



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

**Sistema de reconocimiento
automático en arqueobotánica**



Presentado por Jaime Sagüillo Revilla
en Universidad de Burgos — 15 de febrero de 2017
Tutores: Álgvar Arnaiz González, José Francisco Díez
Pastor y Virginia Ahedo García



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



Álvar Arnaiz González, José Francisco Díez Pastor profesores del departamento de Departamento de Ingeniería Civil, Área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Jaime Sagüillo Revilla, con DNI 12782524-K, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Sistema de reconocimiento automático en arqueobotánica.

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, 15 de febrero de 2017

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

Descriptores

Arqueobotánica, Fitolito, Reconocimiento de imagenes, Python

Abstract**Keywords**

Índice general

Índice general	III
Índice de figuras	IV
Índice de tablas	V
Introducción	1
Objetivos del proyecto	2
Conceptos teóricos	3
3.1. Segmentación	3
3.2. Binarización	3
3.3. Thresholding	4
Técnicas y herramientas	5
Aspectos relevantes del desarrollo del proyecto	6
5.1. Procesamiento de imágenes	6
Trabajos relacionados	9
Conclusiones y Líneas de trabajo futuras	10
Bibliografía	11

Índice de figuras

Índice de tablas

Introducción

Objetivos del proyecto

Conceptos teóricos

Para la comprensión de este proyecto será necesario la comprensión de algunos conceptos teóricos que introduciré en este apartado:

- Segmentación
- Binarización
- *thresholding*

3.1. Segmentación

La segmentación en el campo de la visión artificial, como se indica en la wikipedia, consiste en subdividir una imagen en varios pixeles u objetos. [2] Cuando segmentamos una imagen lo que pretendemos hacer es cambiar su representación para poder obtener de esta una mayor utilidad o cantidad de información.

En nuestro caso segmentamos la imagen para eliminar el fondo de ella y obtener así una imagen con solo su parte delantera. De esta manera eliminamos el ruido que existe en la imagen y a su vez la simplificamos reteniendo la parte de la imagen en la que se encuentran los objetos que nos interesan.

Posteriormente a ese paso nos interesa, como es obvio, dividir la parte delantera de la imagen resultante en objetos. Para de esta manera obtener cada uno de los objetos por separado de forma ideal.

3.2. Binarización

La binarización de una imagen consiste en la simplificación de los valores de cada pixel a 2 posibles valores, blanco o negro, representando el fondo y el frente de la imagen cada uno de ellos. De esta manera conseguimos una

simplificación muy significativa. Para ejemplificarlo de manera más visual estamos convirtiendo una imagen RGB en la que cada pixel tiene 3 valores y cada uno de estos valores puede variar desde el 0 al 255, lo cual es una explosión de combinaciones muy significativa, frente a una imagen binarizada en la que cada pixel tiene 2 posibles valores, 0 o 1. Como podemos imaginar una simplificación de la imagen de tal calibre facilita el procesamiento de la imagen enormemente.

3.3. Thresholding

Es el método mas simple para la segmentación de una imagen, pudiendose utilizar para la binarización de una imagen, como es nuestro caso. Consiste en reemplazar los píxeles por debajo de una determinada constante a píxeles negros y los que se encuentran por encima a píxeles blancos o viceversa.

Existen distintas maneras de llevar a cabo este proceso, siendo uno de lo más conocidos el método de Otsu.[\[1\]](#)

Técnicas y herramientas

Aspectos relevantes del desarrollo del proyecto

5.1. Procesamiento de imagenes

Como primera aproximación al problema que nos concierne hemos escogido el procesamiento de imagenes mediante la librería Scikit-image para Python. Mediante esta herramienta trataremos de dar solución a nuestro problema siguiendo los siguientes pasos:

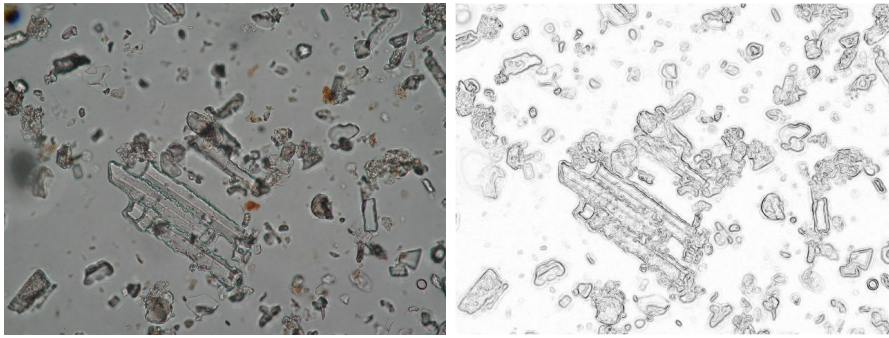
1. Convertimos la imagen a escala de grises
2. Segmentamos los objetos del fondo de la imagen
3. Obtenemos los distintos objetos de la imagen

Convertimos la imagen a escala de grises

La conversión de la imagen original (RGB) a escala de grises viene motivada porque para poder segmentar los objetos del fondo de la imagen mediante el método de *Thresholding* solo se puede partir de una imagen en escala de grises.

Segmentamos los objetos del fondo de la imagen

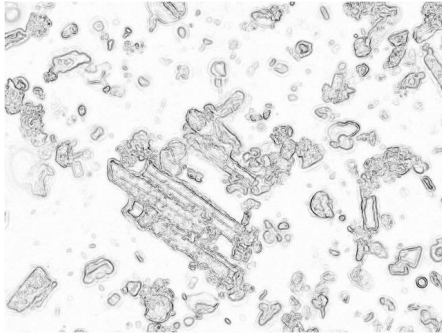
Una vez tenemos la imagen en escala de grises procedemos a transformar nuestra imagen en una imagen en blanco y negro o binarizada. Los motivos por los que binarizamos la imagen es para obtener una imagen que sea más significativa para nosotros y además este simplificada, para facilitarnos su procesamiento.



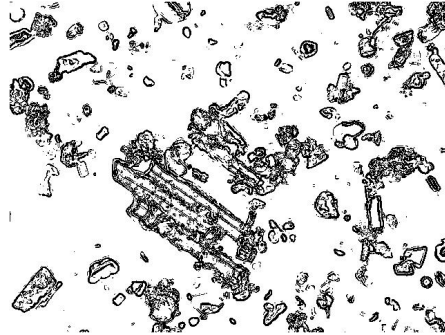
(a) Original

(b) Imagen en escala de grises

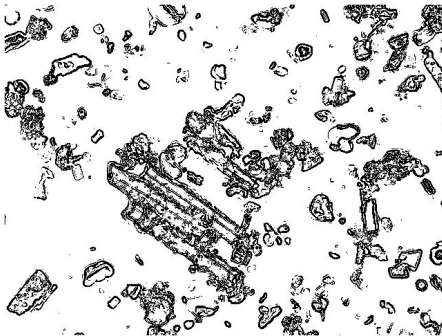
Scikit nos proporciona distintos métodos mediante los cuales podemos segmentar una imagen. A continuación vamos a ver el resultado aplicando distintos métodos, los cuales se van indicando en cada una de las figuras:



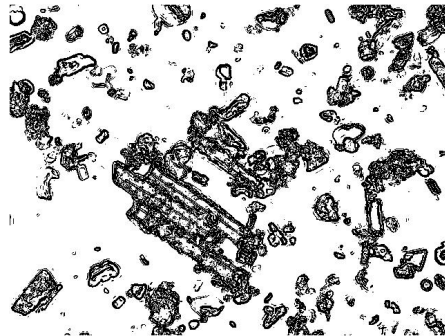
(a) Imagen en escala de grises



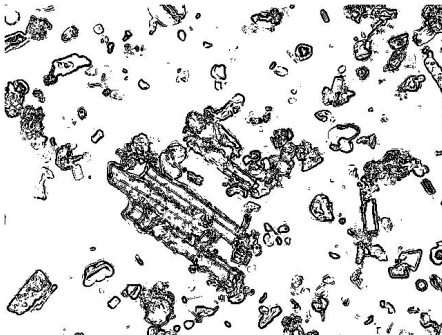
(b) Método de Otsu



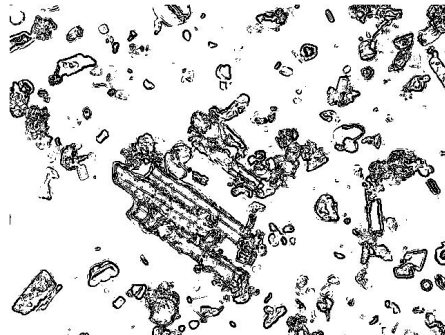
(c) Método de Otsu



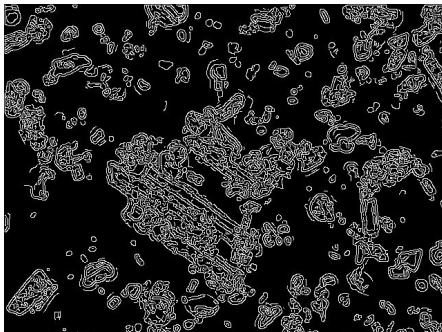
(d) Método de Yen



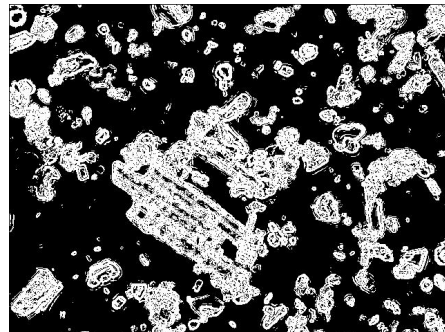
(e) Método de Li



(f) Método de ISODATA



(g) Método basado en bordes



(h) Método adaptativo

Trabajos relacionados

Conclusiones y Líneas de trabajo futuras

Bibliografía

- [1] Otsu's method, February 2017.
- [2] Wikipedia. Segmentación (procesamiento de imágenes) - Wikipedia, la enciclopedia libre.