



UNIVERSIDAD
POLITECNICA
DE VALENCIA

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE
TELECOMUNICACIONES

Procesado Digital de imagen para la estimación del
área ocupada por microorganismos sobre probetas
de cemento y yeso.

Autor

Jose Sánchez García

Directores

Ignacio Bosch Roig

Pilar Bosch Roig



Escuela Técnica Superior de Ingenieros de
Telecomunicación

—
Valencia, septiembre de 2016

AGRADECIMIENTOS

Quisiera agradecer a varias personas la ayuda y el apoyo que me han prestado en la realización del Trabajo Fin de Grado que aquí se presenta.

A Ignacio Bosh Roig por prestarse a ser mi tutor en este trabajo y sobre todo por lograr despertar en mí la motivación y las ganas de superación que han hecho posible este proyecto.

A mis compañeros de trabajo Gonzalo, Santi, Jorge, Carles, Guille y Alicia, por toda la ayuda prestada a lo largo de los días que estuve en el grupo de trabajo y por hacerme sentir como en casa.

A mis compañeros del Colegio Mayor, con los que empecé esta etapa e hicieron ese primer año tan especial.

A mis profesores de instituto: a Elena por comportarse todos estos años, desde que puse un pie en su clase hasta hoy día, como una madre y por hacer que todo parezca mucho más sencillo. A Rafa, por toda la caña que metió en ese último curso y que tanto he valorado a lo largo de los años estudiando esta carrera. Y sobre todo a Juan, sin duda el gran motivador de la persona en la que me he convertido hoy en día, tanto a nivel personal como académico. La docencia necesita más personas como ellos, así que gracias de nuevo.

Y por supuesto a mi familia: a mis padres, ellos son la gran causa del lugar que ocupo hoy en día y de todo lo que he conseguido; gracias por haberme hecho llegar a donde estoy. A mis hermanos por todo el apoyo recibido desde lejos y que en muchos momentos han conseguido animarme. A mi abuela por acordarse siempre de mí, por llamarme antes de los exámenes y por enviarme la comida que, aquí, tanto echo de menos. Y, por supuesto, a la persona que más cerca he tenido estos años y que me ha dado fuerzas todos los días para conseguir lo que me proponía, Paula; gracias por esas noches interminables de estudio, por aguantarme cuando las cosas no salían como esperaba y por ayudarme a tomar las decisiones correctas.

Todos ellos son los que, verdaderamente, ponen nombre a este Trabajo Final de Grado que resume mis cuatro años de aprendizaje.

Gracias por todo y a todos.

CONTENIDO

1. INTRODUCCIÓN	5
2. OBJETIVOS	6
3. PLAN DE TRABAJO	7
4. MATERIALES Y MÉTODO	8
ANÁLISIS FOTOGRÁFICO DEL CRECIMIENTO MICROBIOLÓGICO EN LA SUPERFICIE	9
METODOLOGÍA UTILIZADA	10
Organización de datos.....	11
Binarización de imágenes.....	12
Segmentación de probetas	13
Co-registro automático:	18
Análisis de ocupación del microorganismo	36
Diseño de la interfaz de usuario.....	47
5. CONCLUSIONES.....	61
6. LINEAS FUTURAS	62
7. REFERENCIAS	63

1. INTRODUCCIÓN

En el presente proyecto se pretende estimar el área ocupada por microorganismos (bacterias) mediante procesado digital de imágenes (procesamiento morfológico y colorimétrico), para la evaluación de la eficacia microbiológica del biocida BIOTIN R encapsulado con nanopartículas y aplicado sobre probetas de cemento y yeso [1].

El estudio se realizó en dos procesos: la inoculación microbiológica de todas las probetas y el control fotográfico semanal de las probetas a lo largo de todo el tiempo de duración de los ensayos de laboratorio, en el laboratorio de Microbiología Ambientale e Biorestauro (MicroLAB) del Dept. di Bioscienze e Territorio (DiBT) de la Università degli Studi del Molise, sede de Pesche (Isernia) Italia. Por lo que para la estimación del área ocupada, se dispone de diferentes imágenes fotográficas a lo largo del tiempo con distintas características de tamaño, forma, etc...

El proyecto se plantea en dos fases:

- Fase 1, adecuación de las imágenes para su correcto procesamiento posterior: Considerando procesamiento morfológico para la detección de la ubicación de las probetas dentro de las diferentes imágenes tomadas a lo largo del ensayo.
- Fase 2, detección de las áreas de los microorganismos en las diferentes etapas temporales del experimento. En el caso de Bacterias, la evolución temporal implica un amarilleamiento de la superficie de la probeta.

2. OBJETIVOS

El principal objetivo de este Trabajo Final de Grado es desarrollar una metodología que permita realizar un análisis cuantitativo de la evolución del color en las probetas de cemento, para así poder concluir de forma empírica la eficacia del biocida Biotín R. Para ello, se desarrolla una herramienta capaz de procesar los datos de partida (imágenes tomadas en el control semanal de los ensayos) para organizarlos y corregirlos de forma simétrica, y que permita calcular el tanto por ciento de área ocupada a lo largo de las diferentes tomas. El entorno de programación elegido es "Matlab".

Por tanto, los objetivos a cumplir por este proyecto son los siguientes:

- Realizar un estudio de los principales algoritmos y métodos de procesado de imagen y así elaborar un primer esquema de trabajo.
- Realización de códigos individuales que ejecuten los diferentes pasos del esquema para poder estimar por separado la utilidad de los diferentes pasos que se habían establecido en un primer lugar.
- Estudio de los diferentes tipos de algoritmos de corrección y técnicas de transformaciones geométricas de las imágenes.
- Estudio de los diferentes modos de representación de imágenes.
- Diseño e implementación de una interfaz de usuario que permita organizar los datos de partida y realizar las transformaciones y análisis pertinentes, para así estimar el tanto por ciento de área ocupada por el microorganismo, de una forma clara y sencilla para el usuario final.

Además, la herramienta debe reunir ciertos requisitos para ser completamente funcional y útil.

- Los resultados obtenidos deben ser fiables para que la conclusión de efectividad del biocida se realice con la mayor confianza posible.
- La herramienta debe ser sencilla de utilizar para el usuario final, de manera que no le requiera tiempo familiarizarse con el entorno de trabajo.
- Los resultados deben mostrarse de manera clara para su rápida y correcta interpretación.
- En la medida de lo posible, se debe intentar realizar una aplicación lo más eficiente y rápida que permita el entorno "Matlab".

3. PLAN DE TRABAJO

El equipo compuesto por: Prof. Giancarlo Ranalli y la Dra. Pilar Bosch, trabajaron con 24 probetas, 12 en yeso y 12 en cemento. La composición de las probetas consistió en bloques de 4x4 cm de yeso o cemento con la adición de PC462 (1:4): matriz ibersil (sílice precipitada de 0.5 gr) con o sin biocida Biotin R (2 gr) [1].

El biocida Biotin R está constituido por dos moléculas: n-octil-isotiazonona (OIT) y iodopropinilbutilcarbamato (IPBC: sal de amonio cuaternario) disueltos en etanol debido a su baja solubilidad en agua. Estable en un amplio intervalo de Ph (4 A 11) y temperatura (-20°C a +80°C). Amplio espectro de actividad contra algas, hongos, bacterias (gram-positivas y gram-negativas), actinomicetos y líquenes. Se utiliza para exterior o interior en piedra o pintura mural, madera, etc.

Una vez recibidas las muestras en el laboratorio de Microbiologia Ambientale e Biorestauro (MicroLAB) del Dept. di Biocienze e Territorio(DiBT) de la Università degli Studi del Molise, sede de Pesche (Isernia), Italia, se procedió al estudio y evaluación de la eficacia inhibitoria del crecimiento microbiológico del biocida Biotin R encapsulado con nanopartículas.

Para llevar a cabo esta evaluación, las pruebas se realizaron en ambiente interior controlado, con la intención de acelerar tiempos y disminuir las variables que puedan afectar la eficacia del biocida.

4. MATERIALES Y MÉTODO

Se llevó a cabo un control fotográfico semanal de las probetas a lo largo de todo el tiempo de duración de los ensayos de laboratorio mediante una cámara fotográfica réflex NIKON D90. La cámara fue mantenida inmóvil durante todos los ensayos, fijándola con un trípode para mantener las condiciones fotográficas lo más estables posibles (Figura 1). Así mismo se mantuvieron estables las condiciones de las fotografías utilizando siempre la función manual pre-fijada con: resolución de imágenes: 300ppp, velocidad de obturación: 1/20 y apertura del diafragma: F=3.5. Además, se realizó un control de la temperatura y humedad durante todo el periodo de duración de los ensayos mediante la utilización de instrumentos portátiles que permiten el registro de las variaciones diarias (datalogger).



Figura 1. Toma de imágenes en condiciones estables

Las probetas de cemento y yeso con biocida (test) y sin biocida (control) fueron colocadas por triplicado en placas Petri de cristal de 150x15 mm conteniendo 70 ml de medio de cultivo "Potato Broth médium". Las probetas, sumergidas hasta la mitad de su altura, se dejaron durante 24 h en el medio de cultivo para que lo absorbieran hasta saturación.

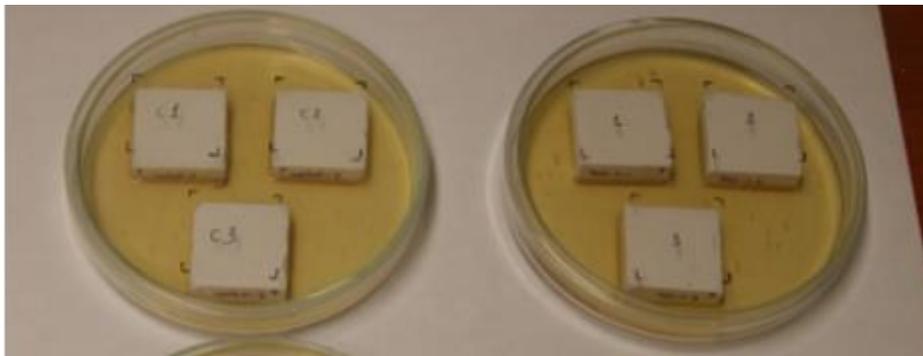


Figura 2. Estado de los ensayos tras la saturación

Tras el periodo de absorción, se procedió a inocular mediante pulverización la superficie de las probetas con esporas de un cultivo mixto de hongos, aislados de la naturaleza (obtenidos de una muestra de suelo). Los ensayos fueron mantenidos durante todo el tiempo a una temperatura de 23.1°C y una HR del 93.8%.



Figura 3. Inóculo de las esporas de cultivo mediante pulverización

El cultivo mixto inoculado estuvo formado por cinco especies fúngicas obtenidas de una muestra de suelo. Los inóculos utilizados fueron identificados mediante observación microscópica correspondiente a las especies *Aspergillus* sp., *Paecilomyces* sp. Y *Penicillium* sp.

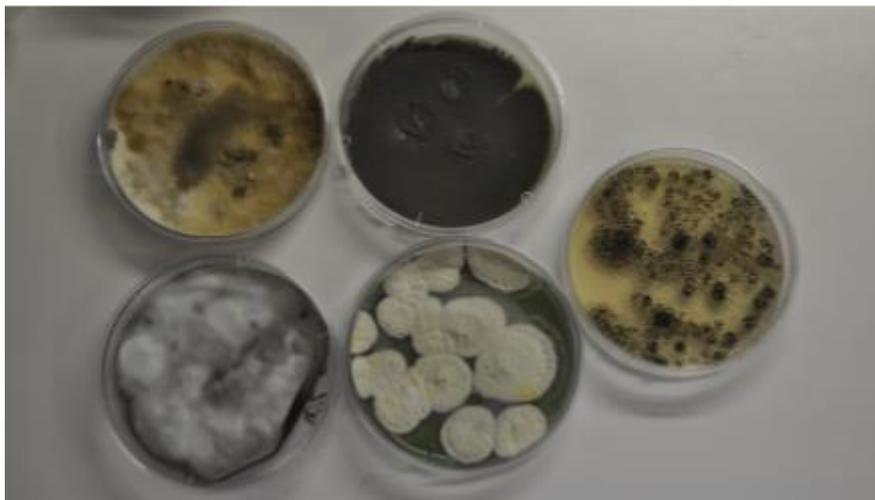


Figura 4. Especies fúngicas obtenidas de la muestra de suelo

ANÁLISIS FOTOGRÁFICO DEL CRECIMIENTO MICROBIOLÓGICO EN LA SUPERFICIE

Se observa un crecimiento microbiológico en las probetas de yeso, en las que el inicio del mismo se detecta a los 15 días del inicio del ensayo, crecimiento que se hace evidente sobre la superficie de las probetas a los 60 días del inicio del mismo. Como se observa en la figura, existe una

mayor cobertura por el biofilm de la superficie de las probetas control respecto a las probetas con el biocida. Esta diferencia es evidente y significativa solo en una de las tres réplicas, mientras que en las otras dos la diferencia con los controles es mucho menor y no significativa estadísticamente.

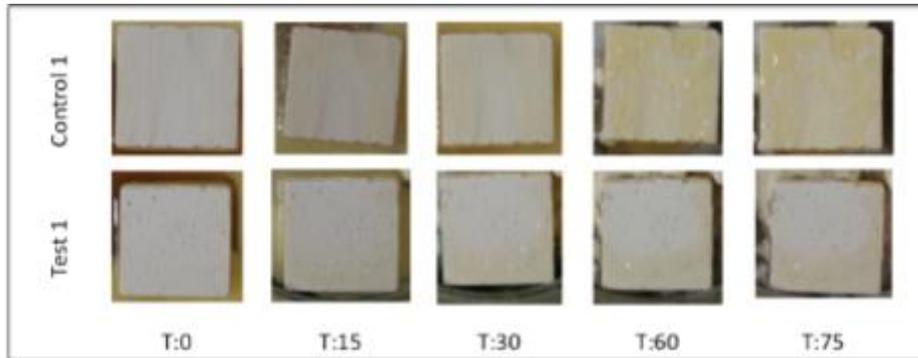


Figura 5. Progresión temporal del crecimiento del microorganismo en la probeta 1.

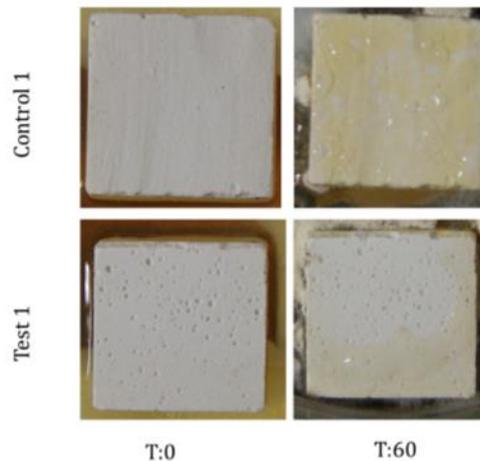


Figura 6. Toma en la que el efecto del biocida comienza a ser claramente visible

METODOLOGÍA UTILIZADA

Los pasos seguidos para el análisis de las probetas han fueron los siguientes:

- Organizar los resultados del control fotográfico para facilitar el procesado en “Matlab”.
- Binarizar las imágenes para poder procesarlas.
- Segmentar las probetas utilizando las operaciones morfológicas pertinentes, obteniendo las máscaras de cada una de ellas.
- Realizar el corregistro de cada una de las tomas temporales en función a la posición inicial

- de las probetas. Para intentar seguir y corregir el movimiento de las probetas.
- Una vez corregida la posición de todas las imágenes, almacenar únicamente los píxeles que componen cada una de las probetas en una estructura de resultados.
- Analizar y cuantificar el nivel de ocupación del microorganismo en función del color de los píxeles.
- Representar los resultados obtenidos.
- Generar una interfaz de usuario que facilite la tarea al usuario final del programa.

Organización de datos

El primer paso será preparar los resultados del control fotográfico semanal para facilitar el procesamiento en MATLAB. Teniendo en cuenta que se realizan dos fotografías por cada fecha de control, una por cada ensayo, se puede proceder a clasificar la información cogiendo los archivos pares por un lado (test) y los impares (control) por otro. Todo ello se guardará en una estructura llamada *file* que contiene varios campos:

controlname/testname: en el que se distingue si el archivo pertenece al ensayo de control o test.

controlname/testname: el nombre del archivo.

controldate/testdate: contiene la fecha del archivo, que se utilizará posteriormente para las gráficas temporales.

controlimage/testimage: contiene la matriz de 3 dimensiones.

Con el siguiente código se consigue ordenar la información de la forma deseada:

```

lee_archivos=dir('C:\Users\josan23\Desktop\Bacterias');
n=1;
m=1;
for k=3:length(lee_archivos)

    if rem(k,2)==0 %pares
        file(m).testname=lee_archivos(k).name;
        file(m).testdate=lee_archivos(k).date;

        archivo=lee_archivos(k).name;
        nombre='C:\Users\josan23\Desktop\Bacterias\';
        file(m).testimage=imread(strcat(nombre,archivo));
        m=m+1;

    else %impares
        file(n).controlname=lee_archivos(k).name;
        file(n).controldate=lee_archivos(k).date;
        archivo=lee_archivos(k).name;

        nombre='C:\Users\josan23\Desktop\Bacterias\';
        file(n).controlimage=imread(strcat(nombre,archivo));

        n=n+1;
    end
end

```

Figura 7. Código MATLAB para organizar la toma de datos en una estructura

Y los archivos quedan ordenados de la siguiente forma, de manera que con la posición 1 podemos acceder tanto a la primera toma del ensayo de test como a la de control:

Fields	controlname	controldate	controlimage	testname	testdate	testimage
1	'2014-07-03 Bacterias Control.JPG'	'03-jul.-2014 13:54:18'	3056x4592x3 uint8	'2014-07-03 Bacterias Test.JPG'	'03-jul.-2014 13:56:04'	3056x4592x3 uint8
2	'2014-07-04 Bacterias Control.JPG'	'04-jul.-2014 13:51:50'	3056x4592x3 uint8	'2014-07-04 Bacterias Test.JPG'	'04-jul.-2014 13:52:44'	3056x4592x3 uint8
3	'2014-07-08 Bacterias Control.JPG'	'08-jul.-2014 14:19:56'	3056x4592x3 uint8	'2014-07-08 Bacterias Test.JPG'	'08-jul.-2014 14:20:58'	3056x4592x3 uint8
4	'2014-07-10 Bacterias Control.JPG'	'09-jul.-2014 17:22:50'	3056x4592x3 uint8	'2014-07-10 Bacterias Test.JPG'	'09-jul.-2014 17:23:20'	3056x4592x3 uint8
5	'2014-07-12 Bacterias Control.JPG'	'11-jul.-2014 19:27:48'	3056x4592x3 uint8	'2014-07-12 Bacterias Test.JPG'	'11-jul.-2014 19:28:30'	3056x4592x3 uint8
6	'2014-07-14 Bacterias Control.JPG'	'14-jul.-2014 14:20:50'	3056x4592x3 uint8	'2014-07-14 Bacterias Test.JPG'	'14-jul.-2014 14:21:36'	3056x4592x3 uint8
7	'2014-07-15 Bacterias Control.JPG'	'15-jul.-2014 12:59:08'	3056x4592x3 uint8	'2014-07-15 Bacterias Test.JPG'	'15-jul.-2014 12:59:44'	3056x4592x3 uint8
8	'2014-07-17 Bacterias Control.JPG'	'16-jul.-2014 17:40:20'	3056x4592x3 uint8	'2014-07-17 Bacterias Test.JPG'	'16-jul.-2014 17:41:14'	3056x4592x3 uint8
9	'2014-07-21 Bacterias Control.JPG'	'21-jul.-2014 13:54:46'	3056x4592x3 uint8	'2014-07-21 Bacterias Test.JPG'	'21-jul.-2014 13:55:54'	3056x4592x3 uint8
10	'2014-07-22 Bacterias Control.JPG'	'22-jul.-2014 14:47:22'	3056x4592x3 uint8	'2014-07-22 Bacterias Test.JPG'	'22-jul.-2014 14:48:34'	3056x4592x3 uint8
11	'2014-07-25 Bacterias Control.JPG'	'25-jul.-2014 13:01:44'	3056x4592x3 uint8	'2014-07-25 Bacterias Test.JPG'	'25-jul.-2014 13:02:06'	3056x4592x3 uint8
12	'2014-09-03 Bacterias Control.JPG'	'02-sep.-2014 22:23:02'	3056x4592x3 uint8	'2014-09-03 Bacterias Test.JPG'	'02-sep.-2014 22:23:28'	3056x4592x3 uint8
13	'2014-09-16 Bacterias Control.JPG'	'16-sep.-2014 13:59:14'	3056x4592x3 uint8	'2014-09-16 Bacterias Test.JPG'	'16-sep.-2014 13:59:48'	3056x4592x3 uint8
14						

Figura 8. Resultado de la toma de datos organizada.

Binarización de imágenes

Una vez obtenida esta estructura se puede proceder a procesar las imágenes. El segundo paso consistirá en pasar la imagen a binario para poder identificar más fácilmente donde se encuentran las probetas. La función elegida es `img2bw`, pero para ello es necesario pasar la imagen a escala de grises en primer lugar. Con `rgb2gray` se puede seleccionar el valor de un parámetro por debajo del cual todo se pone a 0, para intentar eliminar información innecesaria y por encima del mismo, todo se queda a 1.

Saturacion en B/N

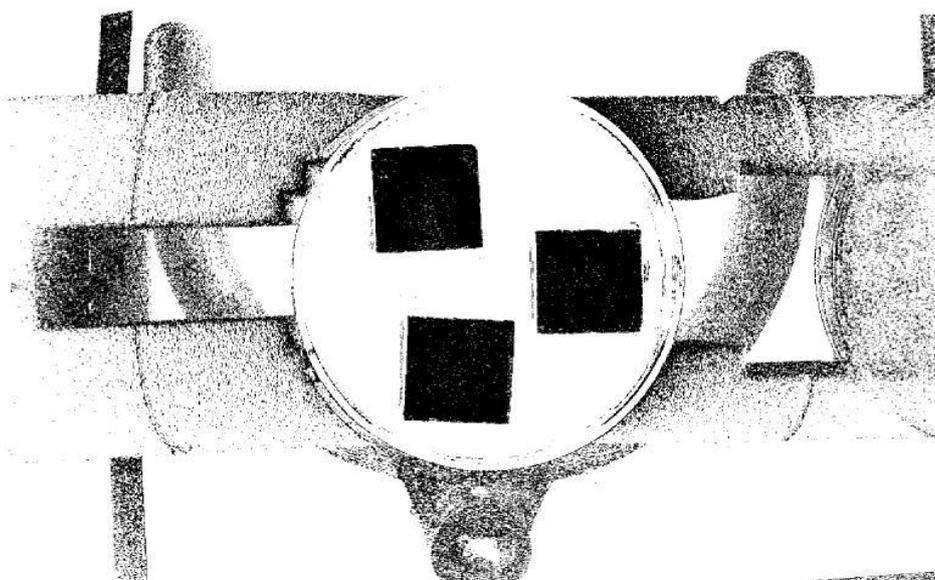


Figura 9. Binarización de la imagen con valor del parámetro = 0.2

Como se puede observar, a primera vista los cuadrados que dibujan las probetas son claramente identificables, sin embargo, lo que se necesita es obtener una imagen en la que solo queden los cuadrados y el resto aparezca en blanco, para reducir tiempo de procesado. Además, si se amplía la imagen, se observa que el interior de los cuadrados no es macizo, sino que existen píxeles vacíos, lo que provocará problemas a la hora de identificar los tres objetos.

Segmentación de probetas

Las técnicas elegidas para solucionar este problema son la *dilatación* y la *erosión*. Estas operaciones son fundamentales en el procesamiento morfológico. De hecho, la mayoría de los algoritmos están basados en estas dos operaciones. Básicamente, la dilatación suma píxeles a las fronteras de los objetos en una imagen, mientras que la erosión elimina los píxeles sobre las fronteras de los objetos [2].

Dilatación binaria:

Dada una imagen A, y un elemento estructural B, (ambos imágenes binarias con fondo blanco), la dilatación de A por B se define como:

$$A \oplus B = \{x \text{ tal que } (\hat{B})_x \cap A \neq \emptyset \}$$

El primer elemento de la dilatación, A, está asociado con la imagen que se está procesando y el segundo recibe el nombre de elemento estructural, la forma que actúa sobre A en la dilatación para producir dicha operación. Al realizar una dilatación con un elemento estructural que no contiene el cero, lo que se realiza es una expansión de la imagen [3].

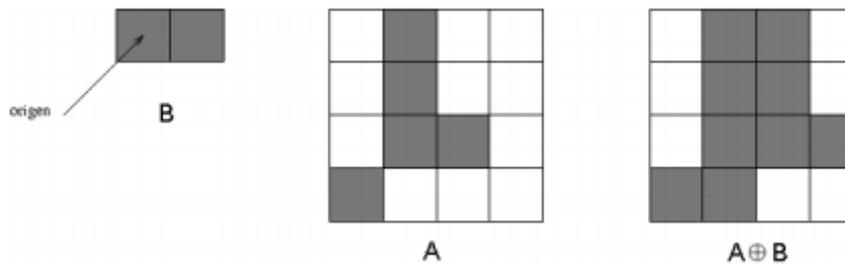


Figura 10. Ejemplo de dilatación

Es decir que, en la dilatación, el valor del píxel de salida es el valor máximo de todos los píxeles en la vecindad del píxel de entrada. Si alguno de los píxeles es igual a uno, el de salida se coloca a uno.

Erosión binaria:

La erosión es la operación morfológica dual de la dilatación. Es la transformación que combina dos conjuntos usando el concepto de inclusión:

Dada una imagen A, y un elemento estructural B, (ambos imágenes binarias con fondo

blanco), la erosión de una imagen A por un elemento estructural B, es el conjunto de todos los elementos x para los cuales B trasladado por x está contenido en A:

$$A \ominus B = \{x \text{ tal que } B_x \subseteq A\}$$

Mientras que la dilatación puede representarse como la unión de los traslados, la erosión puede representarse como la intersección de los traslados negativos, que se concibe usualmente como una reducción de la imagen original.

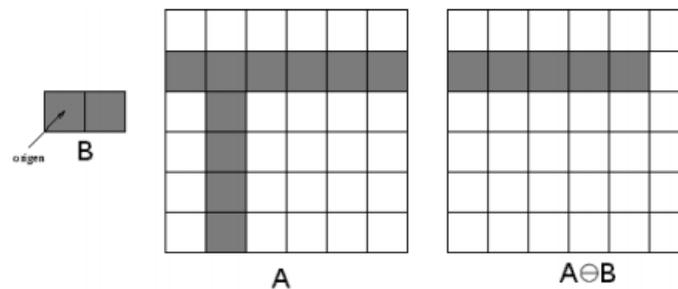


Figura 11. Ejemplo de erosión

Es decir que, en la erosión, el valor del píxel de salida es el valor mínimo de todos los píxeles en la vecindad del píxel de entrada. Si alguno de los píxeles es igual a cero, el píxel de salida se coloca a cero.

Tanto para eliminar todos los píxeles sobrantes alrededor de los tres objetos como para rellenar el interior de los mismos, se modifica el área de los mismos. Por ello, si se realiza el mismo número de dilataciones que de erosiones, y viceversa, se puede mantener la máscara inicial de los objetos.

Las posibilidades de orden de las distintas operaciones son diversas, pero en este caso, las principales son las siguientes: se elige una primera serie de erosiones con un elemento estructural de 3x3 y a continuación de 12x12, para eliminar la mayor parte de los píxeles sueltos de alrededor; una segunda serie, con un elemento estructural de 9x9 y a continuación de 43x43, para rellenar el interior de los tres objetos; y una última serie de 3x3 de nuevo para dejar únicamente las 3 probetas sobre el fondo blanco.



Figura 12. Resultado de la primera serie: erosión, SE=3x3 y 12x12

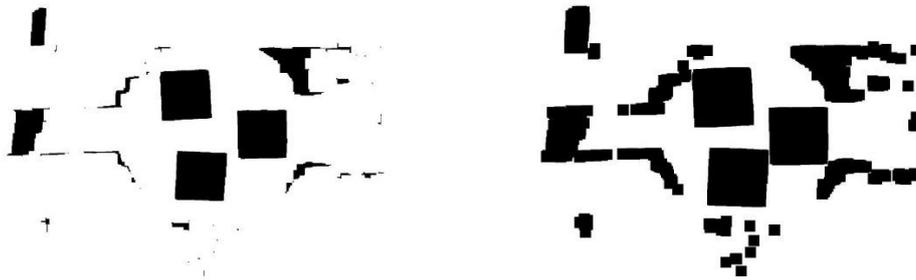


Figura 13. Resultado de la segunda serie: dilatación, SE=9x9 y 43x43



Figura 14. Resultado de la tercera serie: erosión, SE= 3X3.

Una vez se obtiene una imagen binaria donde únicamente se encuentran los tres objetos, llega el momento de localizarlos. Para ello se recurre a la función *regionprops* que proporcionará tanto la localización de las probetas, como el centro de las mismas y su área.

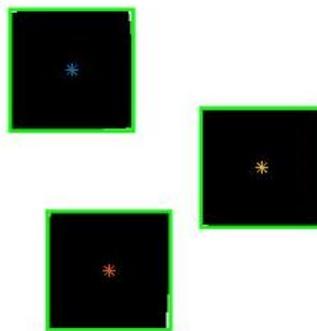


Figura 15. Resultado de la función "regionprops"

Con la localización de las probetas se puede realizar un recorte de las imágenes para obtener cada objeto por separado. Para ello se recurre a la función “imcrop”, a la cual se le pasa la coordenada x, la coordenada y, lo que abarca en horizontal y lo que abarca en vertical.



Figura 16. Probetas segmentadas en diferentes subplots

Si a lo largo de las diferentes tomas temporales, las probetas se mantuvieran en la misma posición, se podría aplicar el resultado obtenido de la función anterior para obtener directamente la estructura de resultados. Puesto que sí existe dicho desplazamiento, sería necesario aplicar todo este procesamiento a cada una de las imágenes independientemente, lo cual aumenta enormemente el tiempo de procesado.

Intentando aplicar estas operaciones morfológicas al resto de imágenes, se observa que conforme se avanza en el tiempo, y el crecimiento del microorganismo comienza a ser visible, se complica la localización de los objetos de interés, siendo necesario un procesado independiente y personalizado de cada imagen, siendo para las últimas tomas indiferenciables. Esto último debido no solo al desplazamiento de las probetas, sino al medio de cultivo, cuyo color deja de diferenciarse claramente del de la superficie de los objetos de interés. Llegado a este punto, es necesario pensar en una alternativa para localizar los objetos en el resto de tomas que no suponga un procesado morfológico [4].

Una opción viable consiste en realizar una transformación de las imágenes que coloque las probetas de cada una de las tomas en la misma posición, para así poder utilizar la posición obtenida en el primer procesado morfológico. Sin embargo, esto tampoco es tan sencillo, ya que también se observa que cada uno de los tres objetos no se desplaza (y/o rota) de forma simétrica, por lo que la transformación deberá realizarse, no de cada imagen, sino de cada probeta por separado.

Para realizar este proceso, se debe obtener en primer lugar, la máscara correspondiente a la localización de los objetos en el primer procesado (realizado con *regionprops*). Con la función *bwlabel* se pueden etiquetar todos los píxeles de un mismo objeto con un número automáticamente. Así pues, el primer objeto tendrá todos los píxeles que lo compongan a uno, el segundo a dos y el tercero a tres [5].

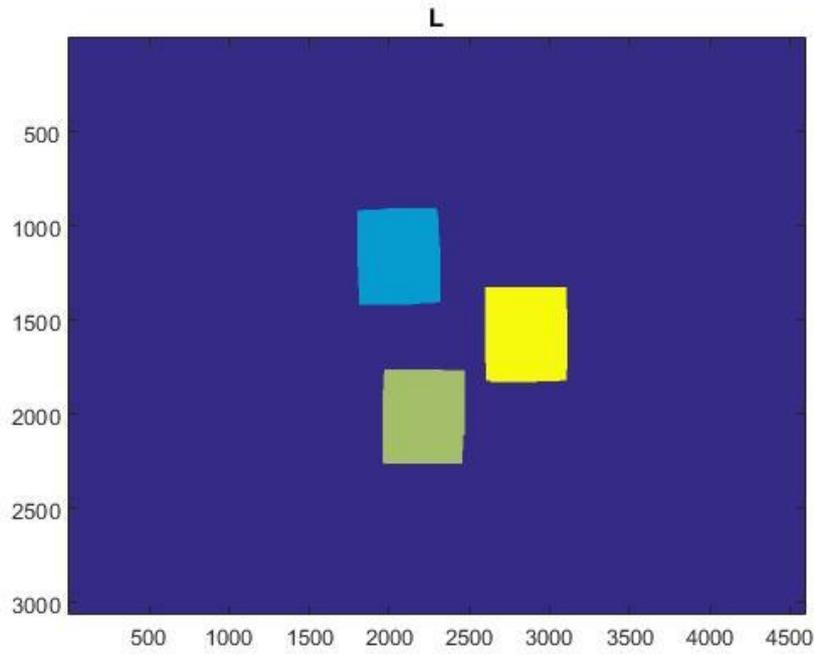


Figura 17. Máscara de las tres probetas, cada uno etiquetado con un número

Ahora se debe generar una máscara independiente para cada objeto. Para ello se copia la máscara original en una nueva variable, se localizan los píxeles que contienen los otros dos objetos que no se necesitan y se ponen a cero. Este proceso se realiza para cada una de las tres probetas.

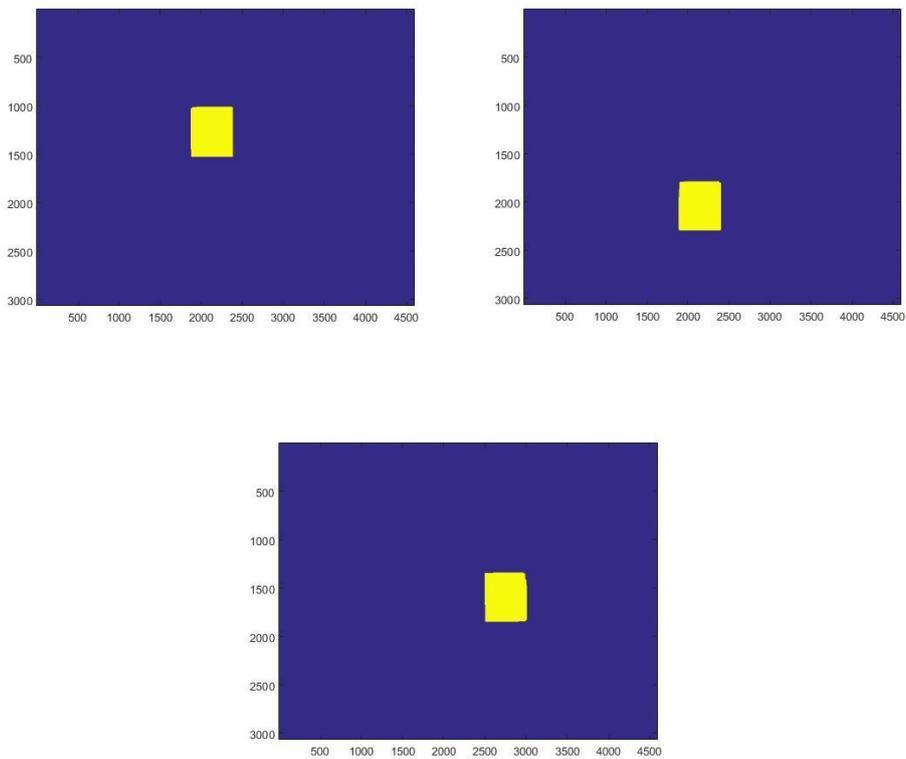


Figura 18. Máscaras independientes para cada uno de los objetos de interés

La función de transformación debe trabajar con imágenes en las que únicamente se desplace el objeto de interés, pero éste debe tener cierto margen alrededor, ya que, si se trabajara con la máscara tal cuál para cada una de las tomas, las probetas saldrían cortadas. Con el fin de abarcar un espacio razonable dentro del cual se pueda observar la probeta en cada una de las tomas temporales, se procede a dilatar cada una de las tres máscaras obtenidas. Se elige un elemento estructural de 400x400 para el ensayo de control, y de 500x500 para el ensayo de test.

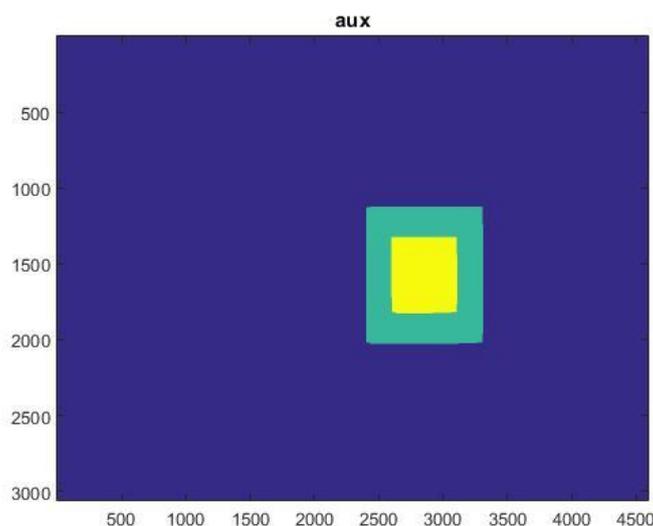


Figura 19. Comparación de la máscara original (amarilla) y la máscara con el margen de localización (verde)

Co-registro automático:

Una vez las imágenes han sido procesadas, y se han segmentado las regiones de interés, se pueden someter a las diferentes transformaciones espaciales. Este proceso es el más trascendental de todo el proyecto ya que de la calidad de la transformación aplicada depende la fiabilidad de los resultados de los parámetros que se van a analizar. Por tanto, el objetivo es conseguir reducir significativamente el error de corrección, es decir, el programa ha de aplicar la transformación necesaria a las imágenes recibidas para obtener una correspondencia pixel a pixel. Este proceso debe realizarse, al igual que la segmentación, sin ningún tipo de asistencia por parte del usuario del programa.

Para enfatizar la importancia de aplicar este concepto y comprender mejor su funcionamiento de forma visual, a continuación, se presentan dos zonas con zoom de MIP pertenecientes a un caso de estenosis arterial, antes de la intervención (imagen izquierda) y después de la misma (imagen de la derecha). En la de la izquierda se ha seleccionado una región de interés (ROI) sobre una arteria. Trasladando las coordenadas de esa ROI sobre la imagen del POST se puede observar claramente como no coincide en el lugar exacto. Estos desplazamientos en las imágenes son debidos a limitaciones de la propia intervención (tensión arterial, respiración y movimientos del paciente, etc.) ya que, durante toda la intervención, el paciente permanece despierto. Si se tratara con estas imágenes sin realizar el corrección, generaría unos resultados anormales sin información relevante. Por tanto, el objetivo de realizar el corrección no es otro sino hacer que este tipo de casos coincidan tras aplicar la corrección pertinente [6].

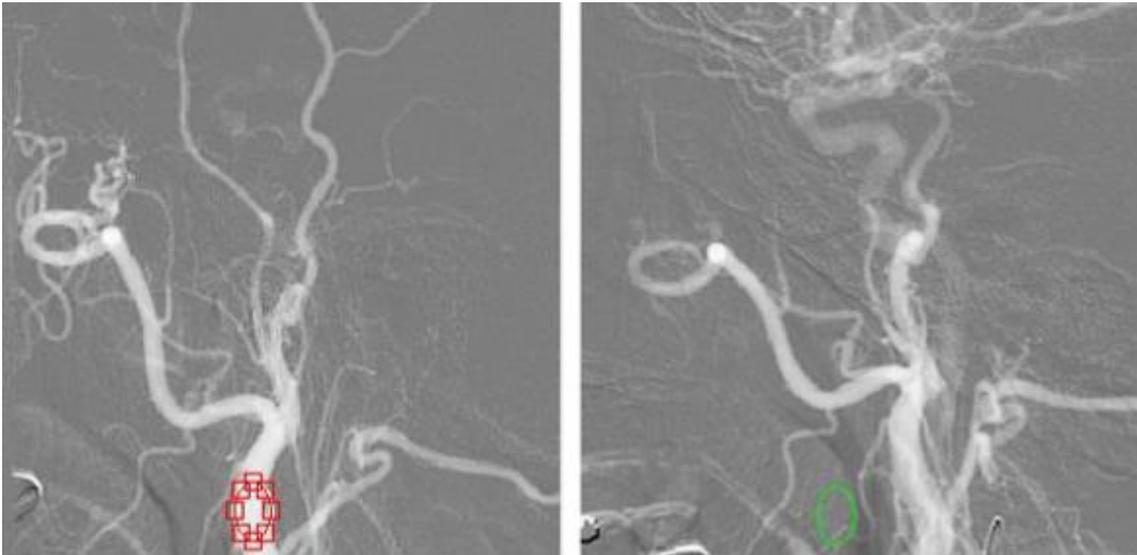


Figura 20. Comparación de la traslación en coordenadas de la misma ROI, sin corregistrar

Estos algoritmos de registro pueden ser clasificados de acuerdo con los modelos de transformación que utilizan para relacionar el espacio de la imagen objetivo al espacio de la imagen referencia. La primera categoría de modelos de transformación incluye las transformaciones lineales, que son la de traslación, rotación escalamiento y otras transformaciones afines. Las transformaciones lineales son de naturaleza global, por lo tanto, no pueden modelar diferencias geométricas locales entre las imágenes. La segunda categoría de transformaciones, es capaz de deformar localmente la imagen objetivo para alinearla con la imagen de referencia. Las transformaciones que se han empleado en este proyecto son lineales ya que el tipo de movimiento que sufren los objetos de interés implica únicamente desplazamiento o rotación; en ningún momento la imagen necesita ser deformada para conseguir correspondencia con la de referencia.

ESTUDIO DE LOS ALGORITMOS DE CO-REGISTRO

Como ya se ha dicho, el modelo de transformación a utilizar es el lineal. Las transformaciones geométricas modifican la relación espacial entre píxeles, y eso en términos del procesamiento digital de imágenes consiste en dos operaciones básicas:

1. Una transformación espacial que define la reubicación de los píxeles en el plano imagen.
2. Una interpolación de los niveles de grises, los cuales tienen que ver con la asignación de los valores de intensidad de los píxeles en la imagen transformada.

Para ello, usando el “Image Processing Toolbox” de “Matlab”, se encontró la función “imregister” y la función “imregtform”. Mediante estas funciones se obtiene una matriz de 3x3, denominada matriz de transferencia, la cual indica la transformación rígida que sufre la imagen movida respecto a la de referencia; en el caso de que no exista desplazamiento, la matriz resultante será la matriz identidad. Para evidenciar su significado, se muestran en primer lugar unos ejemplos visuales y a continuación se explica el significado de cada celda de la matriz.

1	2	3	4	5	6	7	8	9	10
1	2	3							
1.0034	-0.0234	0							
0.0234	1.0050	0							
-6.5389	-6.5424	1							

Figura 21. Matriz de transformación del algoritmo de corregistro

Al no ser una matriz identidad, significa que existe algún tipo o conjunto de transformaciones geométricas. Se puede observar que la última columna es (0 0 1), esto significa que es una transformación afín, ya que por definición, todas las transformaciones afines deben tener en la última columna de la matriz 0 0 1 ([zeros(n,1);1]).

La transformación afín es aquella en la que las coordenadas (x', y') del punto imagen son expresadas linealmente en términos de las del punto original (x,y). Es decir, la transformación viene dada por las ecuaciones:

$$x' = ax + by + m$$

$$y' = cx + dy + n$$

Cuando m=n=0 las ecuaciones anteriores se convierten en el prototipo de una transformación lineal. En resumen, una transformación afín del plano es el resultado de combinar una transformación lineal con una traslación. A continuación, se pueden ver los tipos de transformaciones afines que existen:

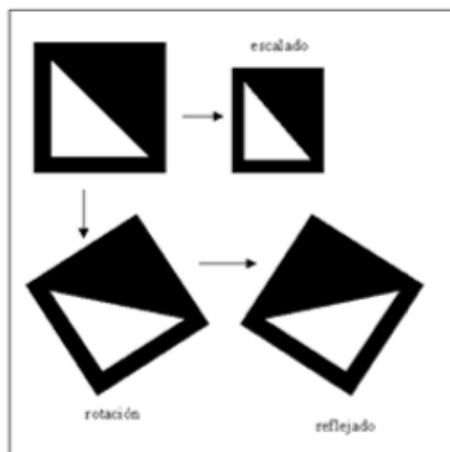


Figura 22. Tipos de transformaciones afines

En la siguiente tabla se puede observar la matriz de transformación (MT) asociada a cada tipo de transformación afín:

Transformación afín	Ejemplo	Matriz de transformación (MT)	
Traslación		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	t_x especifica el desplazamiento a lo largo del eje x. t_y especifica el desplazamiento a lo largo del eje y.
Escalado		$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	s_x especifica el factor de escalado a lo largo del eje x. s_y especifica el factor de escalado a lo largo del eje y.
Inclinado		$\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	sh_x especifica el factor de inclinación a lo largo del eje x. sh_y especifica el factor de inclinación a lo largo del eje y.
Rotación		$\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	q especifica el ángulo de rotación

Tabla 1. Matrices de transformación asociadas a cada transformación afín

En este caso, la función que se probará en primer lugar es “imregister”, que tiene la siguiente morfología:

```
moving_reg = imregister(moving, fixed, transformType, optimizer, metric)
```

Donde:

Moving: imagen que se debe corregir en escala de grises.

Fixed: imagen de referencia en escala de grises.

TransformType: tipo de transformación que debe realizar la función.

Optimizer/metric: configuración deseada para las imágenes a registrar.

Para el tipo de transformación geométrica se debe elegir uno de los siguientes cuatro tipos:

Transform Type	Description
'translation'	(x,y) translation.
'rigid'	Rigid transformation consisting of translation and rotation.
'similarity'	Nonreflective similarity transformation consisting of translation, rotation, and scale.
'affine'	Affine transformation consisting of translation, rotation, scale, and shear.

Tabla 2. Tipos de transformaciones para el correregistro y su correspondiente nomenclatura MATLAB

Se escoge el tipo 'rigid' ya que en las tomas se producen variaciones tanto de posición como de rotación. En cuanto a los parámetros optimizer y metric se utiliza el siguiente comando, que los generará automáticamente (monomodal simboliza que las imágenes han sido capturadas en el mismo dispositivo):

```
[optimizer,metric]=imregconfig('monomodal');
```

El resultado tras aplicar la función, comparado con la imagen sin la corrección aplicada, es el siguiente:

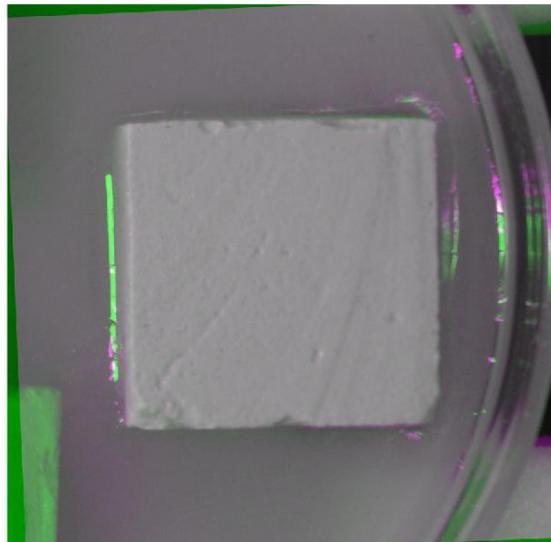


Figura 23. Resultado del correregistro automático aplicado a la segunda toma

Una vez se obtiene la imagen corregida, sería oportuno dejar únicamente los píxeles que abarcan la probeta en cuestión, y el resto dejarlos a 0 para facilitar el análisis de color. Para ello lo más sencillo será generar una máscara con el BoundingBox obtenido de la segmentación:

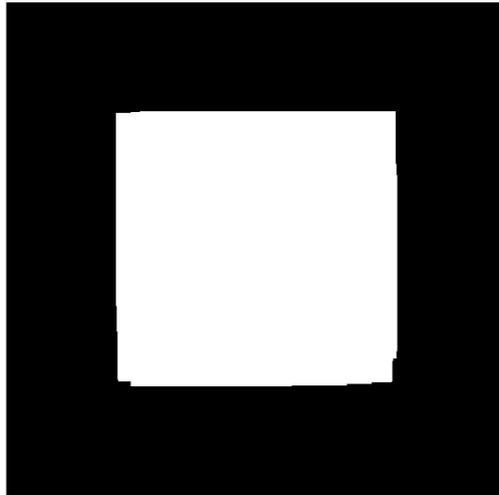


Figura 24. Máscara usada para obtener únicamente los pixeles que abarcan la probeta objetivo

Ahora solo queda multiplicar esta máscara por la imagen corregida y se obtendrá la información deseada para proceder al análisis de color:

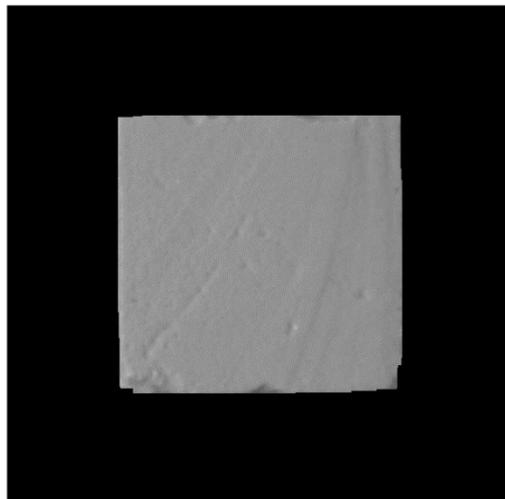


Figura 25. Resultado de multiplicar la máscara por la imagen corregida

Sin embargo, llegado a este punto se plantea un leve problema, la imagen obtenida es en escala de grises y la información que se precisa debe ser en formato rgb para poder realizar un análisis completo del color. La función inversa a la utilizada, `rgb2gray`, no existe, por lo que es necesario profundizar en la función `"IMREGISTER"` para poder realizar la transformación a la imagen a color.

Tras mucho investigar, se da con las funciones `"IMREGTFORM"` e `"IMWARP"` cuyo funcionamiento es similar al de la función `"IMREGISTER"`, con la diferencia de que se puede utilizar la matriz de transformación de la manera que se desee.

La función "imregtform" utiliza también las imágenes a registrar en escala de grises para calcular la matriz de transformación, con la diferencia de que mientras que la función "imregister" aplicaba directamente esta matriz a la imagen en escala de grises, esta función permite almacenar esta misma en una variable. La morfología es la misma que para la función "imregister".

```
tform = imregtform(moving, fixed, 'rigid', optimizer, metric);
```

La función "imwarp" utiliza la matriz de transformación calculada para aplicarla a la imagen que se elija. Esto permite pasarle la imagen en formato rgb y realizar la corrección correcta. La morfología de esta nueva función es la siguiente:

```
imagenout=imwarp(moving,tform, 'OutputView',...  
...imref2d(size(moving)));
```

Donde:

'OutputView': genera el tamaño y la localización de la imagen de salida en función del sistema mundial de coordenadas.

Imref2d: crea un objeto imref2d dado un tamaño de imagen concreto. Esta sintaxis construye un espacio de referenciapara el caso por defectoen el cual el sistema mundial de coordenadas está co-alineado con el sistema intrínseco de coordenadas.

El resultado de aplicar esta función es el siguiente:

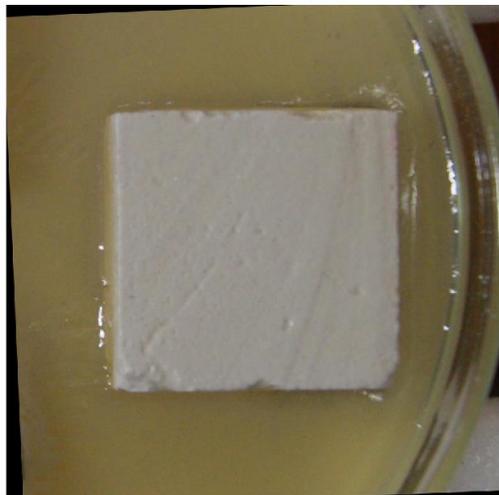


Figura 26. Resultado del corregistro usando la nueva función "imwarp" que permite obtener un resultado RGB

Lo único que queda es multiplicar la máscara anteriormente calculada, por cada una de las componentes (R, G y B) de la matriz. Ahora sí que se obtiene el resultado preciso para ser almacenado en la estructura de resultados.



Figura 27. Resultado de aplicar la máscara a la nueva imagen corregistrada a color

Para obtener la batería de resultados, se realiza un bucle que recorra todas las posiciones de la estructura inicial y vaya generando y guardando todos los resultados en dos estructuras diferentes, una para el ensayo de control y otra para el de test.

En un primer momento, para evitar grandes diferencias entre las imágenes que se van a corregistrar (ya que el entorno de las probetas se va ensuciando conforme pasa el tiempo, lo que dificulta la corrección directa con la primera toma) se decide realizar el corregristo siguiendo el siguiente patrón: La segunda imagen se corregrista con respecto a la primera, la tercera con respecto a la segunda corregrida, la cuarta con respecto a la tercera corregrida, y así sucesivamente. Para ello, no solo se debe almacenar en la estructura el resultado de cada iteración, sino también guardar de una iteración a la siguiente, la imagen anterior corregrida (ya que el resultado se guarda con la máscara aplicada, y se pierde información).

El resultado obtenido para el ensayo de control es el siguiente:

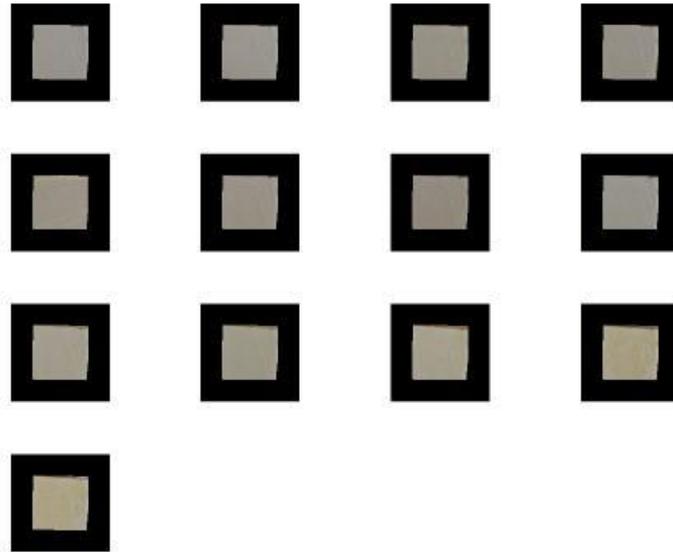


Figura 28. Resultado del correjistro automático para todas las tomas de la probeta 1 del ensayo CONTROL



Figura 29. Resultado del correjistro automático para todas las tomas de la probeta 2 del ensayo CONTROL

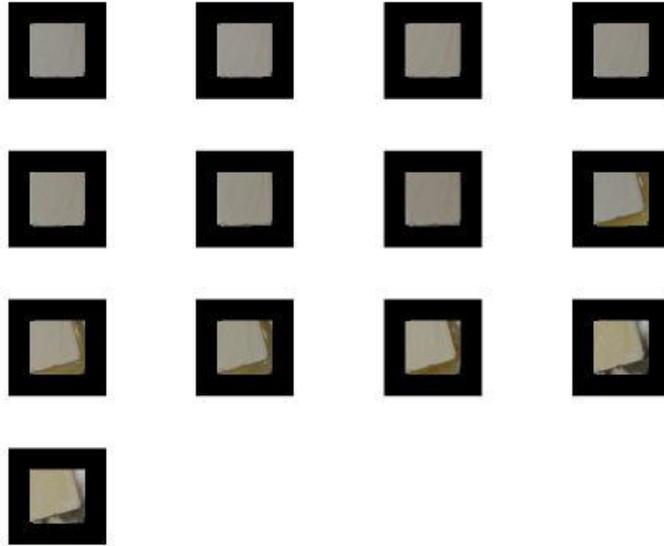


Figura 30. Resultado del correregistro automático para todas las tomas de la probeta 3 del ensayo CONTROL

El resultado obtenido para el ensayo de test es el siguiente:



Figura 31. Resultado del correregistro automático para todas las tomas de la probeta 1 del ensayo TEST

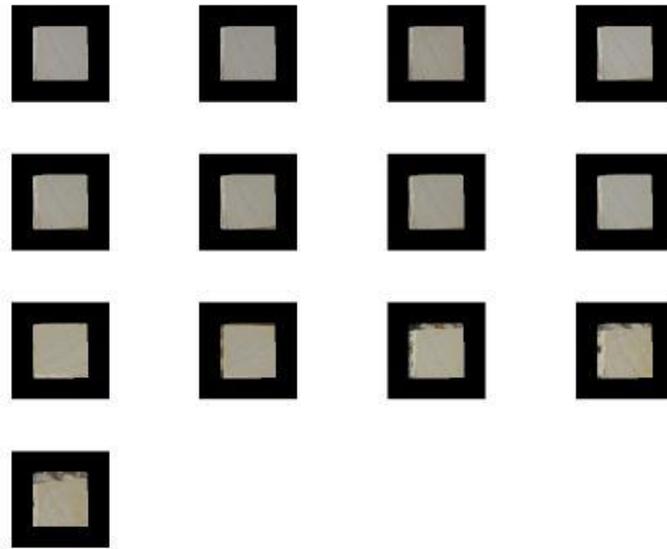


Figura 32. Resultado del correjistro automático para todas las tomas de la probeta 2 del ensayo TEST

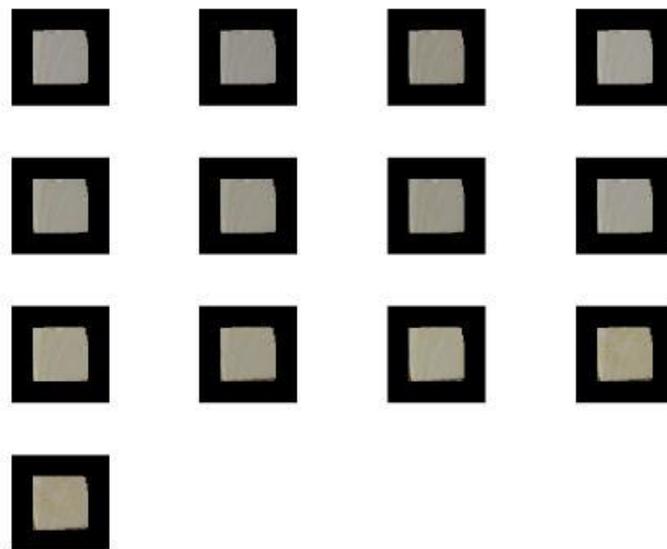


Figura 33. Resultado del correjistro automático para todas las tomas de la probeta 3 del ensayo TEST

Como se puede observar, la corrección ha sido parcialmente correcta ya que ha funcionado perfectamente para la primera probeta del control y para la segunda y tercera del test, sin embargo, para las restantes ha fallado en la mitad de las tomas temporales.

Para solucionarlo, se piensa en una segunda opción. Es posible, a pesar de comparar con las imágenes corregidas para evitar cambios bruscos en el entorno de las probetas, que esto continúe siendo insuficiente. Por ello, se intenta ejecutar de nuevo el bucle, pero esta vez utilizando las imágenes corregidas con la máscara aplicada para eliminar ese fondo que puede resultar confuso para la función, aplicando la corrección con respecto a una imagen donde únicamente se encuentra la probeta en cuestión.

Esto genera un resultado prácticamente idéntico, por lo que se puede deducir que el entorno de las probetas no es algo que entorpezca la acción del corregistro.

En la tercera opción, se piensa que el problema no es el entorno que rodea a la probeta, sino la gran variación de desplazamiento y rotación de las probetas. Así que, para evitar correcciones bruscas (ya que se comprueba que las variaciones de la primera imagen con respecto a la última siempre son más evidentes que las de cada una con su correlativa) se decide realizar el corregistro guardando las matrices de transformación de cada iteración y siguiendo el siguiente patrón: la segunda con respecto a la primera, la tercera con respecto a la segunda y después se le aplica la matriz de la corrección de la segunda, la cuarta con respecto a la tercera y después se le aplican las matrices de corrección de la tercera y de la segunda, y así sucesivamente.

El resultado obtenido siguiendo esta opción es totalmente erróneo, siendo el peor de los tres hasta el momento ya que para ninguno de los dos ensayos consigue realizar la corrección correcta de todas las tomas de alguna de las probetas. Además, cabe mencionar el gran coste computacional que supone almacenar y aplicar todas las transformaciones sucesivas en cada iteración, lo cual hace esta opción directamente inviable. De esta prueba también se deduce que el problema no reside en la complejidad de las correcciones de las últimas tomas.

Se plantea por tanto una última opción, realizar el corregistro directamente de todas las tomas con respecto al primer caso. El patrón a seguir en este caso es el más sencillo hasta ahora: la segunda imagen se corregistra con respecto a la primera, la tercera con respecto a la primera, y así sucesivamente.

El resultado obtenido para el ensayo de control es el siguiente:

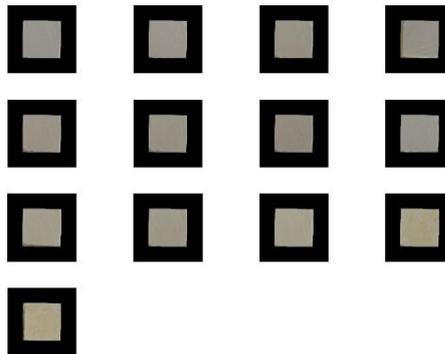


Figura 34. Resultado de la probeta 1 del ensayo de CONTROL

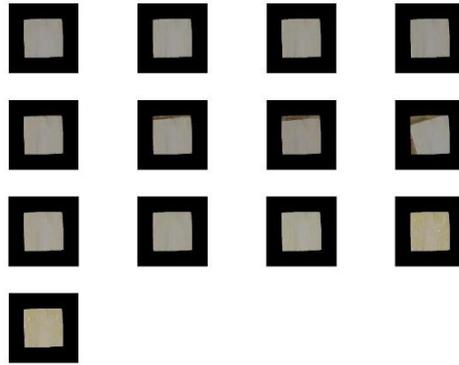


Figura 35. Resultado de la probeta 2 del ensayo CONTROL

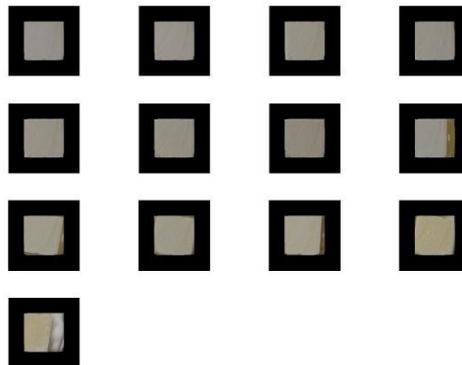


Figura 36. Resultado de la probeta 3 del ensayo CONTROL

Como se puede observar, el resultado es similar a la primera opción; pese a que se reduce el número de tomas corregidas de forma errónea, el algoritmo sigue sin proporcionar el resultado esperado. Es necesario que todas las tomas de todas las probetas de cada ensayo se encuentren corregidas perfectamente, de lo contrario, el cálculo del área de ocupación del microorganismo será totalmente erróneo. Además, es curioso comprobar que en todas las opciones, existen casos temporalmente posteriores, correctamente corregidos, a casos erróneamente corregidos, lo que significa que el problema tampoco reside en que el algoritmo tenga limitaciones por exceso de desplazamiento o rotación.

A pesar de que no existe una opción que dé una solución perfecta al problema, y que ésta última presenta un resultado similar a la primera, se elige la última por tener un menor coste computacional y una sintaxis más sencilla. Por todo esto, llegado a este punto, se plantea una alternativa complementaria al corrección automático. Se trata del corrección manual.

CORREGISTRO MANUAL

Esta opción surge porque hay ocasiones en las que las imágenes no funcionan bien con el corregistro automático. La única razón posible restante de esta situación es que, en las imágenes, no exista un patrón determinado de variación. Es decir, en la imagen original, existen tres objetos que a lo largo del tiempo se mueven de forma totalmente independiente. Esto ocasiona que en las imágenes no exista una variación común a todas ellas (en mayor o menor grado), con lo cual, a pesar de haber segmentado los objetos por separado, en algunas aparecen trozos de otro objeto, o variaciones externas al objeto del cual se pretende corregir la posición. Esto significa que sería necesaria una segmentación individual para cada toma, lo cual llevaría al punto de partida ya que no se podría automatizar ninguna parte del proceso. Es por ello que se valora la necesidad de un complemento para solucionar los casos en los que la corrección automática no funcione correctamente: el corregistro manual.

Para la realización del corregistro manual, se utiliza la función “cpselect” de la toolbox de “Matlab”. Con esta opción, el usuario final del programa podrá seleccionar los puntos principales de las imágenes que quiera hacer coincidir. A continuación se muestra un ejemplo en el cual, el corregistro automático no funciona correctamente:



Figura 37. Ejemplo del mal funcionamiento del corregistro automático

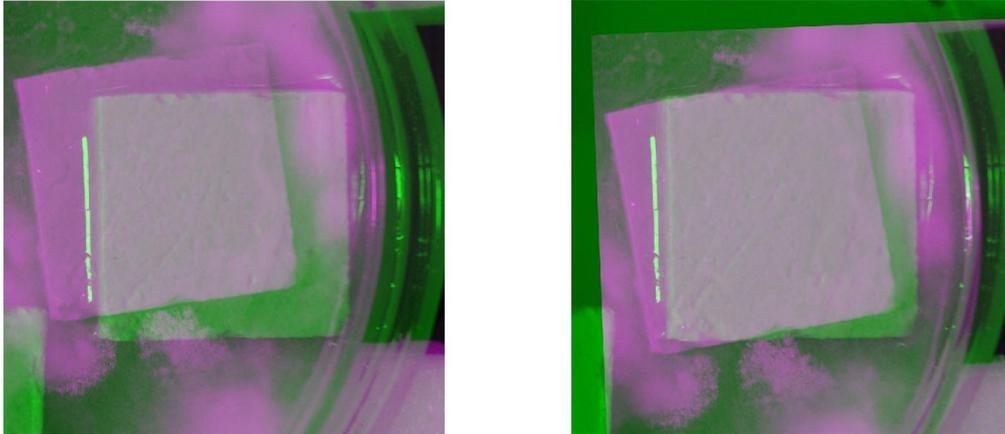


Figura 38. Resultado de la superposicion de las imagenes corregistradas y las originales

En este caso, se puede observar el problema comentado anteriormente. Aparte de la variación de posición de la probeta, existe una diferencia notable entre las dos imágenes. En la de la derecha aparece una parte de otra de las probetas, y por supuesto la suciedad del entorno. Esto genera que no exista un único patrón de desplazamiento entre las dos imágenes ya que las probetas se mueven de forma independiente, y en ningún momento se le da al correjistro automático información de cuál es el objeto que debe correjir. Aquí es donde cobra sentido la opción de correjistro manual.

Para estos casos, se debe generar un algoritmo que permita realizar una corrección manual, indicando cual es el objeto de interés. La sintaxis utilizada para ello es la siguiente:

```
[movingPoints, fixedPoints]=cpselect(moving, fixed, 'Wait', true);

tform = cp2tform(movingPoints, fixedPoints, 'nonreflective ...
...similarity');

[registered, xx, yy]=imtransform(moving, tform, 'xyscale', 1);

registeredfinal=imtranslate(registered, [round(xx(1)), round(yy(1
...)], 'FillValues', 0);
```

La función “cpselect” abre la siguiente interfaz y se pone en pausa hasta que se realice la función pertinente. En ella se deben seleccionar una serie de puntos (cuatro en este caso), y hasta que no se realice esta acción, la ejecución se queda pausada para no generar error, gracias a los parámetros: ...‘Wait’, true).

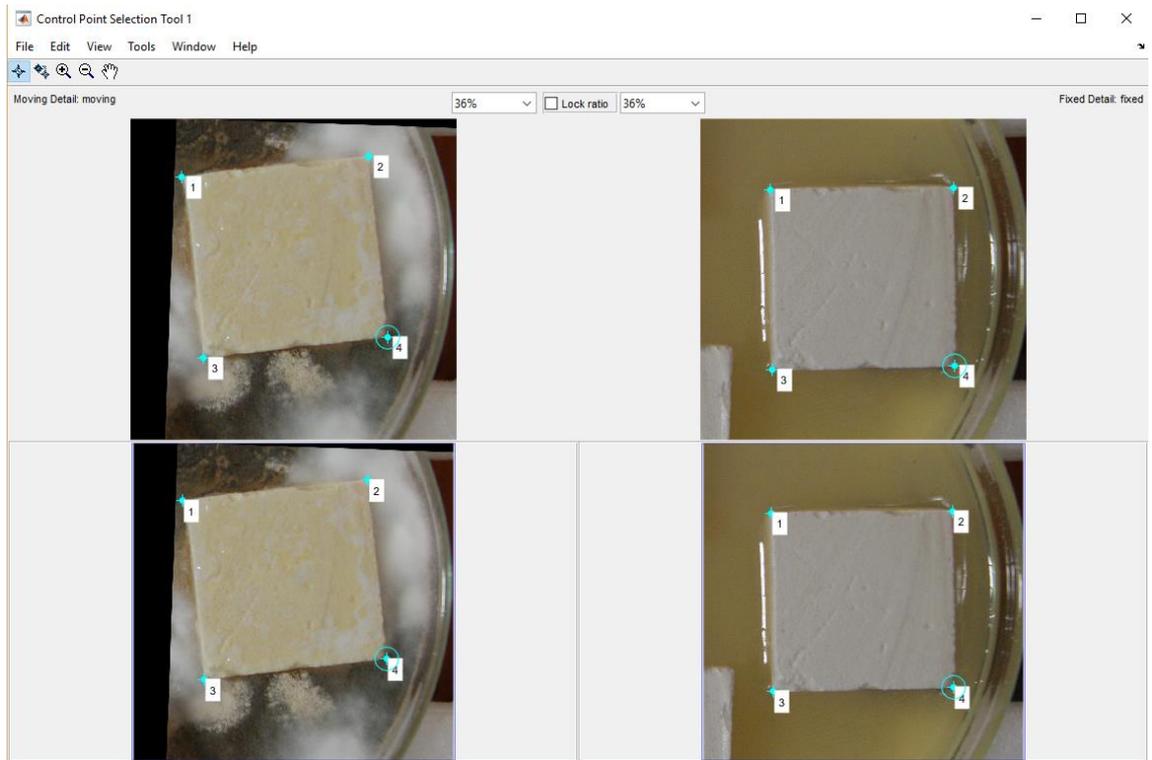


Figura 39. Pantalla de ejecución del correjistro manual

En la pantalla se muestra las dos imágenes que se desea correjistrar, y al terminar de realizar la selección, se genera la matriz de transformación con la función “cp2tform” usando esos puntos.

La aplicación de la matriz de transformación se realiza en dos pasos:

1. Se ejecuta la función “imtransform” para aplicar la matriz
2. Se ejecuta la función “imtranslate” para correjir la traslación

Este segundo paso es necesario ya que la función realiza la corrección en función al Sistema Mundial de Coordenadas. Es por ello que en las variables xx e yy se guarda la diferencia de pixeles en ambos ejes.

Una vez realizados estos dos pasos, el resultado obtenido es el siguiente:

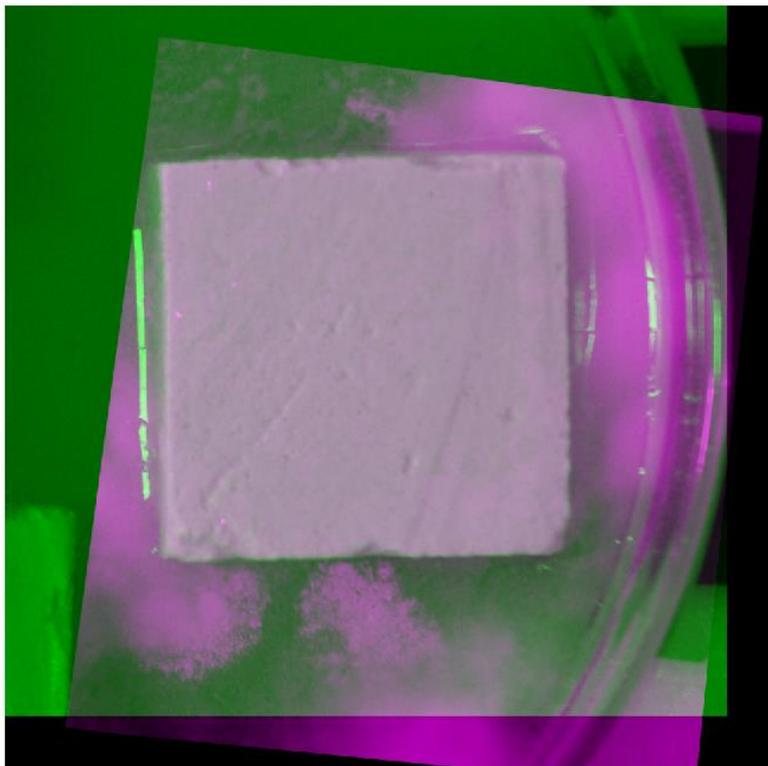


Figura 40. Resultado del correjistro manual (imagenes superpuestas)

Como se puede observar el ajuste es correcto, por lo que este proceso se puede aplicar a aquellos casos en los que el correjistro automático no funcione como es debido.

Al intentar realizar una comparación pixel a pixel, se observa un último problema antes de poder ejecutar este proceso para todos los casos necesarios. La función, al realizar la corrección, modifica el tamaño de imagen por lo que saltará un error al intentar aplicarle la máscara de resultados. Para solucionar este inconveniente, se crean unas condiciones después del correjistro que modifique el tamaño de la imagen al de la imagen original:

- Si la imagen arreglada es de menor tamaño que la de referencia, se completa con ceros.
- Si la imagen arreglada es de mayor tamaño que la de referencia, se recorta.
- Y si es una mezcla de los casos anteriores, se recorta en eje x y se completa con ceros en el eje y, o viceversa.

El código para realizar esta corrección en función del tamaño de la imagen es el siguiente:

```

%%%caso1 La imagen arreglada de mayor tamaño que la fija. Rellenamos
if (size(fixed,1)>size(registeredfinal2,1)) &
(size(fixed,2)>size(registeredfinal2,2))

registeredfinal2(size(registeredfinal2,1):size(fixed,1),size(registeredfinal2,2):size(fixed,2),:)=0;
reg=registeredfinal2;
figure,imshowpair(fixed,reg);

end

%%%caso2 La imagen arreglada de menor tamaño que la fija. Recortamos
if (size(fixed,1)<size(registeredfinal2,1)) &
(size(fixed,2)<size(registeredfinal2,2))

reg=registeredfinal2(1:size(fixed,1),1:size(fixed,2));
figure,imshowpair(fixed,reg);

end

%%%caso3
if (size(fixed,1)>size(registeredfinal2,1)) &
(size(fixed,2)<size(registeredfinal2,2))

registeredfinal2(size(registeredfinal2,1):size(fixed,1),1:size(fixed,2),:)=0;
reg=registeredfinal2(1:size(fixed,1),1:size(fixed,2));

figure,imshowpair(fixed,reg);

end

%%%caso4
if (size(fixed,1)<size(registeredfinal2,1)) &
(size(fixed,2)>size(registeredfinal2,2))

registeredfinal2(1:size(fixed,1),size(registeredfinal2,2):size(fixed,2),:)=0;
reg=registeredfinal2(1:size(fixed,1),1:size(fixed,2));

figure,imshowpair(fixed,reg);

end

```

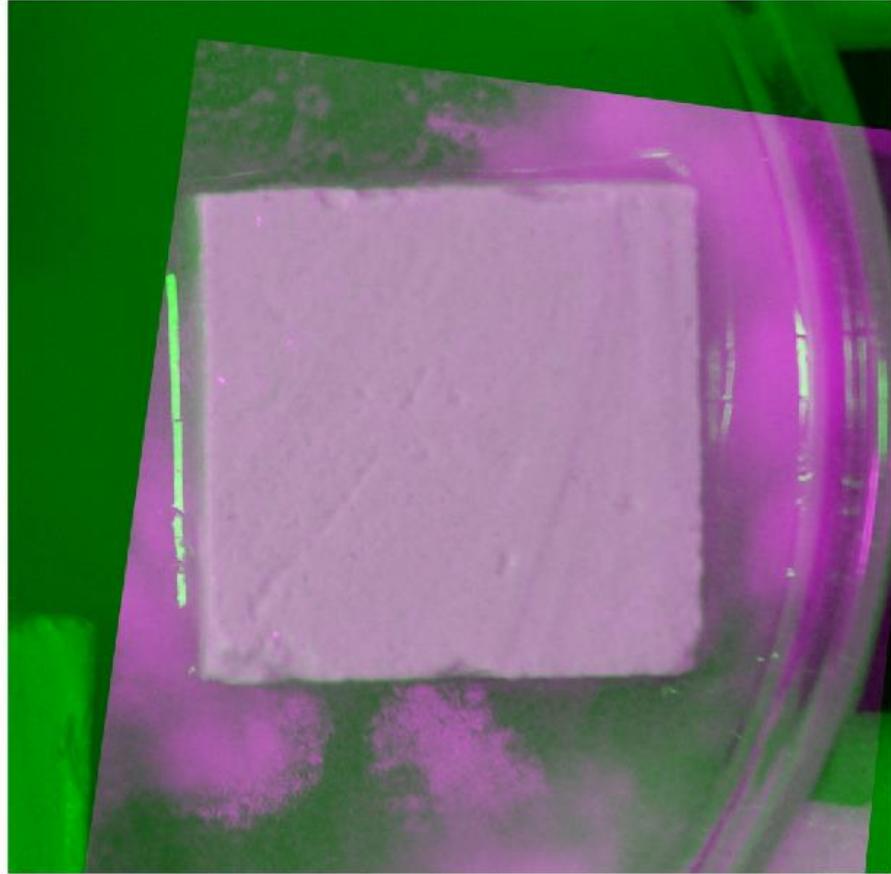


Figura 41. Ajuste del tamaño de la imagen resultado al de la imagen original

Ahora sí que se obtiene un resultado totalmente correcto y se puede aplicar a las imágenes para las que no ha funcionado el corrección automática. Llegado este punto se pueden preparar las estructuras de resultados para proceder con el análisis del grado de ocupación del microorganismo.

Análisis de ocupación del microorganismo

Una vez se preparan y organizan correctamente los datos, se puede proceder al propósito de este proyecto: realizar un análisis del color de la superficie de las probetas y estimar el grado de ocupación del microorganismo para los dos ensayos, con el objetivo de determinar la diferencia entre ambos y así concluir la efectividad del biocida Biotín R.

Para realizar este análisis de color existen varios métodos basados en diferentes modos. En primer lugar, se realiza un estudio de los diferentes modos de representación de imágenes para elegir el óptimo.

MODOS DE COLOR

Se llama modo de color al sistema de coordenadas que permite describir el color de cada pixel utilizando valores numéricos [7].

Los modos más utilizados son los siguientes:

Modo monocromático: se corresponde con una profundidad de color de 1 bit. La imagen está formada por píxeles blancos o píxeles negros puros.



Figura 42. Modo monocromático

Modo Escala de Grises: maneja el color negro y permite 256 tonos de gris entre el blanco y el negro puros.



Figura 43. Modo escala de grises

Modo Color indexado: utiliza un canal de color indexado de 8 bits pudiendo obtener con ello, hasta un máximo de 256 colores.



Figura 44. Modo indexado

Modo RGB: Cada color se forma por combinación de tres canales. Cada canal se corresponde con un color primario: Red (rojo), Green (verde), y Blue (azul). Asigna un valor de intensidad a cada color que oscila entre 0 y 255. De la combinación surgen hasta 16,7 millones de colores. Ejemplo: El valor R:255, G:0, B:0 representa al color rojo puro.



Figura 45. Modo RGB

Modo HSB/HSV: cada color surge de los valores de estos tres parámetros: **Hue (Tono)** que es el valor del color: rojo, azul, verde, etc. En GIMP se expresa en grados y oscila entre 0 y 360. **Saturation (Saturación)** que se refiere a la pureza del color y va del 0% al 100%. **Brightness (Brillo)** referencia la intensidad de luz del color, es decir, la cantidad de negro o blanco que contiene estando su valor entre 0 (negro) y 100 (blanco). Ejemplo: El color rojo puro tiene un código RGB como (255,0,0) y también un código HSB (0,100,100). En la mayoría de programas de tratamiento de imágenes se puede elegir un color introduciendo su código RGB (opción más frecuente) o alternativamente su código HSB. En ambos casos la imagen maneja una paleta de colores de 24 bits.

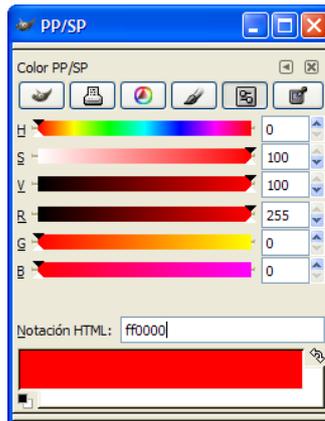


Figura 46. Modo HSV

Modo CMY: cada color se forma por combinación de cuatro canales. Cada canal se corresponde con un color primario de impresión: **C**yan (Cian), **M**agent (Magenta), **Y**ellow (Amarillo). Cada canal puede tener como valor entre 0 y 255. Se trata de imágenes con una profundidad de color de 32 bits.

De todos los modos que se acaban de nombrar, en primer lugar se obvian los que no contemplan los planos de color, es decir, ni el binario, ni el de escala de grises ni el indexado. En cuanto a los restantes, el modo RGB es el modo del cual se parte, y no resulta cómodo trabajar con él teniendo en cuenta que el color que buscamos identificar correctamente es el amarillo, el cual depende de varios de estos canales. En cuanto a los otros dos, parece interesante comprobar si el modo CMY proporciona una representación cómoda para el análisis, ya que uno de sus canales es el amarillo.

MODOS CMY

Tras investigar en la Toolbox, no existe ninguna función de "Matlab" que realice la transformación directa de RGB a CMY, por tanto se crea un pequeño código que realiza esta conversión:

```
function [C,M,Y] = rgb2cmy(R,G,B)

C=1.0 - (double(R)./double(255.0));
M=1.0 - (double(G)./double(255.0));
Y=1.0 - (double(B)./double(255.0));

M=uint8(255.0*M);
C=uint8(255.0*C);
Y=uint8(255.0*Y);

end
```

Tras realizar la conversión se obtienen tres planos equivalentes a los RGB que representan la imagen:

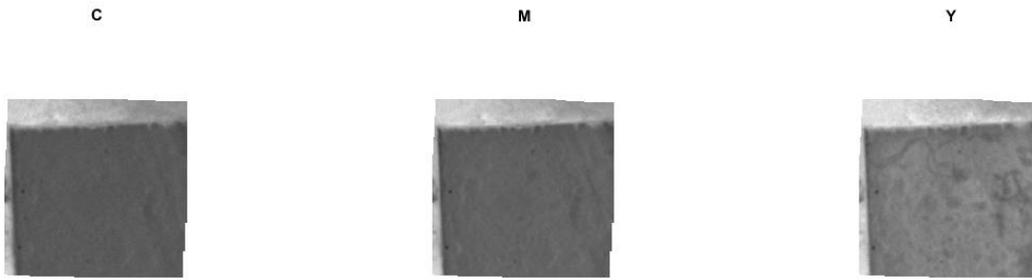


Figura 47. Desglose de las matrices C, M, Y

Para poder observar mejor si el resultado obtenido facilita el análisis que se pretende realizar, a continuación, se compara una de las tomas más avanzadas de una de las probetas (en la que el color amarillo es claramente visible) y el plano, y de este modo:

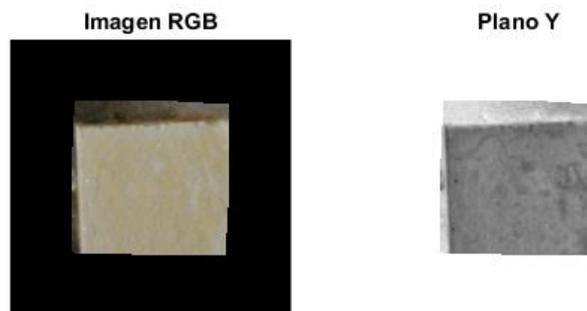


Figura 48. Comparación de la imagen original y el plano Y (amarillo)

Aquí se puede observar claramente que transformar la matriz con este modo no facilita la tarea, ya que, de hacerlo, el plano "Y" debería coincidir en buena medida con la zona amarilla de la superficie de la probeta. Para terminar de comprobar que este modo no es el correcto, se procede a transformar a CMY la primera toma de una de las probetas (que debería traducirse en valores mínimos ya que la probeta está limpia) y la última toma de la misma (en los que los valores deberían ser elevados en las zonas correctas). Una vez hecho esto, se restan los planos Y de ambas imágenes y se observa que la media de la diferencia de valor de la comparación pixel a pixel es inferior a 10 unidades. Por tanto no merece la pena seguir con este modo; se pasa ahora a comprobar la utilidad del modo HSV.

MODO HSV

Se ha visto como los modelos RGB y CMY están relacionados con el hardware (es decir, orientados a dispositivos como monitores o impresoras) o a alguna aplicación donde el objetivo es la manipulación de color. Sin embargo, éstos no son útiles para describir los colores en términos prácticos para la interpretación humana. Por ejemplo, uno no se refiere al color de un automóvil dando porcentajes del contenido de cada uno de los colores primarios. En cambio, el modelo HSV corresponde al modelo más cercano a la manera en que los humanos percibimos el color. Cuando los humanos ven un color lo describen en términos de su tono (H), saturación (S) y su brillo/valor (I/V) [8].

El tono es un atributo cromático que describe un color puro (amarillo, naranja, etc), mientras que la saturación proporciona una medida del grado en que el color puro está diluido en luz blanca. Este modelo debe su utilidad a dos hechos:

1. La componente de intensidad (I) está desacoplada de la información cromática.
2. Las componentes de tono (H) y saturación (S) están íntimamente relacionadas con la forma en que los humanos perciben el color.

Por ello resulta la herramienta óptima para el procesamiento de imágenes a color.

Es común que se desee elegir un color concreto (como este es el caso del amarillo), cuando es así, resulta útil usar la ruleta de color HSV. En ella el matiz se representa por una región circular; una región triangular separada puede servir para representar la saturación y el valor de color deseados de la región triangular [9, 10].

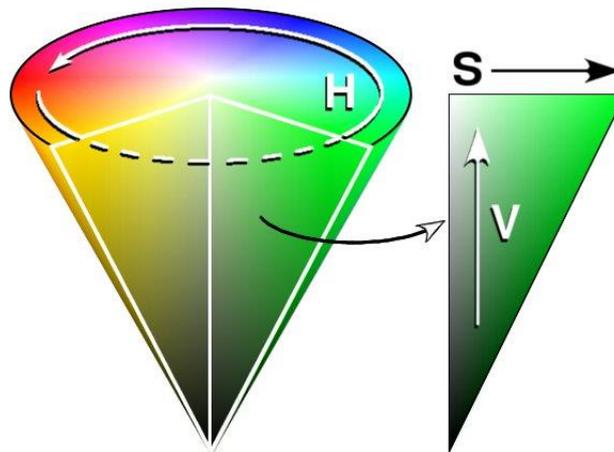


Figura 49. Esquema del modo HSV

El matiz se representa como un grado de ángulo cuyos valores posibles van de 0 a 360 (aunque en algunas aplicaciones se representa de 0% A 100%) Cada valor corresponde a un color.

Ángulo	Color
0 °	Rojo
30 °	Naranja
60 °	Amarillo
90 °	Verde Chartreuse
120 °	Verde
150 °	Verde Primavera
180 °	Cian
210 °	Azul claro (azur)
240 °	Azul
270 °	Violeta
300 °	Magenta
330 °	Rosa

Tabla 3. Equivalencia de ángulos y tonos para el modo HSV

La saturación se representa como la distancia al eje de brillo negro-blanco. Los valores posibles van de 0 a 100%. Cuanto menor sea la saturación de un color, mayor tonalidad grisácea habrá y más decolorado estará [11].

100% → Matiz puro
0% → Grisáceo

El valor representa la altura en el eje blanco-negro. Los valores posibles van del 0 al 100%. 0 siempre es negro. Dependiendo de la saturación, 100 podría ser blanco o un color más o menos saturado.

En este caso sí que existe una función en “Matlab” para realizar la conversión: “rgb2hsv”. Así que el primer paso es aplicarla a la imagen y visualizar los diferentes planos.

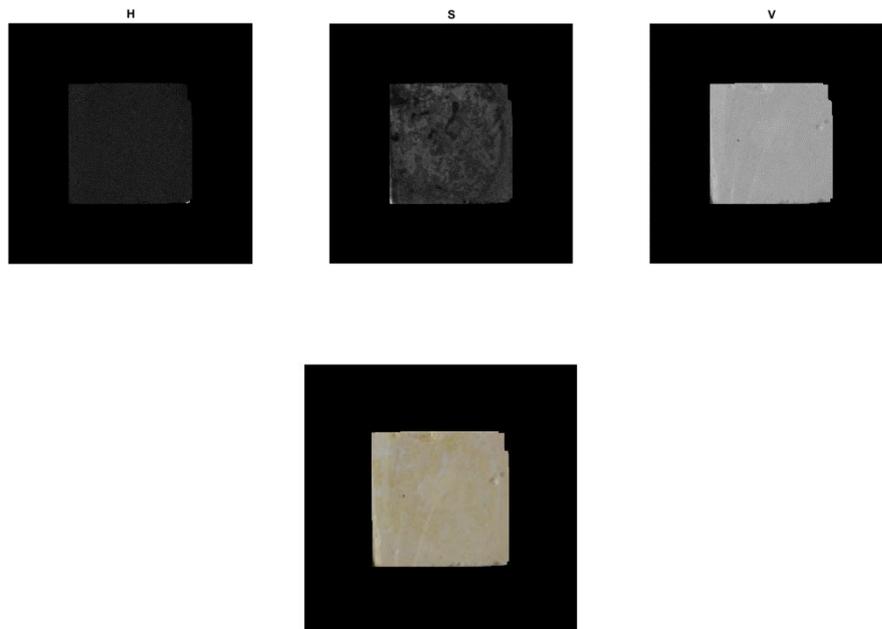


Figura 50. Desglose de panos H, S y V, y la imagen original

Como se puede observar, donde realmente reside la información interesante es en el tono (H) y, sobre todo, en la saturación (S). Es posible que, de primeras, la H no parezca información relevante, pero si se recuerda bien, la H viene a representar qué matiz se desea visualizar, y en este caso es el amarillo, es decir 60° . Cuando se trabaja en "Matlab", los límites van de cero a uno, por lo que si se tiene en cuenta que 30° es naranja y 90° es verde, se elige un rango razonable (de 40° a 80°), es decir de 0,1111 a 0,2222. Esto se puede visualizar también en el histograma del tono, en el cual se comprueba que este es el camino a seguir.

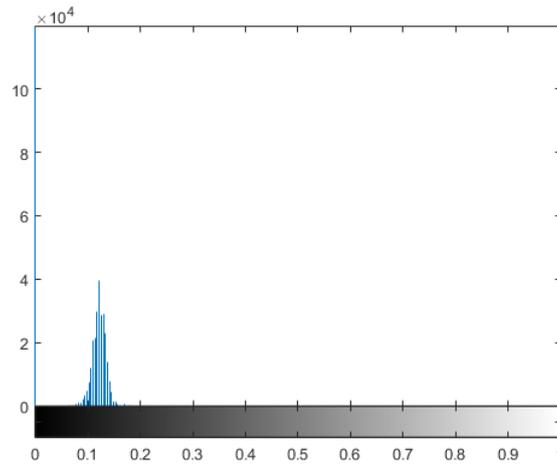


Figura 51. Histograma del tono (H)

En cuanto a la saturación, esto se complica algo más. Cabe recordar que la saturación va desde un tono muy grisáceo hasta el matiz puro de color, en este caso amarillo puro. Ya que las probetas no son blancas puras, algo que sería prácticamente imposible, sería recomendable filtrar la saturación y eliminar la pequeña parte de saturación que representa la probeta origen. Para poder estimar mejor el cual es el punto de inflexión, se procede a representar las imágenes tanto de la primera toma de una de las probetas como de la última, y sus correspondientes histogramas:

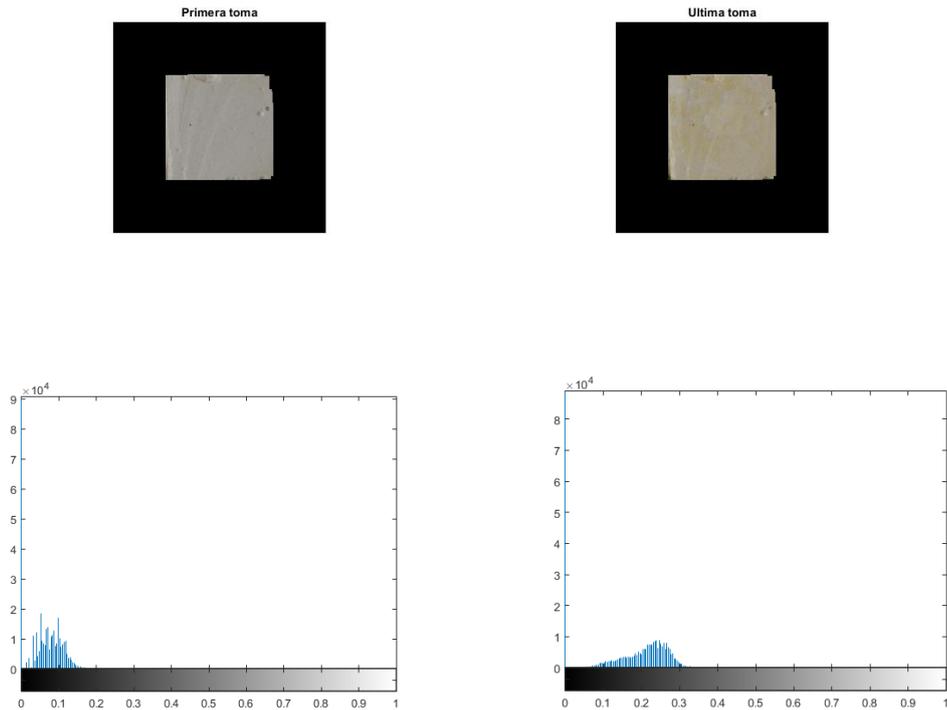


Figura 52. Comparación de los histogramas de tono de la primera toma y la última

Como se puede observar, en ambos histogramas de saturación hay zonas comunes, algo que no interesa, ya que la única variación que existe en las imágenes es el paso de blanco a amarillo. Todo lo que se repita se debe eliminar para no falsear la medida. Es por ello que, observando los histogramas, el umbral mínimo para la saturación se coloca alrededor de 0,15.

El siguiente paso consiste en realizar una máscara para poder contar los píxeles que, según el criterio elegido, coinciden con lo que se denomina un píxel "ocupado". Cabe decir que el objetivo de este análisis no es determinar el grado de saturación de la ocupación, sino determinar la ocupación en función de la saturación. Dicho de otra manera, no importa lo saturado que está el píxel de amarillo, simplemente importa que se pueda considerar amarillo.

Los píxeles que se dejarán a uno en la máscara serán los que cumplan estas dos condiciones al mismo tiempo:

1. El valor de tono (H) debe estar por encima de 0,1111 y por debajo de 0,2222.
2. El valor de saturación (S) debe superar el valor de 0,15.

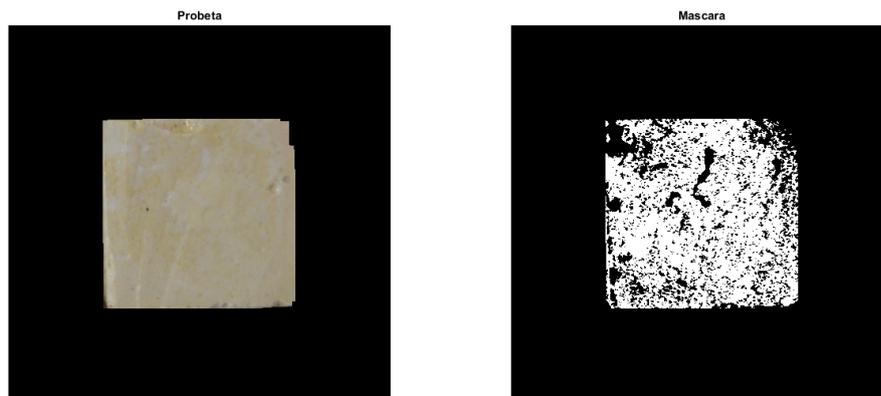


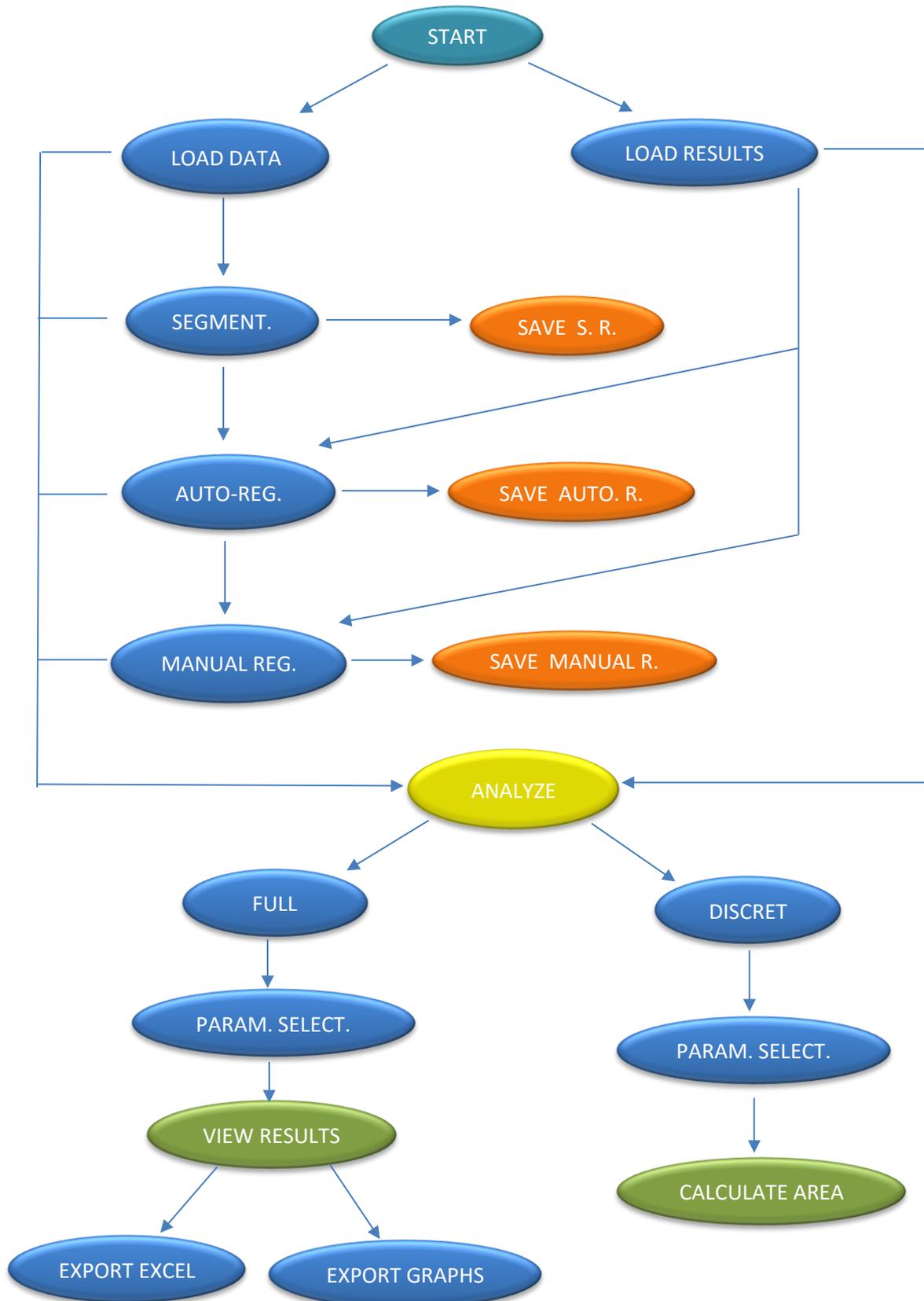
Figura 53. Máscara obtenida con los parámetros: $0,1111 < H < 0,2222$ y $S > 0,15$

Una vez se obtiene la máscara equivalente, únicamente queda realizar un recuento de los píxeles que se encuentran a uno. Para ello se recurre a la función "numel". Para obtener el tanto por ciento de ocupación, que es el dato final que se desea como resultado, se divide este número de píxeles a uno por el número de píxeles total y se multiplica por 100.

Este proceso puede aplicarse a todas las probetas y obtener una matriz con los resultados final que, posteriormente se representarán en gráficas y se exportarán a Excel. Una vez se obtienen todos los algoritmos y se comprueba que funcionan correctamente, llega el momento de realizar una interfaz para facilitar el trabajo al usuario final.

FLUJOGRAMA

A continuación, se presenta un diagrama de bloques en el cuál se pretende reflejar el sistema de funcionamiento del programa realizado, para comprender mejor todo lo explicado anteriormente.



Diseño de la interfaz de usuario

Las interfaces de usuario, también conocidas como GUI, permiten un control sencillo (con uso de ratón) de las aplicaciones de software, lo cual elimina la necesidad de aprender un lenguaje y escribir comandos a fin de ejecutar una aplicación. Las apps de “Matlab” son programas autónomos de “Matlab” con un frontal gráfico de usuario GUI que automatizan una tarea o un cálculo. Por lo general, la GUI incluye controles tales como menús, barras de herramientas, botones y controles deslizantes.

Para realizar esta interfaz, se van creando las diferentes partes de la misma siguiendo el orden de programación de los algoritmos. En primer lugar, la clasificación de los datos de partida. Se opta por un diseño en el que la pantalla de inicio incluya acceso directo a todas las acciones: tanto a la carga de directorio, la carga de resultados obtenidos en sesiones anteriores, ambos corre registros y el análisis final de los datos. El aspecto inicial es el que se muestra a continuación:

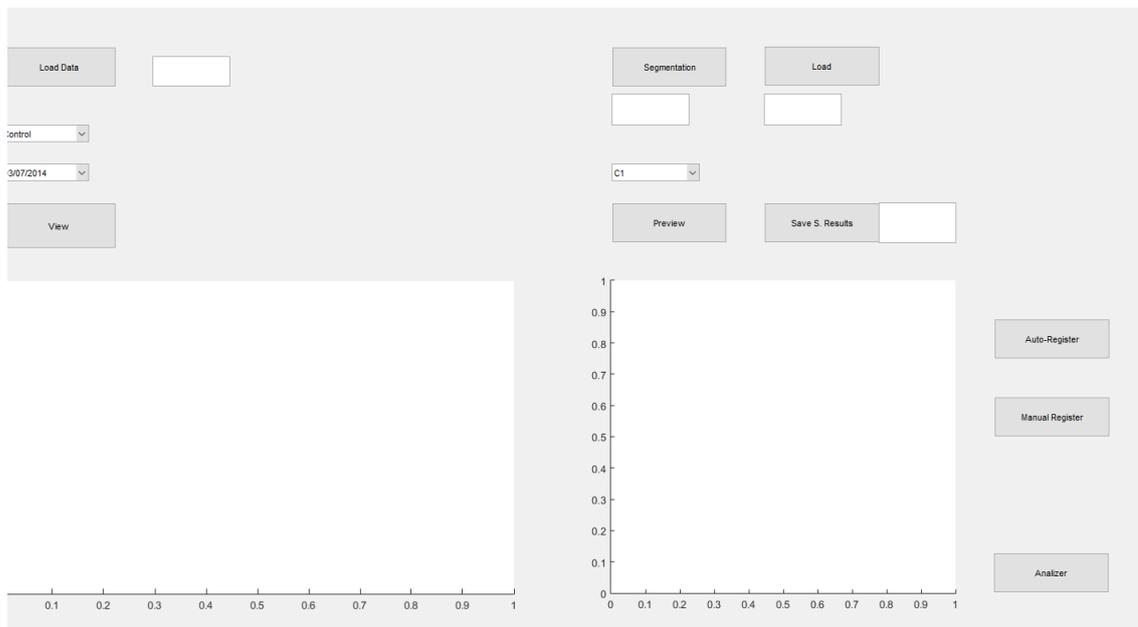


Figura 54. Pantalla de inicio de la interfaz de usuario

Inicialmente sólo se podrán realizar dos opciones:

1. Con el botón “Load Data”, se selecciona el directorio en el cual se encuentran las imágenes de las diferentes tomas y se organizan todas las tomas.
2. Con el botón “Load” situado a la derecha, se pueden cargar los resultados obtenidos de sesiones anteriores (ya que los procesos de corre registro conllevan un tiempo considerable). En primer lugar, se pedirán los resultados de control y, acto seguido, los de test.

El resto de opciones no se encontrarán disponibles hasta que no exista, al menos, cualquiera de las dos cargas de datos.

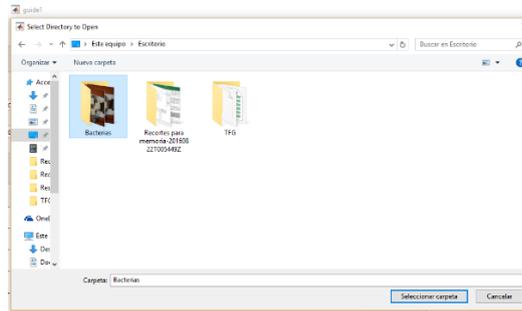


Figura 55. Carga del directorio donde se encuentran las imágenes base

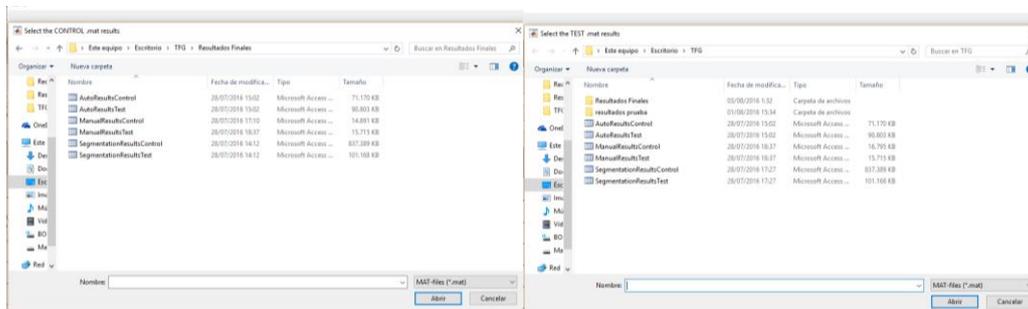


Figura 56. Carga de los diferentes estados de los archivos ya procesados

Tanto para las acciones de carga de datos o resultados, como para las de procesado de los datos, existe un indicador que nos avisa en qué momento el proceso está en ejecución y cuándo lo ha finalizado. Entre tanto, no será posible realizar ninguna otra opción.

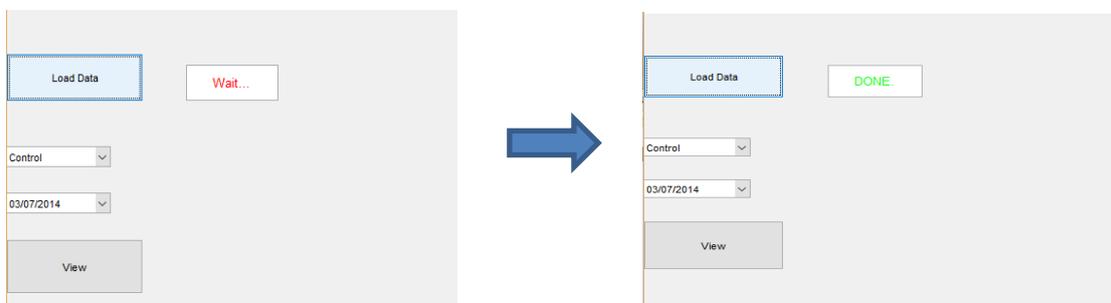


Figura 57. Cambio de estado del indicador de carga de datos

Una vez realizado la primera de las dos opciones, se puede realizar cualquiera de las acciones de procesado de imagen. Asimismo, también se puede visualizar cualquiera de las tomas que acaban de ser organizadas según fecha, probeta y ensayo. Para ello se selecciona en los menús desplegables qué probeta, qué ensayo y qué fecha, y se pulsa el botón “View”.

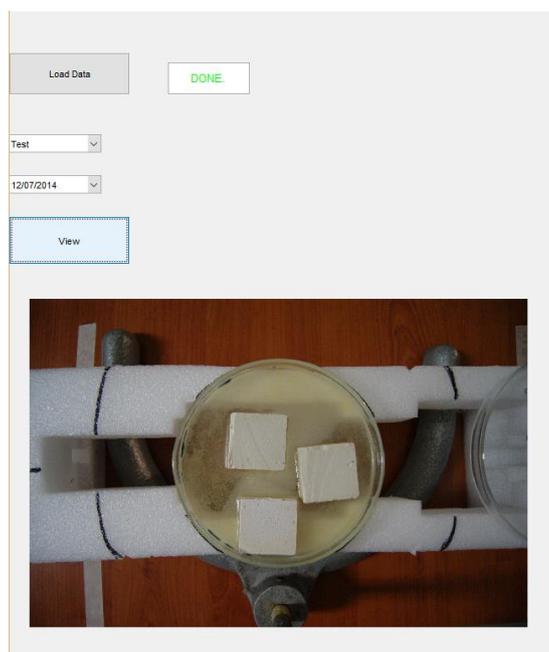


Figura 58. Ejemplo de visualización de una de las imágenes base una vez organizadas en la estructura

A partir de este punto, la siguiente opción que se encuentra disponible es la de realizar la segmentación de las imágenes, cargadas mediante la opción “Load Data”.

Finalizado este proceso, ya existen unas estructuras de resultados generadas que se pueden visualizar con la opción “Preview”, utilizando los mismos menús desplegables del paso anterior:

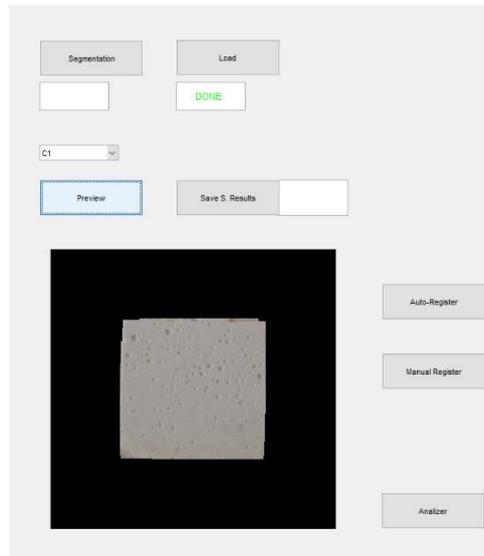


Figura 59. Visualización del resultado de la segmentación de imágenes

Si el usuario encuentra correcto el resultado obtenido, tiene la opción de guardar el resultado con el botón “Save S. Results” y también proceder directamente con el análisis del área de ocupación.

En caso de que la segmentación no consiga arreglar completamente los datos, una nueva opción aparece disponible en la interfaz. Se trata del corrección automática. Al pulsar sobre “Auto-Register”, se abre una nueva ventana en la que se debe pulsar el botón para que comience el registro. Una vez finalizado, existe la posibilidad de visualizar el resultado, en función de la probeta y el ensayo, obteniéndose una figura con todas las tomas ordenadas por fecha.

Si el resultado obtenido es correcto, se podrá guardar el resultado con el botón “Save Auto-Reg Results”, por si se desea continuar con el análisis en otro momento, y pasar directamente al análisis de área de ocupación con el botón “Proceed to Analyze”.



Figura 60. Pantalla donde se puede realizar el corregistro automático

En el caso de que los resultados continúen parcialmente incorrectos, existe la posibilidad tanto en el mismo momento (con el botón “Manual Register” de la pantalla que se observa en la figura 56 como en una sesión posterior cargando los resultados del registro automático en la pantalla de inicio, de realizar la corrección manual de las tomas que continúen mal ajustadas.

Este proceso permite, en primer lugar, previsualizar los casos en los que se observó un error en la corrección automática, con la contraposición de la primera toma de cada probeta de cada ensayo, y la toma seleccionada de la misma con los menús desplegables.

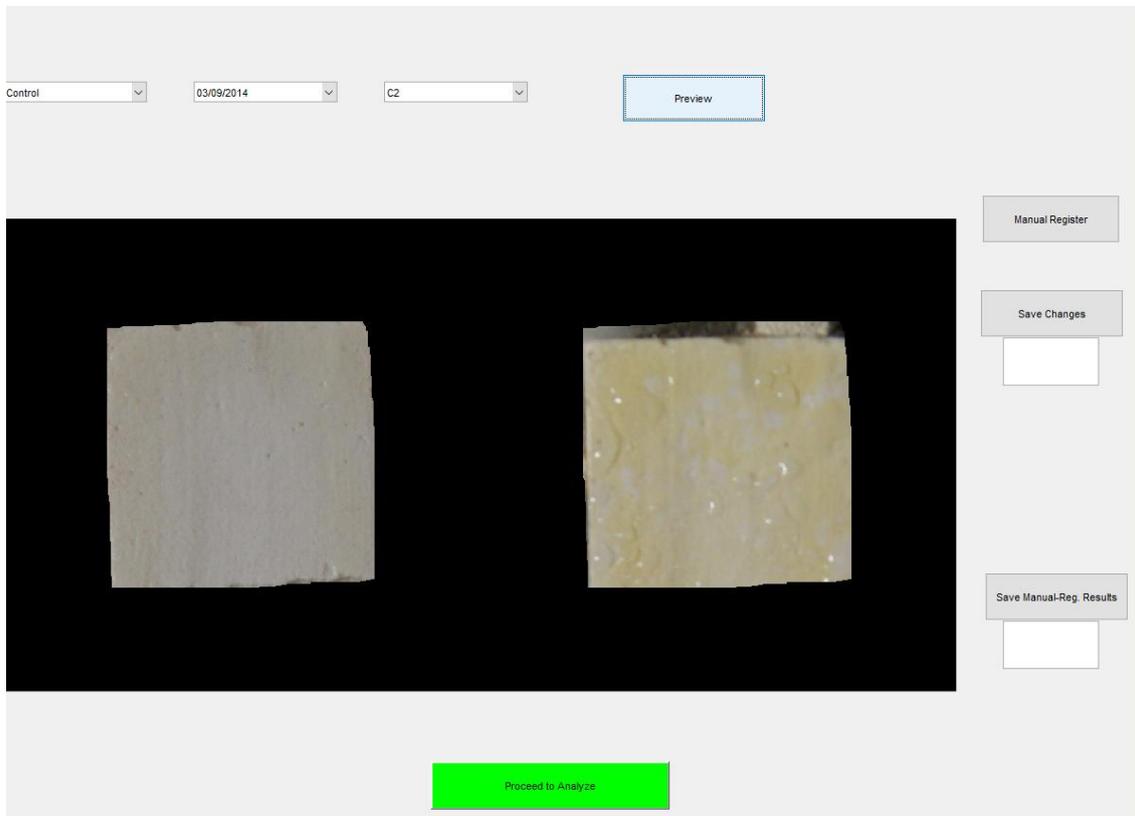


Figura 61. Pantalla de corregistro manual. Visualización de las imágenes que se desean corregir

Al pulsar el botón “Manual Register”, se abre la pantalla de selección de puntos para que el usuario marque los cuatro que considere más relevantes, tanto en la que se pretende corregir como en la de referencia. Para este proyecto se recomienda marcar con precisión las cuatro esquinas de las probetas para facilitar la tarea al algoritmo.

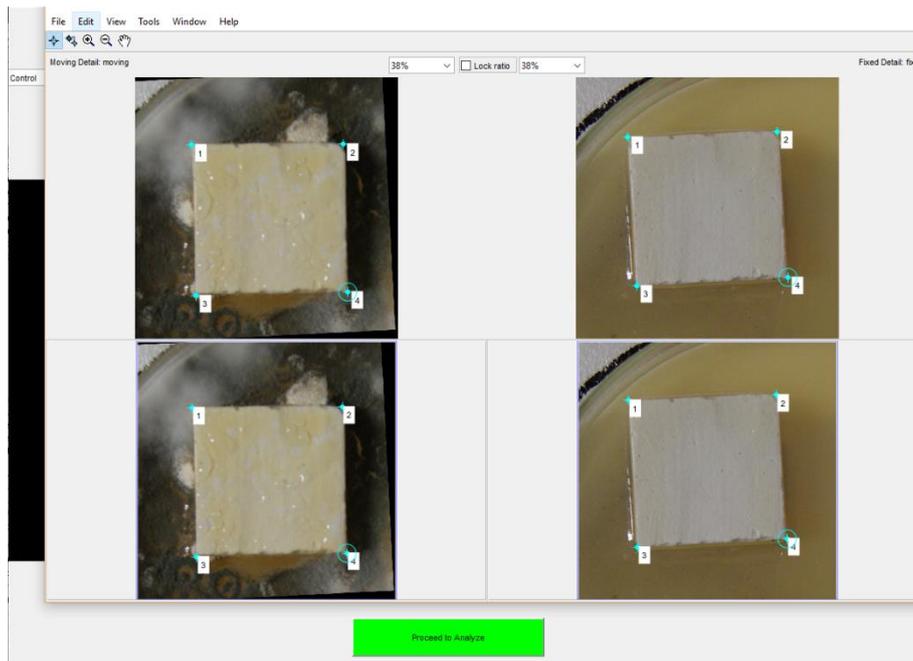


Figura 62. Selección de puntos de interés para el correjistro manual

Una vez terminado este proceso, el usuario podrá visualizar directamente el resultado del registro manual donde antes se visualizaba la comparación de tomas. La figura representará ambas tomas (la de referencia y la corregida) superpuestas, para comprobar que ambas coinciden en el espacio. Si el resultado es correcto, se debe guardar el caso concreto pulsando el botón “Save Changes”, antes de pasar a una nueva toma incorrecta.

Una vez corregidos todos los casos deseados, se podrá pulsar el botón “Proceed to Analyze” para pasar directamente al análisis de área de ocupación, y también guardar los resultados de la corrección manual para continuar con el análisis en una sesión posterior (ya que los procesos de corrección son computacionalmente costosos).

Cabe decir que, al existir la posibilidad de guardar los resultados después de aplicar cada uno de los procesos de segmentación y correjistro, se puede proceder directamente al análisis de color siempre y cuando no se parta del resultado de la carga del directorio (sin haberse realizado, al menos, la segmentación).

Una vez se procede con el análisis, desde cualquier de las pantallas vistas anterior mente, se abre una nueva en la que se presentan dos opciones de análisis:

1. Análisis Completo
2. Análisis Individual



Figura 63. Pantalla de selección del tipo de análisis colorimétrico

ANÁLISIS COMPLETO

Este modo se encarga de realizar un análisis completo del área de ocupación del microorganismo, en todas las probetas de ambos ensayos, y organizar los resultados en gráficas que faciliten la interpretación de los mismos. Así mismo dar la posibilidad de exportar los datos finales (tanto las gráficas en formato “.png” como los valores en formato de tabla Excel) para que el usuario pueda realizar un informe sin preocuparse de organizar y generar de nuevo los datos obtenidos.

Para poder estimar el área de ocupación, sería necesario comprobar primero de qué punto se parte, es decir, sabiendo cómo se encontraba la probeta en la primera toma (en cuanto a color se refiere), se podrán establecer unas pautas de actuación. Es por ello que la interfaz permite seleccionar dos imágenes para comparar los histogramas de las mismas. Estos histogramas serán, tanto los de tono (H) para encontrar el color buscado, como los de saturación (S) del tono en cuestión.

Llegado a este punto, el usuario podrá seleccionar el valor del umbral máximo y mínimo para el tono y el valor del umbral mínimo para la saturación.

Se decidió realizar la interfaz de esta manera ya que esto otorgaría versatilidad al análisis. Si por ejemplo, se quisiera realizar el análisis en función a otro color, modificar los umbrales mínimo y máximo del tono permitiría hacerlo.

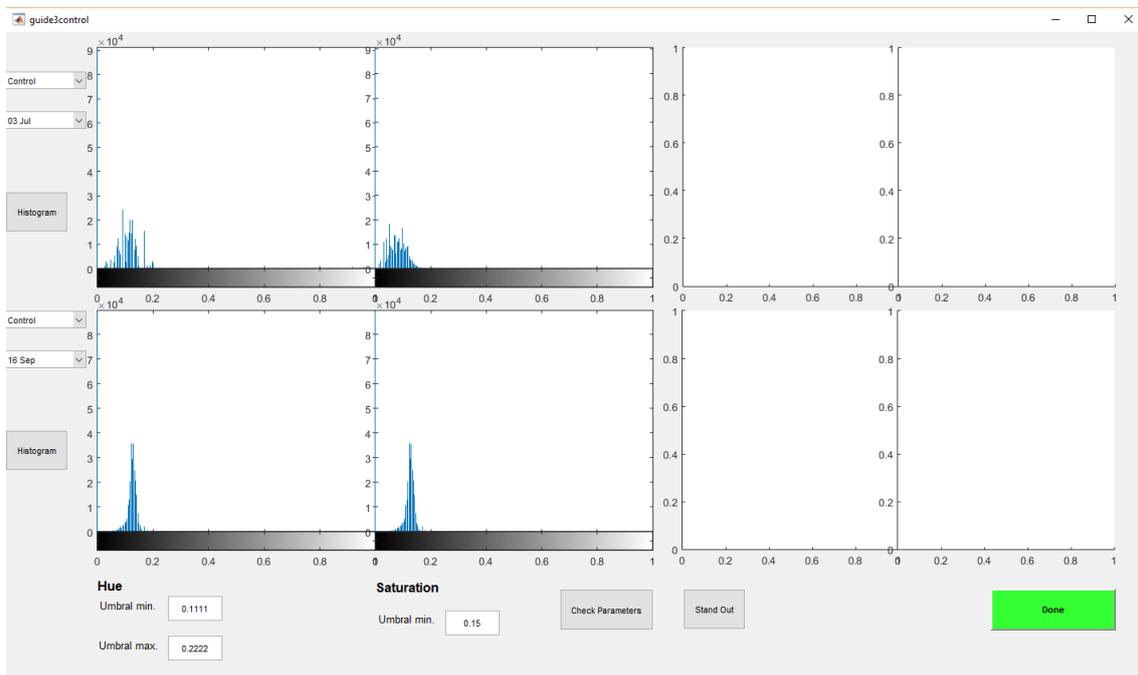


Figura 64. Análisis completo. Selección de parámetros umbral para tono y saturación

Para que el usuario tenga una guía del rango en el que se puede encontrar los valores óptimos de los parámetros, se establecen unos valores iniciales que previamente se han considerado correctos.

Pulsando el botón "Check Parameters" se puede observar, a la izquierda, las imágenes de las probetas seleccionadas y, a la derecha, las mismas imágenes multiplicadas por la máscara que se obtendría con esos parámetros, es decir, los píxeles resaltados en verde serían los que se tomarían como "ocupados".

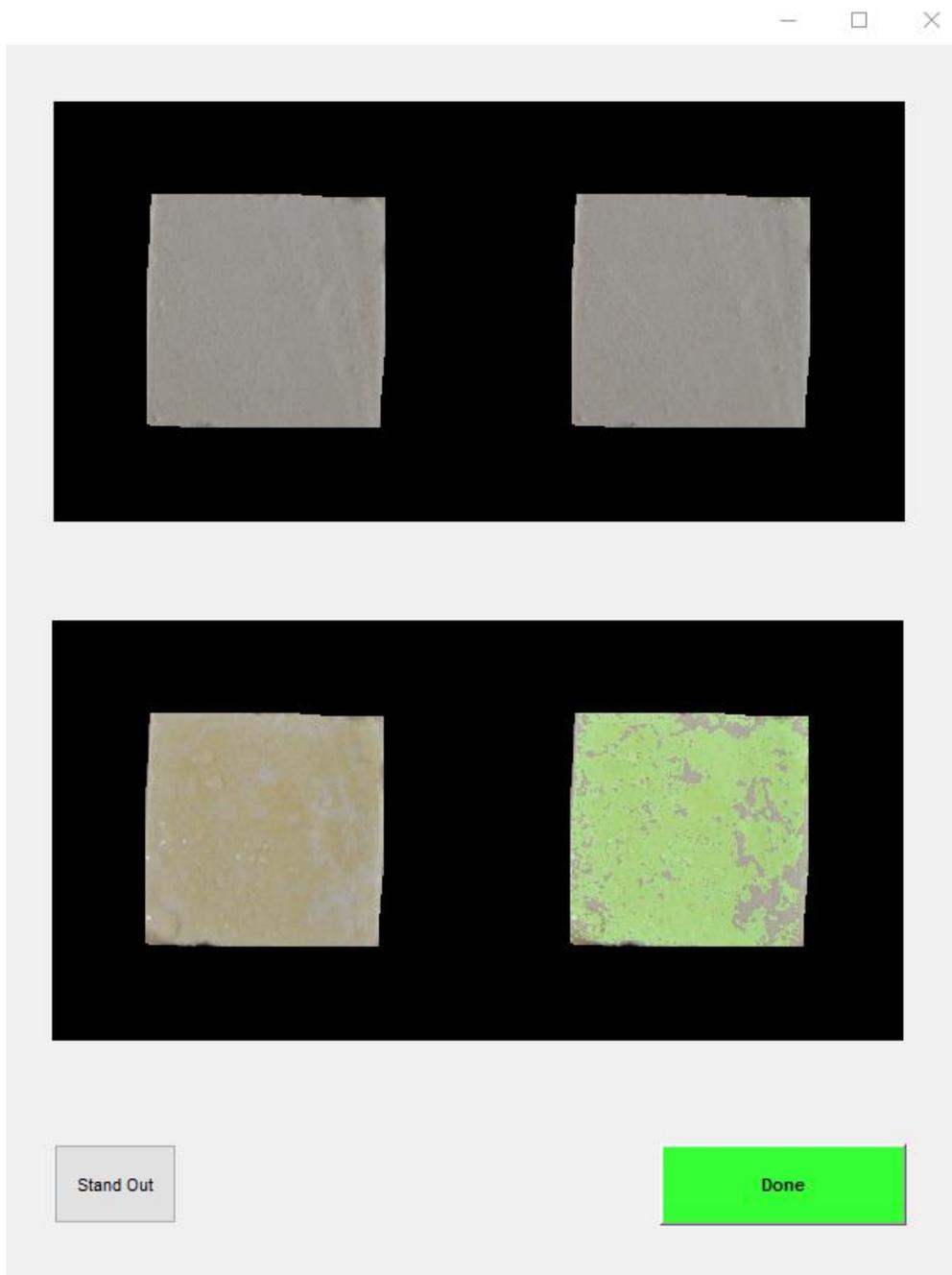


Figura 65. Previsualización de la aplicación de los parámetros seleccionados

Realizando este proceso, se observa que, en algunos casos, no resulta tan obvia la veracidad del resultado como en la figura anterior. Esto es debido a que el área de ocupación no se hace claramente visible para el ojo humano hasta las últimas tomas. Es por ello que se añade una nueva función para poder resaltar la saturación de los píxeles de la imagen. A continuación se puede observar un ejemplo del resultado obtenido al pulsar el botón "Stand Out".

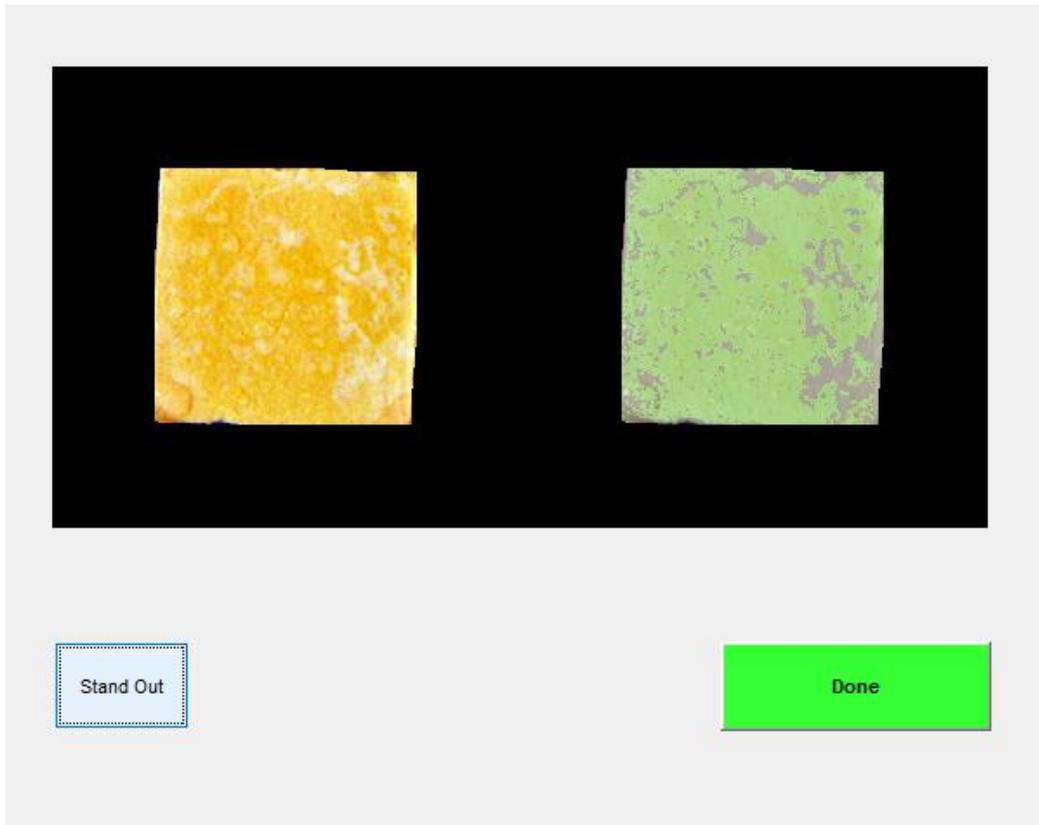


Figura 66. Función del botón "Stand Out"

Pulsando este botón podemos alternar entre ambos modos de visualización de las probetas ya que no en todas se visualizará mejor realizando esta acción.

Una vez se encuentren los valores que se consideren óptimos para el análisis, se podrá pulsar el botón "Done" para que el programa comience a aplicar el proceso de estimación de área en función del valor de los parámetros seleccionados.

COMPARACIÓN DE RESULTADOS

Llegado a este punto, se puede proceder a la comparación de los resultados calculados. Estos resultados consisten básicamente en el recuento del número de píxeles que se encontraban a uno en las máscaras generadas en función de los parámetros de saturación y tono. Esto proporcionará el resultado en número de píxeles, por lo que, además, se pasará a tanto por ciento ya que lo que interesa es los píxeles ocupados en función del total de ellos.

La comparación se realiza en tres gráficas, una para cada probeta, donde se encuentran los datos organizados por fecha de la toma y representados en forma de barras verticales; se consideró que éste era el tipo de gráfico en el que se podría concluir con mayor claridad. Pese a que la diferencia temporal no es simétrica entre los diferentes datos de cada serie, se decide colocarlos de forma equidistante para favorecer la interpretación.

Pulsando el botón "Compare" se obtiene lo siguiente:

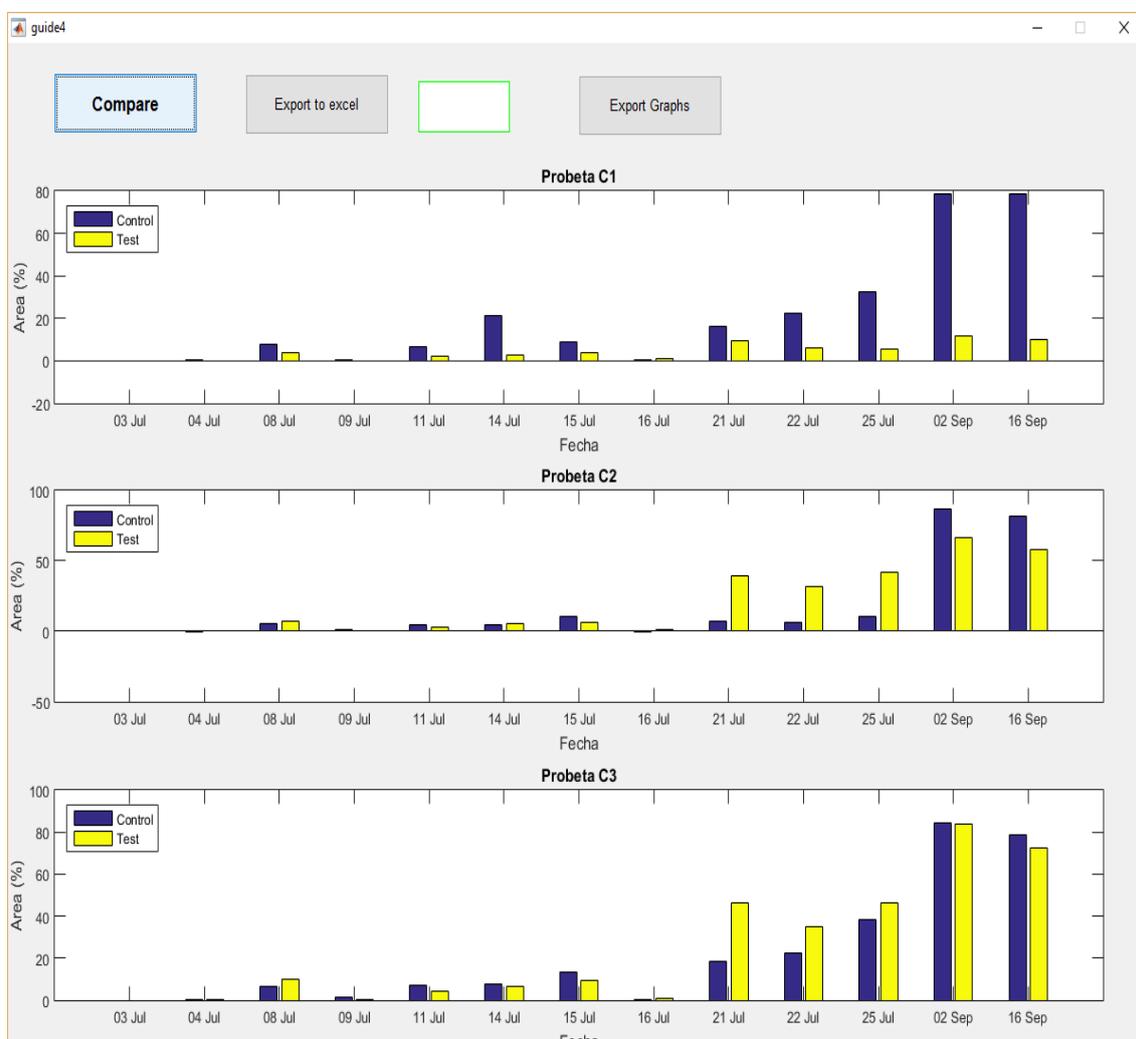


Figura 67. Visualización gráfica de los resultados obtenidos

Además, existe la posibilidad de exportar los resultados obtenidos en formato de hoja Excel pulsando el botón "Export to Excel". Así el usuario tendrá acceso a los valores exactos, tanto de las áreas de ocupación en cada uno de los ensayos, como la diferencia de ocupación entre el ensayo de control y el del test del biocida Biotin R.

Cuando el indicador pase a color verde, se guardará un archivo Excel llamado "Resultados" que se localizará en la misma carpeta del programa.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Area Ctrl (%)	03/09/2016	04/09/2016	08/07/2014	10/07/2014	12/07/2014	14/07/2014	15/07/2014	17/07/2014	21/07/2014	22/07/2014	25/07/2014	03/09/2014	16/09/2014	
2	Probeta C1	0	0,60845021	7,81880297	0,75531751	6,84694528	21,2925905	8,6623992	0,65872554	16,1846965	22,6595252	32,4612346	78,4485113	78,494828	
3	Probeta C2	0	-0,21373975	5,27844249	1,06018013	4,49744054	4,79404317	10,782241	-0,31944799	7,36821318	6,71653927	10,6618188	86,7756275	81,7349317	
4	Probeta C3	0	0,14630635	6,60608384	1,2666844	6,97850001	7,83756083	13,5701097	0,35755082	18,3051152	22,3359726	38,3146604	84,1437558	78,5320857	
7	Area Test (%)	03/09/2016	04/09/2016	08/07/2014	10/07/2014	12/07/2014	14/07/2014	15/07/2014	17/07/2014	21/07/2014	22/07/2014	25/07/2014	03/09/2014	16/09/2014	
8	Probeta C1	0	-0,10611821	3,60801918	0,13322034	2,3639931	2,96901959	3,96645443	0,94590262	9,71287008	6,20333471	5,56128136	12,0157879	9,84494526	
9	Probeta C2	0	0,20514013	7,49867803	0,19988974	2,94846745	5,28076446	6,77187443	1,01182462	39,470911	31,525575	41,5751912	66,5535333	57,6616275	
10	Probeta C3	0	0,30574822	9,78394314	0,36791442	4,194459	6,51415747	9,36598297	0,94031263	46,0761008	34,8963506	46,0409124	83,7155838	72,2617549	
13	Dif. (%)	03/09/2016	04/09/2016	08/07/2014	10/07/2014	12/07/2014	14/07/2014	15/07/2014	17/07/2014	21/07/2014	22/07/2014	25/07/2014	03/09/2014	16/09/2014	
14	Probeta C1	0	-0,71456842	-4,21078379	-0,62209716	-4,48295218	-18,3235709	-4,69594477	0,28717708	-6,47182643	-16,4561905	-26,8999532	-66,4327234	-68,6498827	
15	Probeta C2	0	0,41887988	2,22023554	-0,86029039	-1,54897309	0,48672129	-4,01036659	1,33127262	32,1026978	24,8090357	30,9133724	-20,2220942	-24,0733042	
16	Probeta C3	0	0,15944187	3,17785929	-0,89876998	-2,78404101	-1,32340336	-4,20412672	0,58276181	27,7709856	12,560378	7,72625203	-0,42817199	-6,2703308	

Figura 68. Hoja excel generada a partir de los resultados obtenidos

Además de los datos, es posible que el usuario desee exportar las imágenes de las gráficas para incluirlas en un informe de conclusiones del proyecto. Para ello se añade un botón llamado "Export Graphs" el cual, al pulsarlo, permite seleccionar el nombre del archivo que se va a guardar, que será uno por gráfica.

Observando los resultados obtenidos, se plantea la posibilidad de que el análisis completo no obtenga los resultados esperados para todas las tomas. Esto puede ser debido a las condiciones de captura de las fotografías, las cuáles pueden haber sido tomadas con diferente luz.

Para ello existe otro tipo de análisis, que permite modificar los parámetros de las tomas individuales elegidas para obtener el porcentaje de área ocupada.

ANÁLISIS DISCRETO

Para obtener el área de ocupación de una única probeta en una fecha concreta, existe el tipo de análisis discreto. En él se puede seleccionar el ensayo, la fecha y la probeta en cuestión utilizando los menús desplegables, y pulsando el botón “View Selected”, se puede visualizar la probeta elegida.

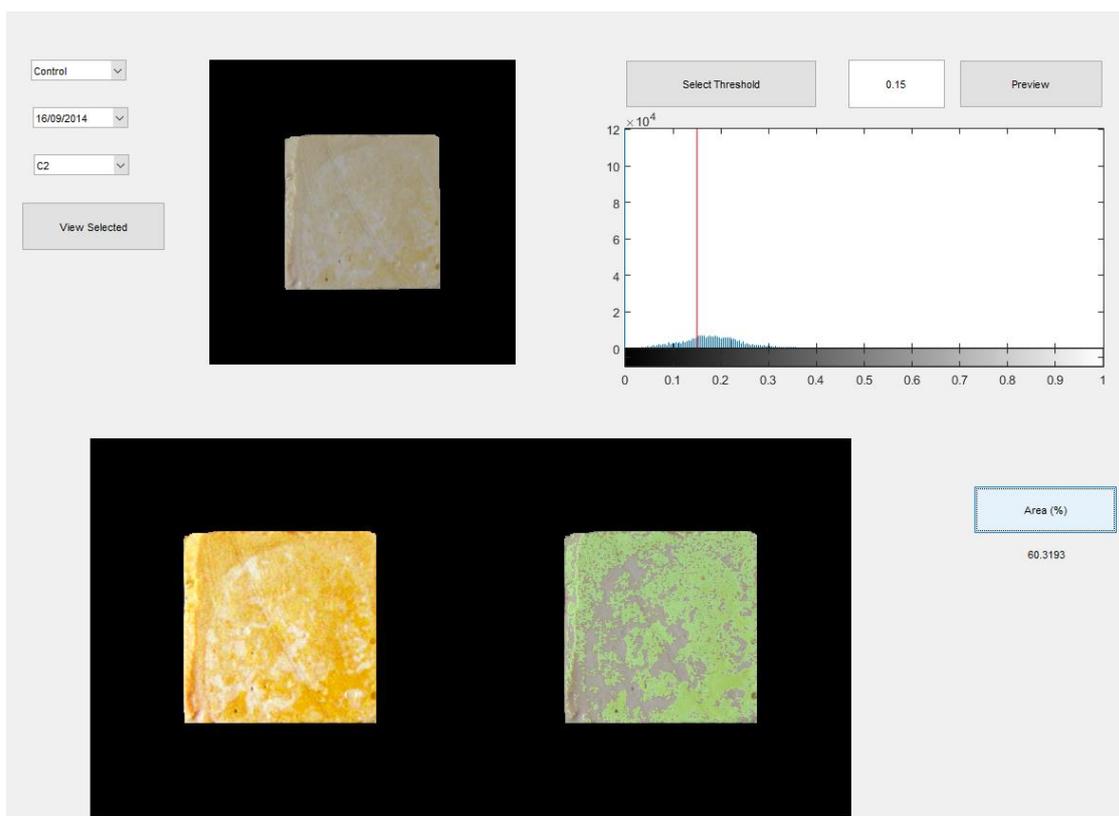


Figura 69. Análisis discreto. Selección del parámetro umbral de saturación

Pulsando el botón “Select Threshold”, aparece el histograma de saturación del tono y, cliqueando sobre el mismo, se puede seleccionar el umbral mínimo de saturación, visualizándose la máscara resultado que se obtendría en comparación con la imagen original.

Si la estimación sigue sin ser la deseada, basándose en el valor del indicador, el usuario puede introducir el valor a mano después de cinco intentos.

Una vez seleccionado el valor considerado como óptimo, se pulsa sobre el botón “Area (%)” para que el algoritmo aplique los parámetros a la imagen en cuestión. Así se obtiene el valor del área de ocupación, en tanto por ciento, para la probeta en concreto seleccionada. El usuario podrá introducir este nuevo valor, modificando la tabla Excel anteriormente generada por el análisis completo.

5. CONCLUSIONES

El presente Proyecto Final de Grado nace de la necesidad detectada, por de la Dra. Pilar Bosh y del grupo de trabajo del laboratorio de Microbiología Ambientale e Biorestauro (MicroLAB) del Dept. di Biocienze e Territorio (DiBT) de la Università degli Studi del Molise, sede de Pesche (Isernia) Italia, de estimar el área ocupada por microorganismos (bacterias) mediante procesado digital de imágenes (procesado morfológico y colorimétrico) de forma automática.

El desarrollo del trabajo está centrado en la obtención de datos fiables y lo más exactos en la medida de lo posible a la hora de estimar el porcentaje de área ocupada por bacterias, de cada una de las probetas y en cada una de las fechas en las que se realiza el control fotográfico de los ensayos. Para ello se requería un estudio de los conceptos más importantes del tratamiento de imagen y de las diferentes técnicas de procesado.

Debido a la variación asimétrica de la posición para cada una de las probetas, en un primer lugar se pretende realizar la detección de las mismas de forma independiente en cada una de las imágenes, sin embargo, aparecen dos problemas importantes: por un lado, la modificación del entorno de los objetos del interés, lo que prácticamente imposibilita la detección de las probetas en la mayoría de las últimas tomas, y el gran coste computacional que supondría este proceso. Es por ello que se observó la necesidad del uso de técnicas de corrección, para conseguir hacer coincidir de forma automática las imágenes de cada serie, y así, poder realizar sobre las mismas un estudio considerado fiable.

Una vez que el procesado de imagen hace su trabajo correctamente, la aplicación permite un análisis colorimétrico cómodo para el usuario, simplificando las diferentes opciones que existen para llevarlo a cabo y, a su vez, dotándolo de versatilidad para no limitarlo únicamente a un color ni a un grado de saturación concreto.

El programa se ha realizado pensando en facilitar el trabajo al usuario, de manera que, pese a que el programa esté diseñado de forma que sea versátil, existen unos valores predefinidos en todas las decisiones y parámetros, así como para la carpeta donde se guardarán los resultados y el nombre de los mismos.

Además de todo lo anterior, la muestra de los resultados obtenidos se realizó pensando en la claridad visual, de forma que la persona pudiera detectar rápidamente la eficacia del biocida sin tener que comprobar los valores numéricos uno a uno. Como complemento se realizó una herramienta que permitiera organizar y exportar los resultados obtenidos de forma que fuese útil para la persona que fuera a realizar el posterior informe y no obligarla pues a realizarlos de forma manual.

Finalmente, a partir de lo obtenido, se puede concluir que el funcionamiento de este programa responde perfectamente a las necesidades planteadas en los objetivos de este proyecto.

6. LINEAS FUTURAS

Los objetivos de este proyecto han sido cubiertos con éxito con el desarrollo del mismo y la aplicación responde correctamente a la primera de las premisas demandadas: el análisis del área de ocupación. Sin embargo, resulta inevitable que durante el proceso de desarrollo existan posibilidades complementarias que hubieran ayudado a mejorar esta herramienta y que, por falta de tiempo y la necesidad de ceñirse a unas pautas determinadas se han tenido que dejar a un lado. Esto abre las puertas a retomar este trabajo con el fin de transformarlo en una herramienta más eficiente y versátil.

Por ello, a continuación, se exponen algunas de las mejoras que se podrían explorar en futuras versiones o ampliaciones de la aplicación realizada:

- Optimizar el coste computacional de los procesos de tratamiento de imagen y corrección, los cuales suponen el 80% del tiempo de obtención de resultados correctos.
- Trasladar el código a otro lenguaje más potente como podría ser Java o C para optimizar los recursos hardware del ordenador.
- Profundizar más en el corrección automático de imagen para intentar prescindir de la herramienta de corrección manual y dejarlo únicamente como herramienta opcional para cualquiera de los casos.
- Para otorgarle mayor versatilidad, ampliar la aplicación de forma que admita distintos objetos de interés, como podría ser el número de objetos a identificar, el tipo de análisis dependiendo de si son bacterias, hongos o algas, u otorgarle la posibilidad de realizar un análisis tanto de porcentaje de ocupación de píxeles ocupados, como del grado de saturación de los mismos.
- Como propuesta de mejora para la toma de datos iniciales, es decir, del control fotográfico, se plantea que las imágenes se realicen simplificando la variación de los elementos de interés de una toma a otra. Esto puede ser: reducir el número de probetas, eliminar las modificaciones del entorno de las mismas o incluso fijar las probetas a la placa de Petri para evitar desplazamientos.

7. REFERENCIAS BIBLIOGRÁFICAS

[1] Dra Pilar Bosh Roig, Prof. Giancarlo Ranalli: Informe de resultados de la evaluación de la eficacia microbiológica del biocida Biotin R encapsulado con nanopartículas y aplicado sobre probetas de cemento y yeso, 2014.

[2]<http://www.monografias.com/trabajos102/morfologia-matematica-utilizando-matlab/morfologia-matematica-utilizando-matlab.shtml>

[3]Francisco Gabriel Ortiz Zamora: Procesamiento morfológico de imágenes en color. Aplicación a la reconstrucción geodésica, 2002.

[4]<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Imagenyvideo/morfologico/practica.htm>

[5] <http://dmi.uib.es/~catalina/docencia/PDS/cap5.pdf>

[6] Patricia del Castillo Hernández: PFC Procesado de imagen angiográfica para la cuantificación del flujo sanguíneo cerebral, 2014.

[7] <http://www.ite.educacion.es/formacion/materiales/107/cd/imagen/imagen0104.html>

[8] http://turing.iimas.unam.mx/~elena/PDI-Mast/Tema_6_C.pdf

[9]<http://www.comunicacion-multimedia.info/2010/05/modos-o-modelos-de-color-hsb-o-hsv-y.html>

[10] https://www.ecured.cu/Modelo_HSV

[11] <http://www.forpas.us.es/aula/gimp/files/GIMPC7.pdf>