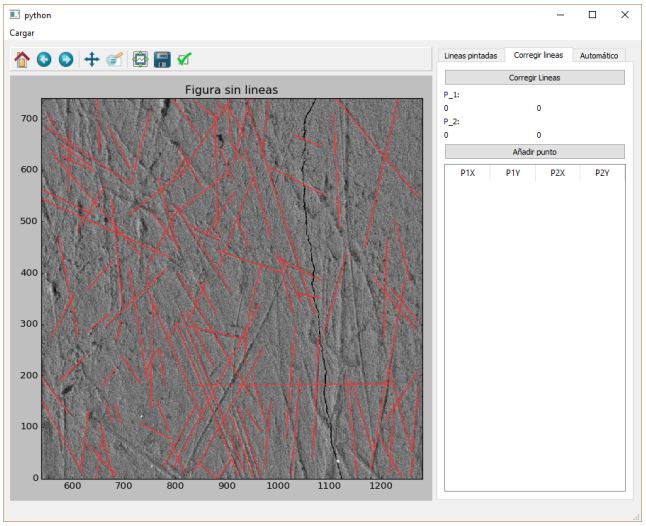


Figura 5.15: Diseño de la interfaz de usuario

Barra de herramientas

Es una sección de la interfaz que nos va permitir de momento la carga de imágenes para su procesado (También funciona con Ctrl+O).5.16

Barra de herramientas de la imagen

Es una sección de la interfaz que nos permitirá manipular la imagen para realizar estas acciones:5.17

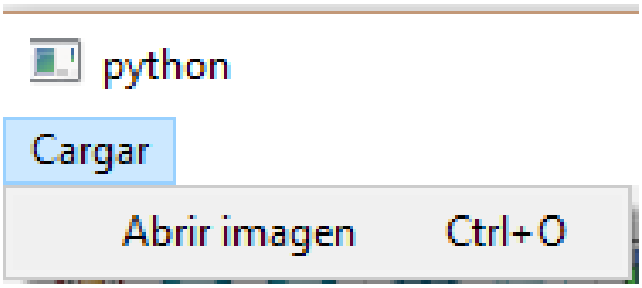


Figura 5.16: Barra de herramientas de la aplicación

- Volver al principio.
- Retroceder un paso.
- Avanzar un paso.
- Desplazar la imagen.
- Aumentar la región que seleccionemos.
- Configurar la región, bordes, etc.
- Guardar la imagen con su escala.
- configurar el tamaño y numeración de los ejes.
- Coordenadas actuales del ratón.
- Niveles de color de los canales R G B del espacio RGB.

Figura 5.17: Barra de herramientas de la imagen

FigureCanvas de la imagen

Es la región donde se muestra la imagen que va a ser ligeramente mas grande que la parte de las pestañas.5.18

Pestañas

Sección de la imagen donde estarán implementadas las funcionalidades para visualizar las lineas que son detectadas por el algoritmo en la imagen.5.19

- El modo primero es la detección de los surcos en rojo en los dientes

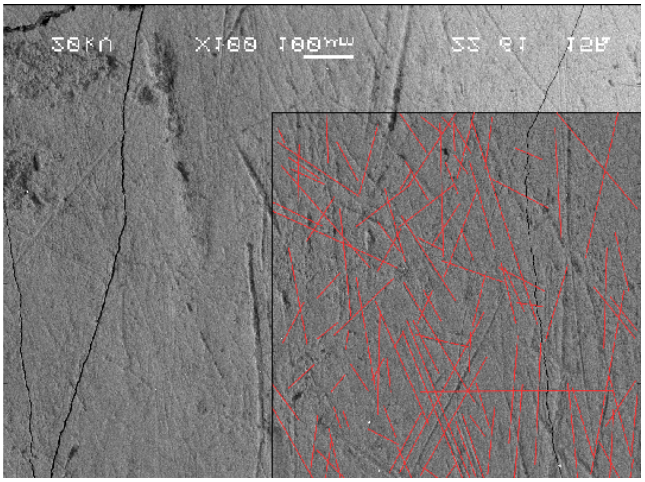


Figura 5.18: FigureCanvas de la imagen

- El modo segundo es la corrección/detección manual de las lineas por una persona y quedando reflejadas en la imagen.
- El modo tercero es la ultima parte del proyecto que consistirá en hacer todo el proceso anterior de forma automática.

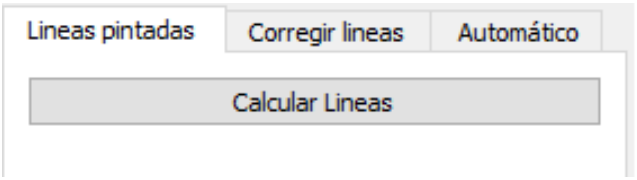


Figura 5.19: Pestañas

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] G. Bradski. Transformada de hough, 2000. [imagen de la documentacion de OpenCv].

[2] Paul Hough. Patente de paul hough line transform, 1962. [Internet; descargado 26-septiembre-2016].

[3] Nahum Kiryati, Heikki Kälviäinen, and Satu Alaoutinen. Randomized or probabilistic hough transform: unified performance evaluation. *Pattern Recognition Letters*, 21(13–14):1157 – 1164, 2000. Selected Papers from The 11th Scandinavian Conference on Image.

[4] Cesar Represa. Manual de hardware de aplicacion especifica, 2015. [imágenes del manual].

[5] Scypi.org. Biblioteca de funciones python, 2013. [Biblioteca de funciones].

[6] Alvy Ray Smith. Modelo hsv, 1978. [HSV es lo mismo que HSB pero no que HSB NI HSL].

[7] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python, 2014. [Internet; descargado 26-septiembre-2016].