



## PROCESAMIENTO DE IMÁGENES BIOLÓGICAS

Memoria del proyecto de final de carrera correspondiente a los estudios de Ingeniería Superior en Informática presentado por Leandro Jiménez Rodríguez y dirigido por Javier Serrano García.

Bellaterra, Septiembre de 2008

El firmante, Javier Serrano García, profesor del Departamento de Telecomunicaciones e Ingeniería de Sistemas de la Universidad Autónoma de Barcelona

CERTIFICA:

Que la presente memoria ha sido realizada bajo su dirección por Leandro Jiménez Rodríguez

Bellaterra, Septiembre de 2008

---

Firmado: Javier Serrano García

*Me gustaría dedicar este proyecto a las personas más especiales.*  
*A mi hermana Paula, por estar siempre pendiente de mí y su interés en mis estudios.*  
*A mi amigo Rafa, por estar siempre en los momentos más difíciles.*  
*A mi amiga Ana D. y su familia, por su ayuda en todo momento y apoyo, ya que probablemente no estaría aquí sin su ayuda.*  
*A Héctor por su apoyo cuando lo necesité.*  
*A Ana G., Jorge, Javi Juan y David por toda su ayuda y buenos momentos durante estos últimos años.*  
*A mi tía Inma y mis primas, por alegrarme los días más nublados.*  
*A Javi y Sonia por su ayuda y por esos momentos que necesitaba despejarme.*  
*A mi padre y hermano por su ayuda.*  
*A toda mi familia por estar siempre ahí.*  
*Y en especial, me gustaría dedicarle este proyecto a mi madre y mi abuela porque fueron ellas que ante todo querían que estudiase.*



# Agradecimientos

La realización de este proyecto ha sido posible gracias a muchas personas, a mi tutor Javier Serrano por ayudarme durante la elaboración del proyecto, a Maribel Geli por ayudarme en el temario de biología y a Ricardo Toledo por orientarme cuando lo necesité. También me gustaría agradecerse a mi familia y amigos por todo su apoyo.

Gracias.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Presentación . . . . .	1
1.2. Motivación . . . . .	2
1.3. Objetivos . . . . .	3
1.4. Planificación del proyecto . . . . .	3
1.5. Costes . . . . .	5
1.6. Contenido de la memoria . . . . .	6
<b>2. El procesamiento de imágenes biológicas</b>	<b>7</b>
2.1. La célula y sus estados . . . . .	7
2.1.1. Concentración de la proteína . . . . .	7
2.1.2. Tipos de estado de la célula . . . . .	9
2.1.3. Septim ring . . . . .	10
2.2. Procesamiento de imágenes y librerías . . . . .	11
2.2.1. Procesamiento de imágenes . . . . .	11
2.2.2. Librerías . . . . .	11
<b>3. Requisitos y funcionalidad</b>	<b>15</b>
3.1. Requisitos funcionales . . . . .	15
3.2. Requisitos no funcionales . . . . .	16
<b>4. Localización y tracking de los puntos luminosos</b>	<b>17</b>
4.1. Imagen Integral . . . . .	17
4.2. Primer método . . . . .	18

4.2.1. Implementación . . . . .	21
4.3. Segundo método . . . . .	22
4.3.1. Pseudocódigo . . . . .	26
4.3.2. Implementación . . . . .	27
4.4. Tracking de los puntos . . . . .	27
4.4.1. Implementación . . . . .	28
<b>5. Detección de células y septim ring</b>	<b>29</b>
5.1. Detección de células . . . . .	29
5.1.1. Adaboost . . . . .	29
5.1.2. Características Haar-Like . . . . .	30
5.1.3. Entrenamiento . . . . .	30
5.1.4. Filtro de mediana . . . . .	31
5.1.5. Implementación . . . . .	31
5.2. Detección del septim ring . . . . .	32
<b>6. Aplicación</b>	<b>35</b>
<b>7. Evaluación de resultados</b>	<b>37</b>
7.1. Localización de los puntos luminosos . . . . .	37
7.1.1. Método 1 . . . . .	37
7.1.2. Método 2 . . . . .	38
7.1.3. Comparativa de los 2 métodos . . . . .	40
7.2. Localización de células y septim ring . . . . .	40
7.2.1. Evaluación de la localización de células . . . . .	41
7.2.2. Septim ring . . . . .	43
7.3. Tracking de los puntos . . . . .	44
7.4. Aplicación final . . . . .	44
<b>8. Conclusiones y líneas futuras</b>	<b>45</b>
8.1. Conclusiones . . . . .	45
8.2. Líneas futuras . . . . .	46



<b>A. Manual de usuario</b>	<b>47</b>
A.1. Instalación . . . . .	47
A.2. Funcionamiento . . . . .	47
A.3. Formato del fichero de salida . . . . .	48
A.4. Terminar aplicación . . . . .	49
<b>Bibliografía</b>	<b>51</b>



# Índice de figuras

2.1. Entorno y partes de la célula. . . . .	8
2.2. Muestra de célula y concentración de la proteína. . . . .	9
2.3. Distintos estadios celulares. . . . .	10
2.4. Septim ring. . . . .	10
4.1. Imagen integral [ATI]. . . . .	18
4.2. Cálculo del área de los píxeles colindantes. . . . .	20
4.3. Resultados del método 1. . . . .	21
4.4. Niveles de intensidad despues de aplicar Laplace. . . . .	23
4.5. Erosión y dilatación . . . . .	25
4.6. Original y resultado . . . . .	25
5.1. Filtro de mediana. . . . .	31
5.2. Casos a diferenciar para localización del septim ring. . . . .	33
6.1. Interfaz gráfica de la aplicación. . . . .	36
7.1. Problema del método 1 con el ruido. . . . .	38
7.2. Imágenes con ruido. . . . .	39
7.3. Problema de resolución. . . . .	39
7.4. Problema con células demasiado juntas. . . . .	43
A.1. Interfaz gráfica de la aplicación. . . . .	48



# Índice de cuadros

1.1. Planificación. . . . .	4
1.2. Costes. . . . .	5
1.3. Recursos. . . . .	5
3.1. Ejemplo de fichero de salida. . . . .	16
7.1. Resultados filtro método 1. . . . .	38
7.2. Resultados filtro método 2. . . . .	39
7.3. Resultados filtros para imágenes de distintos vídeos. . . . .	41
7.4. Resultados de diez vídeos. . . . .	42

# Capítulo 1

## Introducción

### 1.1. Presentación

En los inicios de la década de 1960, Osamu Shimomura fue la primera persona en aislar la proteína verde fluorescente o GFP (*Green Fluorescent Protein*). Esta proteína es producida por la medusa *Aequorea Victoria*, la cual emite bioluminiscencia dentro de la zona verde del espectro visible de la luz.

Tras este descubrimiento, a lo largo de las últimas décadas se ha desarrollado el uso de la GFP y sus formas modificadas se usan como biosensores (instrumento para la medición de parámetros biológicos) en células vivas. En concreto, la fusión de GFP a las distintas proteínas que componen una célula, permite visualizar mediante microscopia de fluorescencia, la localización subcelular de la misma y su nivel de expresión o concentración en el interior celular. El hecho de que se utilice tanto éste método es debido a que no es un método agresivo, es decir, que al introducir GFP en una célula, ésta no se ve alterada en ninguna de sus funciones. Hasta la fecha de hoy, muchas bacterias, células de hongos, plantas, aves y mamíferos, incluyendo humanas, se han manipulado genéticamente para que se expresen determinadas proteínas de interés fusionadas a GFP.

La disponibilidad de GFP y sus derivados ha modificado la forma de usar de la microscopía de fluorescencia en la biología celular añadiendo la dimensión temporal al sistema. La complejidad del procesamiento de las imágenes de mi-

microscopía de fluorescencia in vivo, es una de las principales causas de que se empiece a desarrollar sistemas informáticos para el procesamiento de imágenes de microscopía de fluorescencia, que pueden ser usados para mejorar la calidad de las imágenes su adquisición o su análisis posterior. El análisis de las células marcadas con proteínas fluorescentes ha redefinido la forma de entender muchos procesos biológicos que ya habían sido estudiados en el pasado.

El CSIC, Consejo Superior de Investigaciones Científicas, está haciendo estudios sobre el comportamiento in vivo de una proteína en distintos estadios del ciclo celular usando GFP. Los estudios se realizan sobre el incremento transitorio de concentración de dicha proteína en puntos del córtex celular. Para poder realizarlo, el gen que codifica dicha proteína se fusiona en pauta de lectura correcta con el gen que codifica GFP. El gen fusión se introduce entonces en el genoma de la célula, sustituyendo el gen original de manera que la proteína de interés se comporta ahora de igual manera que la proteína original en lo que se refiere a sus funciones celulares, pero emite además fluorescencia verde cuando se expone la célula a luz ultravioleta. Puesto que la cantidad de fluorescencia emitida en un punto es proporcional a la cantidad de moléculas de proteína fluorescente en dicho punto, el aumento de la concentración de esta proteína en una zona de la superficie celular produce la aparición de un punto luminoso en el interior de la célula.

Desde el punto de vista biológico tiene interés el análisis de la intensidad lumínica de los puntos así como el análisis del tiempo que perduran. También puede ser interesante comparar el comportamiento de los puntos lumínicos que pertenecen a células que se encuentra en un determinado estadio del ciclo celular.

## 1.2. Motivación

Todo este análisis lo suele hacer una persona y lo que el CSIC le interesaría es automatizar esta adquisición de datos y realizar un análisis mediante la implementación de una aplicación, ya que observar el vídeo hay que estar pausando y anotando los datos, pasar los datos al ordenador que puede que se produzca un error al introducir los datos y realizar los cálculos, todo esto conlleva mucho

tiempo. Automatizando este proceso se ahorraría tiempo y esfuerzo, permitiendo así el procesamiento de un mayor número de datos y en consecuencia la robustez estadística de las conclusiones derivadas.

### 1.3. Objetivos

El objetivo de este proyecto es la realización de una aplicación con interfaz gráfica para el CSIC capaz de automatizar el análisis de las concentraciones de una proteína, vista como puntos luminosos, dadas unas secuencias de vídeo donde se muestran las células previamente tratadas con el GFP.

Para alcanzar el objetivo de este proyecto, se ha dividido en 4 subobjetivos:

1. Búsqueda de los puntos luminosos.
2. Distinguir entre los tipos de estadio celular y localizar el *septim ring*.
3. Seguimiento en la secuencia de vídeo de los puntos luminosos.
4. Realizar la aplicación con una interfaz gráfica y el procesamiento de datos obtenidos anteriormente.

### 1.4. Planificación del proyecto

División del proyecto en fases:

1. Entrevista con la responsable del CSIC para establecer los requerimientos de la aplicación.
2. Estudio y planificación del proyecto.
3. Investigación para la localización de los puntos luminosos.
4. Implementación y testeo del algoritmo de localización de puntos luminosos.
5. Investigación para la localización de las células.



Tarea	Fecha inicio prevista	Fecha fin prevista	Fecha inicio real	Fecha fin real
Entrevista CSIC	19/10/2007	30/10/2007	19/10/2007	30/10/2007
Planificación	22/10/2007	02/11/2007	22/10/2007	02/11/2007
Investigación				
Puntos luminosos	03/11/2007	14/12/2007	03/11/2007	14/03/2008
Puntos luminosos 2	No prevista	No prevista	28/07/2008	14/08/2008
Localización células	16/01/2008	28/02/2008	15/03/2008	01/07/2008
Tracking	02/04/2008	15/04/2008	16/07/2008	27/07/2008
Implementación y test				
Puntos luminosos	15/12/2007	15/01/2008	03/01/2008	14/03/2008
Puntos luminosos 2	No prevista	No prevista	15/08/2008	20/08/2008
Localización células	01/03/2008	01/04/2008	11/05/2008	16/07/2008
Tracking	16/04/2008	01/05/2008	01/09/2008	03/09/2008
Aplicación y datos	02/05/2008	30/05/2008	10/09/2008	17/09/2008
Memoria y Documentación	07/04/2008	10/06/2008	01/06/2008	17/09/2008

Cuadro 1.1: Planificación.

6. Implementación y testeo del localizador de células.
7. Estudio para la realización del tracking de los puntos luminosos.
8. Implementación del tracking y testeo del tracking.
9. Realización interfaz gráfica junto al procesamiento de los datos obtenidos.
10. Testeo de la aplicación.
11. Realización de la memoria y documentación de la aplicación.

En el cuadro 1.1 se presenta las fechas para la planificación inicial y la real. Como se puede apreciar las fechas de la planificación inicial no se ha cumplido por una serie de motivos:

- Dificultad en encontrar información útil sobre la investigación para la localización de los puntos luminosos y células.
- Motivos laborales y de estudios que impidió la dedicación de más horas al proyecto durante los meses de curso.
- El nuevo conjunto de vídeos facilitado más tarde hizo que se tuviese que investigar otro método para la localización de los puntos.

Tarea	Encargado	Tiempo días	Tiempo horas	Precio €
Entrevista CSIC	Director [25 %] y Técnico [25 %]	8	32	544
Planificación	Director [20 %] y Técnico [30 %]	10	40	607,2
<b>Investigación</b>				
Puntos luminosos	Técnico [30 %]	80	192	2304
Puntos luminosos 2	Técnico	14	112	1344
Localización células	Técnico [30 %]	73	168	2016
Tracking	Técnico	7	56	672
<b>Implementación y test</b>				
Puntos luminosos	Técnico [70 %]	50	196	2352
Puntos luminosos 2	Técnico	5	40	480
Localización células	Técnico [70 %]	47	184,8	2217,6
Tracking	Técnico	3	24	288
Aplicación y datos	Técnico	7	56	672
Memoria y Documentación	Técnico [50 %]	75	130	1560
Proyecto entero		210	1230	15056,8

Cuadro 1.2: Costes.

Recurso	Remuneración €/hora
Técnico	12
Director	22

Cuadro 1.3: Recursos.

## 1.5. Costes

En esta sección se presentan los costes si se hubiese realizado el proyecto en una empresa, dado que no ha sido el caso, estos costes son hipotéticos. Para la realización de este proyecto se han establecido dos trabajadores, el director del proyecto, que es el encargado de supervisar el trabajo realizado y guiar al segundo trabajador, el técnico, que es el encargado del estudio y implementación del proyecto. En el cuadro 1.2 se encuentra el tiempo, en días y horas, y el porcentaje de dedicación de cada uno en las distintas tareas. También se muestra el coste de cada una de las tareas y el total del proyecto. Este coste sería el de un ingeniero informático y un director de proyectos. En el cuadro 1.3 se muestra el sueldo bruto por horas de cada uno.

## **1.6. Contenido de la memoria**

La memoria consta de ocho capítulos y un anexo. En el primero se ha mostrado la introducción, planificación del proyecto y sus posibles costes si se hubiese realizado en una empresa. El segundo capítulo hace una breve introducción de conceptos sobre biología y procesamiento de imágenes. En el tercer capítulo se exponen los requerimientos de la aplicación. En los capítulos cuatro, cinco y seis, explica la implementación realizada. En los capítulos siete se expone los resultados obtenidos. Y en el último capítulo se explica las conclusiones realizadas y las líneas futuras. Y para terminar, en el anexo se incluye la documentación de la aplicación.

## **Capítulo 2**

# **El procesamiento de imágenes biológicas**

En este capítulo se explicará algunos conceptos sobre las células a analizar, para así poder entender la petición del CSIC, y a continuación de esto también se explicará algunos conceptos sobre el procesamiento de imágenes y librerías de procesamiento de imágenes.

### **2.1. La célula y sus estados**

El primer sección se explicará los conceptos necesarios para entender lo que el CSIC necesita que la aplicación tenga en cuenta al realizar el análisis. Esta sección esta dividida en tres apartados: el primero de ellos es la concentración de la proteína, el segundo, los tipos de estados de la célula, y el tercero el septim ring. Antes de nada, es necesario aclarar las posibles dudas acerca de las partes que interesa de las células. Ver figura 2.1.

#### **2.1.1. Concentración de la proteína**

Como está explicado anteriormente, las células que se están analizando tienen añadida una variante de la proteína GFP. La GFP, o sus variantes, tiene la peculiaridad de que al ser iluminada por luz ultravioleta es capaz de emitir luz en una

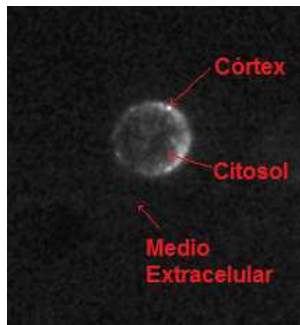


Figura 2.1: Entorno y partes de la célula.

longitud de onda que está dentro del espectro visible y que es captada por el ojo humano, y al ser una de las variantes de la GFP, la luz emitida es de color verde. Entonces, todos los vídeos que se analizarán tendrán tonalidades verdes, y al ser solo de esta gama, las imágenes podrán ser convertidas en escala de grises sin perder información útil y se mejorando el rendimiento del tratamiento de la imagen y su respectivo análisis, ya que solo tendremos que procesar un canal *GRAY* frente a los tres originales del vídeo *RED GREEN BLUE*.

Una vez aclaradas las características de esta proteína, se pasará a explicar cómo influye. Al ser luminiscente, la GFP puede ser usada de marcador de otras proteínas originales de la célula, es decir, podemos hacer que una proteína de la célula en cuestión llegue a emitir esa luz característica. Este hecho es útil para analizar los niveles de concentración de la proteína y estudiarla es mucho más simple de interpretar, ya que si la proteína emite luz a una determinada intensidad y si la proteína está dispersa se verá como un tono verdoso en toda la zona, pero si la proteína está concentrada se verá una zona más iluminada que el resto, tal y como se puede apreciar en la figura 2.2.

Estos niveles de concentración son variantes en el tiempo, lo que quiere decir que podrán aparecer puntos nuevos en distintas posiciones y también podrán desaparecer los puntos que ya estaban en un inicio, o los nuevos que aparecieron. La duración en la iluminación de los puntos se denomina tiempo de residencia de la proteína  $T_p$  y es un aspecto que se analizará más adelante.

Otra cosa a analizar son los niveles de intensidad lumínica, la intensidad lumí-

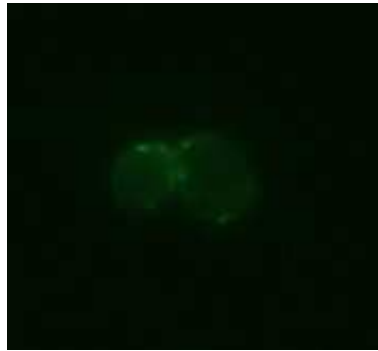


Figura 2.2: Muestra de célula y concentración de la proteína.

nica máxima del punto  $I_p$  y la intensidad lumínica del citosol que es proporcional a la concentración de la proteína disuelta en el interior celular.

Para adquirir todos estos datos, el primer paso es encontrar los puntos dentro de la imagen y una vez obtenidos, se podrá extraer e interpretar toda la información.

### 2.1.2. Tipos de estado de la célula

Como se ha comentado anteriormente, puede ser interesante distinguir entre tres estados celulares, es decir, hacer el proceso de detección de los puntos para luego clasificarlos según el estado que se encuentre esa célula o el tipo de célula que sea. Para ello ahora se describirá los tres estados que distinguiremos.

- *Unbudded* (UB): son las células que no tienen una célula hija. Figura 2.3 a).
- *Small Budded* (SB): son las que tienen una célula hija *SBd*, cuyo diámetro es inferior o igual a la mitad del diámetro de la célula madre *SBm*. Figura 2.3 b).
- *Big Budded* (BB): son las que tienen una célula hija *BBd* cuyo diámetro es superior a la mitad del diámetro de la célula madre *BBm*. Figura 2.3 c).

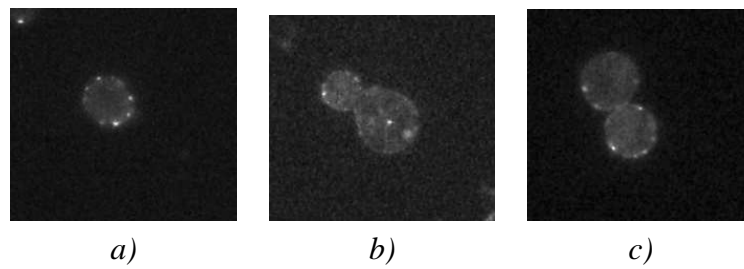


Figura 2.3: Distintos estadios celulares.



Figura 2.4: Septim ring.

### 2.1.3. Septim ring

El Septim ring es la zona donde las células madre e hija están unidas, por lo tanto, sólo se encuentra en dos estados de la célula, en las Small Budded y las Big Budded.

Para realizar la obtención de los puntos donde los niveles de concentración de la proteína son mayores, factor interesante a tener en cuenta para el estudio del CSIC, se debería poder separar entre los puntos encontrados en el septim ring y los que no se encuentran en él. Véase figura 2.4.

## **2.2. Procesamiento de imágenes y librerías**

En esta segunda sección se explicará algunos conceptos del procesamiento de imágenes y se analizarán por encima algunas librerías para el procesamiento de imágenes.

### **2.2.1. Procesamiento de imágenes**

A lo largo de las últimas décadas, el campo del procesamiento de imágenes ha sido objeto de investigación debido al incremento de dispositivos de captura de imágenes, su digitalización y la automatización de tareas que antes sólo podía realizar el ser humano. Gracias a esto se han desarrollado muchos métodos y algoritmos para el tratamiento y procesamiento de imágenes.

La evolución de las computadoras también ha contribuido en el desarrollo de estos algoritmos y métodos, ya que la computación de las imágenes siempre ha sido un proceso muy costoso, y la comercialización de computadoras más potentes a hecho posible el procesamiento de imágenes en tiempo razonable.

Debido a que hay muchos objetivos comunes en el procesamiento de imágenes, existen unas librerías que tienen implementados muchas de las operaciones más usadas.

### **2.2.2. Librerías**

Las librerías de procesamiento de imágenes facilitan la labor del programador que desee realizar una aplicación que requiera procesamiento de imágenes. Estas librerías pueden ser especializadas incluyendo algunos algoritmos específicos y/o operaciones más frecuentes. Como en este proyecto se analizarán y procesarán imágenes, para no tener que implementar todas las operaciones de cero se usará una librería de procesamiento de imágenes.

A continuación se expondrán algunas librerías para el procesamiento de imágenes y se comentarán brevemente.

Matlab: el programa Matlab [Mat] incluye unas estupendas librerías para el procesa-



miento de imágenes y de las más completas. Las librerías de Matlab incluyen las operaciones necesarias para la implementación del proyecto, como la carga de vídeos, la derivada de una matriz, convoluciones, conversión de una imagen de color a escala de grises y/o blanco y negro. Estas librerías se usarán para la fase de desarrollo y pruebas de los algoritmos usados en la aplicación. Pero éstas no pueden ser usados en la aplicación final, ya si se deseara usar estas librerías el CSIC debería pagar licencia para poder tener instalado el Matlab y poder hacer correr la aplicación, dado que no hay presupuesto para la elaboración de este proyecto se tiene que optar por alguna librería de libre distribución. Existen librerías especializadas para el tratamiento de imágenes biológicas para Matlab, pero nos encontramos con el mismo problema de la licencia [SVM].

CImg : son unas librerías bastante completas de libre distribución y para C++, incluyen las funcionalidades necesarias para cargar un vídeo, convertir una imagen a escala de grises y/o blanco y negro, convoluciones. Pero le faltan algunas funciones necesarias como la de derivar una imagen. Estas tampoco facilitan un metodo para realizar y usar un filtro Adaboost.

OpenCV: las OpenCV son unas de las librerías más completas que existen, incluyen funcionalidades como cargar un vídeo, derivar una imagen, convertir una imagen a escala de grises y/o blanco y negro, convoluciones, todas necesarias para la implementación de este proyecto. También incluyen funciones y una aplicación para la realización y uso de un filtro entrenado mediante Adaboost, que se explicará más tarde qué es. Además estas librerías son de código abierto, por lo tanto no tendrá el problema de licencia, otra ventaja que tienen es que son para lenguaje C++, por lo que se podrá elaborar una aplicación independiente que es lo que el CSIC desea. En cuanto a funcionalidad también son muy completas y poseen todas las operaciones necesarias para la elaboración del proyecto [OpCV].

Las librerías que se usarán para la elaboración del proyecto serán las OpenCV, porque cumplen con los requisitos de funcionalidad y son de licencia GPL, por

lo tanto son de libre distribución, también se escogieron estas librerías por ser en lenguaje C++ y por su variedad de funciones. Al ser en C++ se puede realizar una aplicación distribuible sin necesidad de pagar licencia. También se podría haber escogido para algunas funciones las CImg, pero ya que las OpenCV incluyen todas las funcionalidades necesarias, para no complicar el proceso de aprendizaje para implementar la aplicación, sólo se usarán las OpenCV.

Existen otras librerías, pero para usarlas la inmensa mayoría requieren pagar una licencia, dado que las OpenCV cumplen con todos los requisitos de funcionalidad para la realización del proyecto no ha sido necesario tener que adquirir unas de pago. También hay otras librerías de libre distribución, pero no son tan completas como las OpenCV o Matlab, por lo tanto fueron descartadas.

Para concluir este capítulo se repasarán los datos de los vídeos a obtener. Lo primero que se debe obtener son los puntos donde los niveles de concentración de la proteína son mayores y medir el tiempo de residencia. Lo segundo sería identificar el estado de la célula para luego poder separar entre los estados de la célula. Por último, decir si los puntos están en la zona del septim ring o no. Y para la realización de la aplicación final, se usará las librerías OpenCV por los motivos explicados anteriormente.



## Capítulo 3

### Requisitos y funcionalidad

Lo que se desea es crear una aplicación capaz de automatizar el análisis de este tipo de células. Lo que interesa analizar es la intensidad lumínica de los puntos (que es proporcional a la concentración de la proteína en ese punto), el análisis del tiempo que perduran los puntos (que se define como tiempo de residencia de la proteína),  $Tp$ . Respecto a la intensidad lumínica se medirá la intensidad máxima del punto  $Ip$ .

El programa se utilizará para adquirir automáticamente la información sobre lo puntos luminosos descrita anteriormente partiendo de un conjunto de vídeos definidos por el usuario en los que se habrán filmado una o más células en las que aparecen y desaparecen puntos luminosos. Para cada vídeo el programa devolverá información sobre la media de los parámetros  $Ip$ ,  $Tp$ . El programa deberá distinguir a que tipo de célula pertenece el punto hallado,  $UB$ ,  $SBm$ ,  $SBd$ ,  $BBm$ ,  $BBd$  donde las terminaciones indican  $m$  es la célula madre y la  $d$  es la hija, o si pertenece al septim ring  $SR$  o no pertenece al septim ring  $NSR$ .

#### 3.1. Requisitos funcionales

El usuario especificará el vídeo que desee analizar.

El análisis consiste en medir el tiempo en que la proteína se concentra en un punto, tiempo de residencia.

Coordenadas	Tiempo residencia	Intensidad máx.	Tipo célula	Posición
90,59	3	140	BBm	SR
73,142	1	132	SBd	NSR
...	...	...	...	...
	Media	Media		

Cuadro 3.1: Ejemplo de fichero de salida.

Hay que asignar a cada punto un identificador, como por ejemplo un número o mediante sus coordenadas en el frame del vídeo.

Los resultados deben mostrar si se desea el estado de la célula:

- UB
- SB y distinguir la M mother o D daughter.
- BB y distinguir la M mother o D daughter.

Y también donde se encuentre la concentración:

- Si se encuentra en septim ring, SR.
- No se encuentra en septim ring, NSR.

La información de salida debe ser por fichero, en el cual debe estar distribuido de la forma que muestra el ejemplo del cuadro 3.1.

El usuario podrá decidir qué información, de la comentada anteriormente, se debe mostrar en el fichero de salida.

## 3.2. Requisitos no funcionales

La aplicación debe funcionar bajo plataformas Windows™, ya que en los laboratorios del CSIC la maquinaria usa Windows XP™, también debe ser una aplicación que sea de fácil distribución y *user friendly*, ya que se desea distribuir entre los ordenadores del laboratorio sin tener que realizar una instalación compleja y ha de ser usado por el personal del laboratorio fácil y intuitivamente. Además para la elaboración del proyecto no hay presupuestos por lo tanto la aplicación debe ser libre en costes de licencia.

## Capítulo 4

# Localización y tracking de los puntos luminosos

Cuando se habla de la localización de puntos luminosos, lo que se desea obtener son las coordenadas del centroide del conjunto de píxeles que forman parte del punto luminoso. Con las coordenadas se podrá saber si un punto es el mismo en el siguiente frame del vídeo y si se encuentra en el septim ring. Para la obtención de los puntos, se tratará por separado frame a frame del vídeo que se proporcione a la aplicación.

Las imágenes obtenidas de los vídeos son en RGB, es decir, con tres canales distintos. El primer paso que se realizará siempre es convertir la imagen en una de escalas de grises para así tener solo un canal a tratar.

### 4.1. Imagen Integral

Antes de explicar el primer algoritmo se debe explicar la Imagen Integral, de *Viola and Jones* [VioJo], que es un método para realizar de forma rápida el cálculo del nivel medio de intensidad de las regiones de píxeles en una imagen o matriz. Ésta consiste en crear una matriz del mismo tamaño que la matriz de la imagen original donde cada elemento de la Imagen Integral contiene la suma de todos los píxeles localizados en la región superior izquierda de la imagen original.

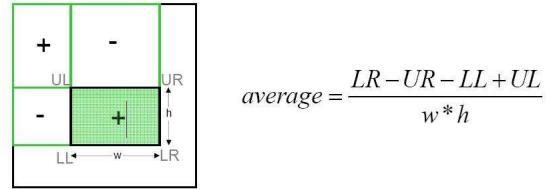


Figura 4.1: Imagen integral [ATI].

Permitiendo calcular las medias de intensidades de áreas de la imagen solo con cuatro búsquedas como se aprecia en la figura 4.1.

## 4.2. Primer método

Este método consiste en tres filtros en cascada, es decir, aplicar un filtro y con los resultados obtenidos, aplicar el siguiente filtro, y así sucesivamente. Los resultados se irán guardando en una lista de puntos *Puntos* que contienen las coordenadas de éstos. A continuación se explicarán los tres filtros y se mostrarán los pseudocódigos de los dos primeros ya que son los más complejos para explicar.

### ■ Primer filtro

Éste consiste en guardar los píxeles que superen un cierto umbral de intensidad  $Ui$ . Para empezar el umbral se calcula para cada imagen. Este umbral se calcula buscando todos los píxeles que superen el nivel de intensidad 10, haciendo esto ayuda a eliminar problemas con el ruido del medio extracelular. Ésto se aplica para intensidad 10, ya que es un nivel muy bajo y es seguro que los máximos de la matriz no serán descartados. Una vez obtenidos los píxeles se hará el sumatorio de todas las intensidades de estos píxeles y se hará la media obteniendo  $Ui$ . Ahora que tenemos el umbral habrá que recorrer toda la imagen de nuevo, píxel a píxel, se calculará el nivel de intensidad medio de esa área  $Ima$ , utilizando la imagen integral, que será de 3x3 píxeles siendo el píxel recorrido el central de la área.

*Pseudocódigo:*

**Entrada:** imagen en escala de grises *imagen*

1. Inicializar variable  $U_i \leftarrow 0$  y lista  $Puntos \leftarrow \emptyset$
2. **for**  $i \leftarrow 0, i < imagen.width$
3.     **for**  $j \leftarrow 0, j < imagen.height$
4.         **if**  $imagen[i, j] > 10$  **then**  $U_i \leftarrow (U_i + imagen[i, j])/2$
5. Calcular  $imagenIntegral$
6. **for**  $i \leftarrow 0, i < imagen.width$
7.     **for**  $j \leftarrow 0, j < imagen.height$
8.         Calcular  $Ima$
9.         **if**  $Ima > U_i$  **then**  

$$Puntos \leftarrow Puntos \cup [i, j]$$

Donde  $Ima$  se calcula con la fórmula que aparece en la figura 4.1.

#### ■ Segundo filtro

El segundo filtro consiste en descartar los píxeles que no cumplan la condición de que la intensidad de éste sea superior a la del área de píxeles colindantes.

Para cada uno de los píxeles que resultaron del filtro anterior ser candidatos a un máximo, se calculará la media de un área 3x3 de sus píxeles vecinos excluyendo del área el píxel candidato que se esté analizando  $Ima2$ , es decir, el centro del área será un píxel que esté como muy cerca a distancia de 1, véase figura 4.2.

Una vez hecho las medias de las 16 áreas se contabilizarán el número de áreas cuya media sea inferior al valor del píxel analizado. Si la mayoría de las medias de áreas son menores al píxel analizado, el píxel es de interés sino es descartado, ya que si la mayoría es mayor, quiere decir que no es un máximo local sino que uno de esos vecinos tiene más probabilidad de ser



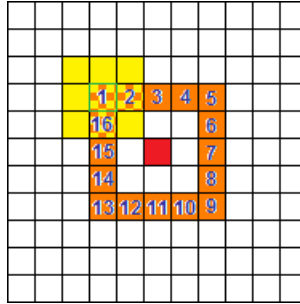


Figura 4.2: Cálculo del área de los píxeles colindantes.

un máximo local. Y así sucesivamente para todos los puntos resultantes del filtro anterior.

*Pseudocódigo:*

**Entrada:** imagen en escala de grises *imagen* y la lista *Puntos* obtenida en el filtro anterior.

1. **while**  $i < Puntos.length$
2.     **for**  $j \leftarrow 0, j < 16$
3.         Inicializar variable  $media \leftarrow 0$
4.         Calcular  $Ima2_j$
5.         **if**  $Ima2_j > imagen[Puntos[i].y, Puntos[i].x]$  **then**  
 $media \leftarrow media + 1$
6.         **if**  $media < 8$  **then**  $Puntos \leftarrow Puntos - Puntos[i]$

Donde  $Ima2_j$  es la media del área de uno de los 16 píxeles que se muestra en la figura 4.2.

#### ■ Tercer filtro

El último de los tres filtros se encarga de eliminar los píxeles cuya media de intensidad de sus colindantes sea mayor a la intensidad del píxel analizado, en otras palabras, se realizara la media del perímetro de la matriz. Así



Figura 4.3: Resultados del método 1.

eliminando píxeles que es más probable que sus vecinos sean máximos que ellos mismos.

Para esto se calculará la media de los píxeles exteriores de la matriz y se comprobará si es menor que el nivel de intensidad del píxel actual, si no lo es, se descartará el píxel, si lo es, se incluirá en la lista definitiva de candidatos.

Una vez aplicados los tres filtros lo que se obtiene es lo que se puede apreciar en la figura 4.3 a), unos grupos de píxeles, y a estos grupos para obtener el punto deseado, se le aplicará el mismo proceso que en el método anteriormente explicado, calcular el centroide de los puntos. Una vez hecho esto los resultados obtenidos son los de la figura 4.3 b).

#### 4.2.1. Implementación

En la primera implementación para la fase de prueba fue realizada en lenguaje Matlab. Éste no se ha llegado a desarrollar en lenguaje C++, como se aclarará en el apartado de Comparativa y evaluación, no fue idóneo para los requisitos de la aplicación.

### 4.3. Segundo método

Para empezar, los valores en que se comprende la imagen son de 0 a 255, siendo 0 el negro y 255 el blanco. Los puntos que se desean encontrar son los más luminosos de la imagen, lo que quiere decir que serán máximos locales en la imagen, lo que a su vez si el valor máximo de la imagen, por ejemplo 160, estos puntos serán de una intensidad algo menor. Otro factor a tener en cuenta son las regiones que se pueden descartar con seguridad, las zonas exteriores a la célula, oscilan entre los valores 1 a 10. Sabiendo esto, se puede eliminar las regiones cuyo valor no supere 10. Así, se eliminan las zonas de no interés que puedan introducir ruido a los resultados. Una vez eliminadas estas zonas, se puede realizar el método para detectar los puntos.

Los puntos luminosos que se buscan están asociados a cambios bruscos en el nivel de intensidad. Este método [CaSh] se basa en estos cambios y consta de tres pasos:

#### ■ Primer paso

La segunda derivada, o *numerical Laplacian*, para una imagen viene dada por el sumatorio de las derivadas parciales de cada dimensión.

$$\Delta f = \sum_{i=1}^n \frac{\partial^2 f}{\partial x_i^2}$$

Permite encontrar las variaciones de intensidad mayores: cuanto mayor sea el cambio en la imagen original, mayor será el valor en los puntos donde surge este cambio. Como se puede observar en la figura 4.4, los picos de la figura son donde se produce más variación en la intensidad de la imagen original.

#### ■ Segundo paso

A continuación, para eliminar los ruidos de alta frecuencia, se le pasará un filtro espacial de baja frecuencia, es decir, aplicar una convolución que difumine. En este caso es una convolución discreta, ya que los píxeles que

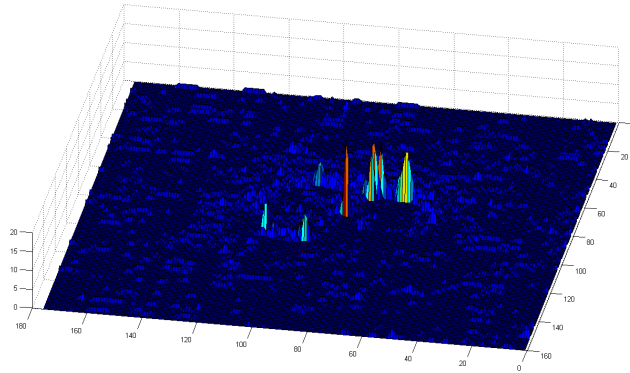


Figura 4.4: Niveles de intensidad despues de aplicar Laplace.

se desean tratar tienen valores discretos. Mediante este proceso se calcula el valor de un determinado punto en función de su valor y del valor de los puntos que le rodean, aplicando una simple operación matemática en función de la cual se obtendrá un valor resultante para el punto en cuestión.[AdDa]

$$g[m, n] = f[m, n] \otimes a[m, n] = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f[m, n] a[m - j, n - k]$$

Donde  $a[x, y]$  kernel,  $f[x, y]$  es la imagen de entrada y  $g[x, y]$  es la imagen filtrada.

Según el kernel o máscara de convolución que sea, se pueden obtener diferentes resultados de cómo enfocar una imagen, difuminarla, detección de bordes u otras operaciones.[GIMP]

El resultado que se desea obtener es el difuminado de la imagen resultante del paso anterior eliminando los ruidos de alta frecuencia.

#### ■ Tercer paso

Una vez obtenida la derivada suavizada, ahora se realizará una transformación en blanco y negro de la imagen. Como los picos de la nueva imagen

## 24CAPÍTULO 4. LOCALIZACIÓN Y TRACKING DE LOS PUNTOS LUMINOSOS

son los cambios más bruscos de intensidades, ahora solo se deberá escoger un umbral, que especifique a partir de ese umbral los valores mayores en la imagen serán blancos y los menores serán negros. Para la realización del proyecto se ha escogido el umbral con el que se obtienen mejores resultados.

Ahora, para poder eliminar los grupos de puntos menores de 3x3 píxeles se aplicará un proceso de erosión en la imagen [SDC]. Al eliminar estas zonas se está reduciendo la introducción de ruido en los resultados, ya que en estas zonas la concentración es menor a las otras, por no tener más de 3x3 píxeles mayores que el umbral o puede ser otra de las zonas a considerar, pero que por el umbral haya sido separada de su grupo de píxeles vecinos, lo que provocaría que éste sea contabilizado como otro punto de concentración a parte.

El proceso de erosión consiste en recorrer los píxeles que forman el contorno de una región de píxeles que se encuentren como mucho a una distancia de 1x1 píxeles y poner a 0 los píxeles que nos indique la matriz de filtro mediante la operación AND píxel a píxel, situando el centro de la matriz filtro en el píxel de la imagen a tratar.

Como resultado hemos eliminado las zonas no deseadas y dejado las que nos interesan, pero con un tamaño reducido.

El siguiente paso es aplicar el proceso contrario a la erosión, la dilatación, que consiste en el proceso inverso: en lugar de poner ceros, se pondrán unos para así restaurar el tamaño de las zonas que nos interesen. En la dilatación, en lugar de hacer una multiplicación, se hará la operación OR píxel a píxel, funcionando exactamente igual que el proceso anterior, excepto por la operación a realizar y la matriz de filtro, que será distinta.

Así pues, el resultado obtenido es el que se puede apreciar en la figura. 4.5, donde se han eliminado zonas no deseadas.

Ya para finalizar el proceso y obtener las coordenadas de los puntos, lo que se realiza ahora es calcular el centroide de los grupos de píxeles que tienen un 1 de valor. Para realizar esto, lo que se realiza es un recorrido de la imagen hasta encontrar un píxel con valor 1, a este píxel se le asigna que es de un grupo y a

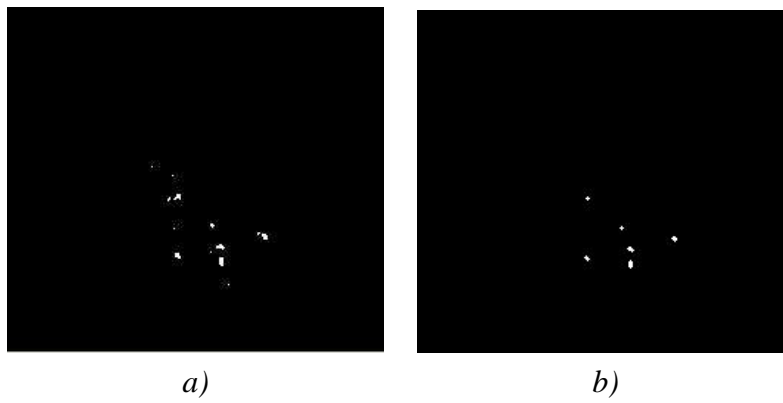


Figura 4.5: Erosión y dilatación

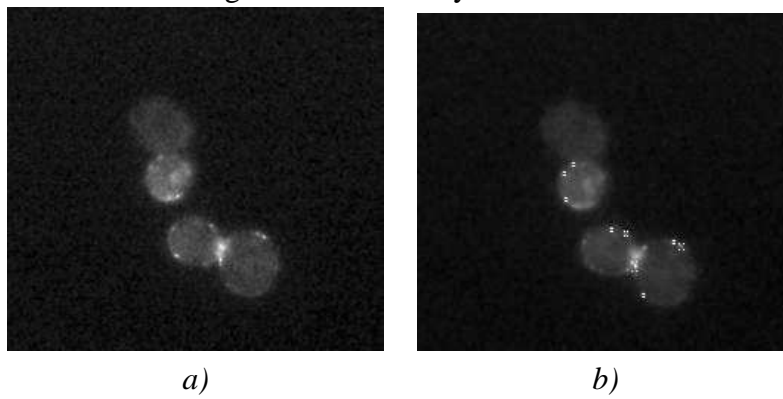


Figura 4.6: Original y resultado

continuación, se exploran los píxeles conexos a éste, con conectividad 8, comprobando primero que el valor sea 1, que no pertenezca al grupo. Una vez hechas estas comprobaciones se añade a la lista del grupo, y así recursivamente hasta que no haya píxeles conexos que no pertenezcan al grupo y cuyo valor sea 1. Así pues, se sigue recorriendo la imagen entera hasta encontrar otro píxel con valor 1 y que no pertenezca a ningún grupo, entonces se repite el proceso anterior.

Una vez explorada toda la imagen, de los grupos se mira el máximo y el mínimo de las coordenadas X e Y obteniendo un rectángulo que contiene el área donde se encuentra el grupo de píxeles, ahora solo quedaría calcular el centro de ese rectángulo y obtendremos el punto deseado. Véase figura 4.6 donde a) es el original y b) es sale las puntas del recuadro y el centroide.

#### 4.3.1. Pseudocódigo

**Entrada:** imagen en escala de grises  $Ima, Kernel$  para la convolución.

1. Inicializar  $PuntosFinal \leftarrow \emptyset, Grupo \leftarrow \emptyset, l \leftarrow 0$
2.  $Im \leftarrow f''Ima$
3.  $Im \leftarrow Im \otimes Kernel$
4.  $Imbw \leftarrow ConversionBN(Im, Umbral)$
5.  $Imbw \leftarrow Erode(Imbw, 3)$
6.  $Imbw \leftarrow Dilate(Imbw, 3)$
7. **for**  $i \leftarrow 0, i < Imbw.width$
8.     **for**  $j \leftarrow 0, j < Imbw.height$
9.         **if**  $Imbw[i, j] == 1$  **AND**  $(i, j) \notin Grupo$  **then**
  - $Grupo[l] \leftarrow Conectividad(Imbw[i, j])$
  - $l \leftarrow l + 1$

10. **for**  $k \leftarrow 0, k < \text{Grupo.length}$
11.     Calcular  $\text{MaxX}, \text{MaxY}, \text{MinX}, \text{MinY}$
12.      $\text{PuntosFinal}[k] \leftarrow ((\text{MaxX} - \text{MinX})/2, (\text{MaxY} - \text{MinY})/2)$

Donde *ConversionBN* es una función que dada una imagen y un umbral la transforma en blanco y negro, *Conectividad* es una función para obtener los puntos conexos, *Erode* función dada una imagen y un número  $n$  la erosiona  $n$  píxeles, *Dilate* lo mismo que la anterior, pero dilata.  $\text{MaxX}$  es el máximo de un grupo de las coordenadas X,  $\text{MaxY}$  es el máximo de un grupo de las coordenadas Y,  $\text{MinX}$  es el mínimo de un grupo de las coordenadas X,  $\text{MinY}$  es el mínimo de un grupo de las coordenadas Y.

#### 4.3.2. Implementación

La primera fase de prueba de este algoritmo se implementó en lenguaje Matlab y se testeó en Matlab para comprobar, analizar y comparar sus resultados posteriormente. Se usó Matlab para la fase de prueba debido que es más simple el desarrollo en este lenguaje y ahorrar tiempo es una implementación C++ que era posible que este algoritmo no fuera utilizado en la versión final de la aplicación.

La versión final de este algoritmo ha sido implementada en C++ junto con las librerías de Procesamiento de Imágenes OpenCV con licencia BSD permitiendo así la libre distribución al CSIC sin problemas de licencias y ahorrando. Todo el proceso se realizó sobre la plataforma Visual Studio 2005.

### 4.4. Tracking de los puntos

Para la implementación de tracking primero se ha de obtener todos los puntos de cada frame de vídeo. Una vez obtenidos los puntos se realizará las comprobaciones necesarias entre el frame actual y el frame anterior. La primera comprobación es, dado un punto en el frame actual, mirar que puntos están situados a una distancia 10 (respecto a píxeles). Dado que puede haber más de uno, debido



a que haya dos o más puntos muy próximos entre si, se mirará de todos los puntos obtenidos con la condición anterior el que se encuentre a la distancia mínima, asegurando que no hayan confusiones entre puntos cercanos.

Para saber en cuantos frames aparece el mismo punto, en la estructura de los puntos, existe un campo llamado frecuencia, el cual indica el número de frames en que el punto ha aparecido en el vídeo solo en el último frame en que aparece el punto, el punto en el resto de frames anteriores, el campo frecuencia esta marcado como un -1 para así poder descartarlos a posteriori y solo tener los puntos con sus frecuencias de aparición.

#### **4.4.1. Implementación**

El tracking se ha realizado y probado en C++ ya que su realización fue después de la implementación en C++ de la localización de puntos.

# Capítulo 5

## Detección de células y septim ring

### 5.1. Detección de células

Para poder determinar el estado de la célula o células hay que localizarlas en la imagen, pero existe un problema que hace que se complique, es que las células al estar unidas dificulta el proceso mucho y reduce el número de métodos para la localización. Para la localización de las células se entrenará un filtro Adaboost con características Haar-Like, ya que éste se adapta a problema de que las células se encuentren unidas.

Este filtro se podrá integrar en el proyecto mediante un función de las librerías OpenCV. La función se le pasará como parametros el filtro previamente cargado y la imagen, y lo que retornará será una estructura que contiene los puntos superiores izquierdos de la región y su altura y anchura. Una vez localizadas las células se podrá averiguar en qué estado se encuentra comparando el número de píxeles y el diámetro de la región en que se encuentre la célula.

#### 5.1.1. Adaboost

El método propuesto para localizar las células es el Adaptive Boosting o Adaboost, que es un algoritmo de aprendizaje, formulado por Yoav Freund y Robert Schapire. Éste puede ser usado conjunto a otros algoritmos de aprendizaje para mejorar su rendimiento. Adaboost es adaptativo en el sentido de que sus sub-

secuencia de clasificadores se ajustan a las instancias mal clasificadas. Un punto negativo que tiene es que es sensible al ruido, y en las imágenes que usamos hay bastante ruido, pero puede ser atenuado con unos filtros que más adelante se explicarán. Por otra parte, el algoritmo es menos susceptible al overfitting que otros algoritmos de aprendizaje. El funcionamiento del adaboost consiste en llamar a un clasificador débil repetidamente en unas series de rondas. En cada ronda se distribuye un peso a cada clasificador, que indica la importancia del conjunto de datos, actualizando los pesos según si son clasificados erróneamente el peso será mayor, o correctamente que será a la inversa, para así luego minimizar el error generado.

### 5.1.2. Características Haar-Like

Para poder crear los clasificadores para el Adaboost se usarán características Haar-Like [Int] en vez de trabajar con intensidades píxel a píxel. El conjunto de características Haar-Like está basado en las sumas del nivel de intensidad de regiones rectangulares. Una característica simple Haar-Like rectangular puede ser definida como la diferencia de la suma de los píxeles interiores al rectángulo, los cuales pueden estar en cualquier posición y escala de la imagen original. Este conjunto de características modificado se llama *2 rectangle feature*. Viola and Jones también definieron 3 y 4 características rectangulares. Cada tipo de característica puede indicar la existencia o no de un tipo de características en la imagen.

Para hacer el cálculo de estas áreas rectangulares se utiliza la Imagen Integral explicado anteriormente.

### 5.1.3. Entrenamiento

Para realizar el entrenamiento se requieren dos tipos de muestras:

- Imágenes de la célula. Estas imágenes son ejemplos positivos y sólo deben contener la célula en sí, para ello se establecerá un archivo con las coordenadas en píxeles de la célula dentro de la imagen. A partir de estos ejemplos positivos se buscarán las características Haar-Like que sean de relevancia



Figura 5.1: Filtro de mediana.

para distinguir las células de la imagen. Para eliminar parte del ruido de la imagen y así obtener mejores resultados, lo que se aplica a la imagen es un filtro de mediana [YoGe].

- Imágenes negativas o background, que son imágenes que no contiene la célula y que sean como el fondo de las imágenes positivas. A partir de estas imágenes se establecerán las características Haar-Like que difieran de las células y sean comunes en grupos de backgrounds.

Una vez preparadas las imágenes positivas y negativas se ejecutará el programa facilitado por OpenCV para empezar el entrenamiento, especificando el número de fases del entrenamiento que se desee [AIT],[Nao],[Flo].

#### 5.1.4. Filtro de mediana

El filtro de mediana es un filtro paso bajo que se utiliza para eliminar el ruido y conservar los bordes. Lo que hace el filtro es reemplazar el valor de cada píxel por la mediana, en lugar de la media, como muestra la figura 5.1. Al hacer el filtro de mediana preservamos los bordes de las células para que el Adaboost pueda detectar correctamente las células, ya que haciendo un filtro de convolución difuminaría los bordes, dificultando la localización de las células.

#### 5.1.5. Implementación

Para el entrenamiento del Adaboost se usó un ejecutable distribuido por OpenCV, que sirve exclusivamente para realizar entrenamiento con características Haar-Like.

Otra herramienta que se usó para la realización del Adaboost fue una aplicación en la cual especificabas un archivo con las imágenes y te mostraba todas las imágenes pudiendo así seleccionar la región que contenga la célula y guardando sus coordenadas en un fichero aparte preparado para ser usado por la aplicación de entrenamiento.

Una vez obtenido el clasificador como un archivo XML, éste ha sido integrado en el código de la aplicación pudiéndose usar con una función de las librerías OpenCV.

## 5.2. Detección del septim ring

Para poder localizar el septim ring se ha de localizar las células en la imagen, como se explica en la sección anterior, para así averiguar cual es la zona de unión de la célula madre e hija o septim ring. Teniendo las regiones donde se hayan las células podemos encontrar donde está el septim ring, para poder averiguar si un punto pertenece al septim ring tiene que cumplir una de las dos condiciones que se exponen a continuación.

1. El punto debe estar dentro de dos áreas de célula. Como se puede apreciar en la figura 5.2 a) existe una región que es común en las áreas de las dos células.
2. Dado que las células no siempre estarán en una posición que facilite la comprobación de la primera condición. La segunda condición es que el punto debe estar situado a un mínimo de distancia entre las áreas de cada célula. Véase el caso de la figura 5.2 b).

Si en la imagen no hay más de una célula, no se realizará ninguna comprobación, si existe más de una célula, se comprobará primero una opción y luego si no cumple la siguiente. En caso de no cumplir ninguna de las dos, se considerará que no pertenece al septim ring.

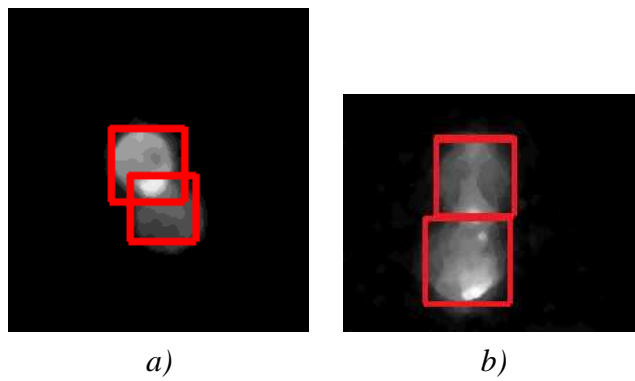


Figura 5.2: Casos a diferenciar para localización del septim ring.



# Capítulo 6

## Aplicación

En este capítulo se explicará como se ha realizado la aplicación. Ésta ha sido desarrollada en entorno Visual Studio 2005 con lenguaje C++ y las librerías OpenCV.

En los primeros pasos de la elaboración se implemento el método para la localización de puntos. Para ello se tubo que integrar las librerías OpenCV y crear tres clases con sus respectivas funciones:

**Puntos** : En esta clase se guarda las coordenadas de los puntos.

**Nodo** : Esta clase contiene la clase Punto, la frecuencia de aparición de los puntos, la intensidad máxima, el tipo de célula y si se encuentra en el septim ring, además contiene un apuntador a otro Nodo para realizar una lista.

**ListaP** : Es una lista de nodos, lo que contiene es un Nodo inicial, un actual y el final, además de el tamaño de la lista y sus operaciones, como resetear el apuntador actual, borrar o añadir Nodo.

Para la localización de las células sólo se ha tenido que trabajar con los datos obtenidos a partir de una función de OpenCV la cual carga el filtro previamente entrenado mediante Adaboost.

Una vez hecha estos primeros pasos, se ha integrado todo lo obtenido en la interfaz gráfica. La interfaz gráfica ha sido creada a partir de una aplicación MFC



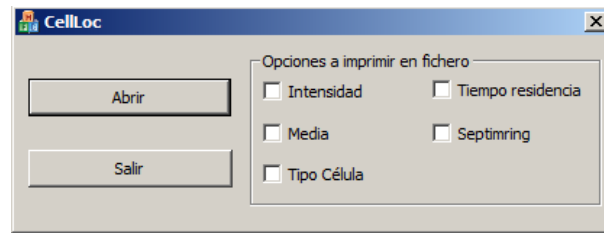


Figura 6.1: Interfaz gráfica de la aplicación.

de Visual Studio 2005. Ésta contiene una botón para abrir el vídeo a analizar, y unas casillas que permiten seleccionar entre las siguientes opciones a mostrar en fichero:

- Septim ring: indica si el punto se encuentra o no en esta zona.
- Tiempo residencia: muestra este tiempo.
- Intensidad: muestra la intensidad máxima.
- Tipo de célula: Muestra en que tipo de célula en que se encuentra el punto.
- Media: muestra las medias de todas las intensidades mencionadas.

Todos estos resultados son escritos en un fichero que se encuentra en el mismo directorio que el vídeo, este fichero de texto tiene el mismo nombre que el vídeo pero con la extensión *txt*.

La aplicación final es la que muestra la figura 6.1.

# Capítulo 7

## Evaluación de resultados

En este capítulo se expone los resultados obtenidos de los dos métodos para la localización de las zonas donde la concentración de la proteína es mayor, los de la detección de las células para su posterior identificación del estado celular (aplicación, seguimiento y septim ring) y el conjunto de todos ellos en la aplicación. También se hará una comparativa entre los métodos de detección de los puntos luminosos.

### 7.1. Localización de los puntos luminosos

A continuación se mostrarán los resultados de los dos métodos para la localización de puntos luminosos.

#### 7.1.1. Método 1

Para este método en un principio fue testeado con 22 vídeos ya que sólo se facilitaron estos al empezar la realización del proyecto. Estos vídeos no tienen demasiado ruido, aunque su resolución es igual a la de los otros 178 que luego se proporcionaron. En el periodo de realización de este método se trabajó con los 22 vídeos. El cuadro siguiente muestra los resultados con los 22 vídeos.

Como se aprecia en el cuadro 7.1, el resultado obtenido es satisfactorio para las 22 imágenes y el número de falsos positivos es debido a que algunas imágenes

Total de puntos en 22 imágenes	Aciertos	Fallos	Falsos Positiv.
176	148 $\rightarrow$ 84,09 %	28 $\rightarrow$ 15,91 %	6 $\rightarrow$ 3,896 %
Total de puntos en 1 imagen ruidosa	Aciertos	Fallos	Falsos Positiv.
4	4 $\rightarrow$ 100 %	0 $\rightarrow$ 0 %	35 $\rightarrow$ 87,179 %

Cuadro 7.1: Resultados filtro método 1.

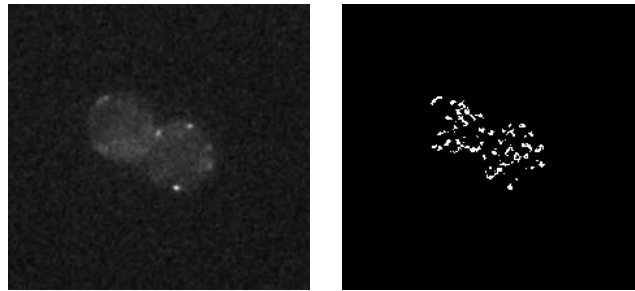


Figura 7.1: Problema del método 1 con el ruido.

contienen un poco más de ruido. A la hora de probar el primer método con el conjunto de 200 imágenes los resultados fueron insatisfactorios. El ruido en las imágenes genera demasiados falsos positivos.

En el cuadro 7.1 se muestra los resultados en una sola imagen, ya que contabilizar todos los falsos positivos manualmente es prescindible, viendo la figura 7.1 se puede generalizar los resultados:

Tal cantidad de falsos positivos es inadmisibles, ya que no se analizarían los datos reales, sería una forma de introducir ruido a los datos analizados.

### 7.1.2. Método 2

En esta sección se presenta los resultados del segundo método explicado. El testeo de este método se ha realizado con una imagen de cada vídeo que el CSIC proporcionó para la realización del proyecto, en total son 200 imágenes para comprobar los resultados de este algoritmo.

Para este test se estableció un mismo umbral manualmente para todas las imágenes, este umbral se obtuvo haciendo pruebas y comparando los resultados obtenidos, se usó el que mejor se adaptaba al conjunto de las 200 imágenes.

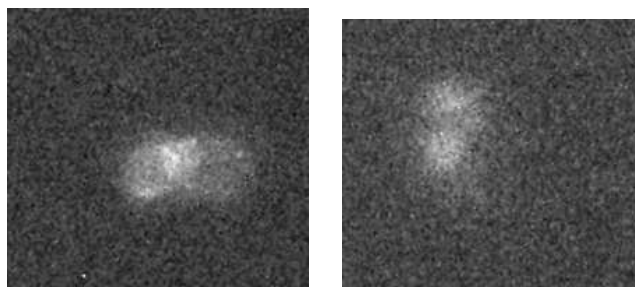


Figura 7.2: Imágenes con ruido.



Figura 7.3: Problema de resolución.

En las imágenes que se analizarán contienen gran cantidad de ruido como se puede observar en la figura 7.2 haciendo que interfiera bastante en los resultados negativamente. Otro factor en contra es la resolución de las imágenes, esta es relativamente baja en proporción *tamaño imagen*  $\leftrightarrow$  *tamaño célula*, ver figura 7.3, haciendo que gran cantidad de puntos luminosos cercanos se confundan como un solo punto muy largo.

A continuación se expone una cuadro con los resultados obtenidos.

Como muestra el cuadro 7.2, en número de falsos positivos es relativamente bajo y el porcentaje de aciertos es aceptable, a pesar de la gran cantidad de ruido que se encuentra en las imágenes y de su resolución, haciendo que este método se

Total de puntos en 200 imágenes	Aciertos	Fallos	Falsos Positiv.
886	616 $\rightarrow$ 69,525 %	270 $\rightarrow$ 30,475 %	18 $\rightarrow$ 2,839 %

Cuadro 7.2: Resultados filtro método 2.

pueda utilizar para la función de detección de puntos con resultados aceptables. Cabe destacar, que la mayoría de los que no se encontraron pertenecen a un mismo conjunto de imágenes, es decir, que por ejemplo si no se encontraron 20 puntos, estos 20 son de 4 imágenes en las cuales había mucho ruido y no se encontró ninguno. Y en la mayoría de las imágenes restantes, el número de acierto es mayor, solo en algunas falla por 1 punto o 2.

### **7.1.3. Comparativa de los 2 métodos**

En esta sección se expondrá una comparativa en los dos métodos para la localización de los puntos luminosos.

Como se puede apreciar en el cuadro del segundo método, el resultado es aceptable, y como se explicó anteriormente, sólo en un caso determinado el método no detecta bien los puntos debido al ruido en las imágenes. En cambio, el primer método se ajusta mejor a los primeros 22 vídeos testeados, ya que estos no tienen mucho ruido, pero por otra parte, la mayoría de vídeos incluyen una cantidad significativa de ruido que interfieren en los resultados obtenidos.

Así pues, lo bueno del segundo método es que no es tan sensible al ruido y puede ser usado en este caso. En cambio, el primer método en las imágenes que no contienen mucho ruido los resultados son mejores que en los anteriores, hay un menor número de no encontrados. Pero en este caso los vídeos incluyen ruido, no podemos decantarnos por el primer método, ya que en la mayoría de casos el resultado es insatisfactorio, por lo tanto, el método que se usa en la aplicación final es el segundo método.

## **7.2. Localización de células y septim ring**

A continuación se presentarán los resultados de las pruebas para la localización de las células y el septim ring.

Filtro $n^{\circ}$ ej. Neg	Aciertos	Fallos	Falsos Pos.	Falsos Pos.— Repe.	Falsos Pos.— Repe. — mala calidad
150	51 $\rightarrow$ 76,119 %	16 $\rightarrow$ 23,881 %	20 $\rightarrow$ 28,169 %	14 $\rightarrow$ 19,718 %	10 $\rightarrow$ 14,084 %
230	54 $\rightarrow$ 80,597 %	13 $\rightarrow$ 19,403 %	36 $\rightarrow$ 40 %	17 $\rightarrow$ 18,888 %	11 $\rightarrow$ 12,222 %

Cuadro 7.3: Resultados filtros para imágenes de distintos vídeos.

### 7.2.1. Evaluación de la localización de células

Para realizar el primer test de la localización, de los 200 vídeos que se facilitó, se tomó una imagen de cada uno, pero para realizar el entrenamiento se usaron 172 para el entrenamiento y 28 para el testeo, ya que para comprobar los resultados correctamente no se debe incluir imágenes del entrenamiento en el conjunto de pruebas. Con esta primera prueba se comprobará la efectividad en la detección según la variedad de las formas de las células. Luego se realizó una segunda prueba, esta se basa en el análisis de cuatro vídeos completos.

#### Test de las imágenes de distintos vídeos

El conjunto de 28 imágenes de distintos vídeos contenían 67 células bien definidas, esta fase de test sólo se probó con las células bien definidas, ya que en el entrenamiento también se tomó sólo las que estaban bien definidas debido a que introducir las que no se distinguían muy bien sería como introducir ruido en el entrenamiento, por lo tanto, el objetivo de este filtro es detectar las células que estén claramente definidas. Para esta primera prueba se entrenó dos filtros para realizar las pruebas, el primero constaba con 150 y el segundo con 230 ejemplos negativos. En el cuadro 7.3 muestra los resultados obtenidos, como se puede apreciar en el primer filtro, la tasa de acierto es elevada, de un 76,119 % en cambio la tasa de falsos positivos respecto al total de detecciones es del 28,169 %, es algo alta, pero de los 20 falsos positivos, 6 de ellos se podrían eliminar, ya que son células detectadas 2 veces, es decir, que la célula se enmarcó dos veces, un recuadro es de mayor tamaño y otro se ajusta a la célula siendo de menor tamaño que el anterior. Aplicando una comprobación de que si hay un área comprendida dentro de otra de mayor tamaño, si se elimina la de mayor tamaño, el número de falsos positivos se reduce a 14 siendo un 19,718 %. Pero de los falsos positivos, 4 de ellos son

Nº vídeo	Aciertos	Fallos	Falsos Pos.	Falsos Pos. – Repe.
1	88 → 56,41 %	68 → 43,59 %	30 → 25,423 %	13 → 11,016 %
2	72 → 90 %	8 → 10 %	20 → 21,739 %	8 → 8,695 %
3	23 → 28,75 %	57 → 71,25 %	13 → 50 %	5 → 22,222 %
4	9 → 10,715 %	75 → 89,285 %	32 → 78,048 %	27 → 65,853 %
5	30 → 100 %	0 → 0 %	13 → 30,232 %	11 → 25,581 %
6	98 → 81,666 %	22 → 18,333 %	35 → 26,315 %	17 → 12,781 %
7	86 → 71,666 %	34 → 28,333 %	12 → 12,244 %	9 → 9,375 %
8	72 → 92,307 %	6 → 7,693 %	8 → 10 %	6 → 7,5 %
9	15 → 19,23 %	63 → 80,77 %	2 → 88,235 %	2 → 88,235 %
10	80 → 97,56 %	2 → 2,44 %	16 → 16,666 %	12 → 12,5 %

Cuadro 7.4: Resultados de diez vídeos.

células que no se consideraron a la hora de establecerlas como de buena calidad, pero el filtro a sido capaz de localizarlas. Esto es bueno, ya que si fuese capaz de detectar todas las células, independientemente de la calidad que sea la muestra es el objetivo final de este filtro. Así pues, el número de falsos positivos son 10, un 14,084 %.

El siguiente filtro como se puede apreciar en el cuadro 7.3, mejora al anterior en porcentaje de aciertos, un 80,597 %. El número porcentaje de falsos positivos inicial es de un 40 %, muy elevado, pero considerando que 12 de los 36 son áreas repetidas y 6 más son células que no se marcaron al principio por su calidad, se consigue un 12,222 % siendo inferior el porcentaje al primero.

Viendo los resultados, se puede observar que este método es viable para el objetivo de localizar las células.

### Test completo los vídeos

En este apartado se mostrarán los resultados obtenidos al localizar las células en diez vídeos completos. El cuadro 7.4 muestra los resultados.

En la mayoría de los casos, el filtro obtiene buenos resultados, superiores al 70 %, en los casos que son inferiores, se encuentran problemas ya mencionados anteriormente, como la proporción de la célula respecto a la imagen o el ruido. En la inmensa mayoría de casos en que se obtengan peores resultados es debido al

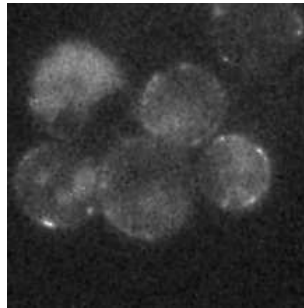


Figura 7.4: Problema con células demasiado juntas.

ruido. Otro problema existente, es que no siempre se hallen el mismo número de células en los frames, haciendo que no se pueda determinar con certeza en estadio celular en que se encuentre la célula, ya sea por el ruido, o porque la célula madre o hija apenas se distingue del fondo de la imagen. Pero aún así, los resultados son satisfactorios pudiendo identificar el estadio celular en que se encuentran las células del vídeo.

### **Limitaciones**

Este proceso para la localización tiene una limitación, es que cuando aparecen más de un grupo de células demasiado juntas, ver figura 7.4, en este caso, la aplicación no puede definir el estadio celular ni la localización del septim ring, ya que las células están pegadas y no se puede distinguir las parejas de células y a consecuencia de esto, el la localización del septim ring tampoco se puede establecer.

#### **7.2.2. Septim ring**

Los resultados de que un punto pertenece al septim ring depende totalmente en que si la localización de las células ha funcionado correctamente. Con las dos comprobaciones, establecer si un punto esta situado en en septim ring es muy efectiva en los vídeos analizados. Sólo en pocos casos se da que debido a que el detector de células establece el área de la célula demasiado ajustado, el requisito de la distancia mínima entre los dos recuadros no se cumple, éste es el único caso



en que no funciona correctamente. En los vídeos testeados no se ha encontrado con este problema.

### **Limitaciones**

La única limitación es que depende totalmente del resultado obtenido en la localización de las células mencionada anteriormente, si no detecta correctamente las células, este método no podrá establecer si pertenece al septim ring los puntos luminosos.

## **7.3. Tracking de los puntos**

El tracking funciona correctamente, pero para un caso específico la implementación realizada no funciona correctamente. El problema hallado es a la hora de detectar los puntos, a veces se da el caso en que había un sólo punto en el frame anterior, y en el siguiente frame aparecen dos puntos cerca de donde había el anterior, entonces a veces el método establece la misma frecuencia para los dos puntos que aparecieron en el frame siguiente. Salvo este problema el método funciona correctamente.

## **7.4. Aplicación final**

La aplicación final no muestra problemas en las pruebas realizadas en distintos ordenadores y con los vídeos probados. Tiene una interfaz simple para el uso intuitivo. No se ha encontrado ningún error.

Como conjunto, toda la aplicación funciona correctamente, salvo las limitaciones comentadas anteriormente, que como se verá en el último capítulo, se hacen diversas propuestas para eliminar estas limitaciones y mejorar los resultados obtenidos.

# Capítulo 8

## Conclusiones y líneas futuras

En este último capítulo se explicarán las conclusiones a las que se han llegado y se expondrán las futuras mejoras para el proyecto.

### 8.1. Conclusiones

El proyecto tenía cuatro objetivos, el primero era la búsqueda de los puntos luminosos, que ha sido cumplido con unos buenos resultados, la distinción entre los tipos de estadio celular y la localización del septim ring, también se ha cumplido, aunque como se explicará en la sección de líneas futuras podría mejorar sus resultados. El seguimiento en la secuencia de vídeo de los puntos luminosos también se ha cumplido. Y el último objetivo, el de realizar una aplicación con interfaz gráfica y el procesado de datos, también se han cumplido.

Este proyecto ha sido de gran interés ya que se ha podido comprobar como la informática puede ser de gran utilidad en cualquier campo. Otro aspecto interesante ha sido el hecho de tener que investigar para poder realizar este proyecto y utilizar las librerías OpenCV que son unas de las más completas que existen actualmente.

Los objetivos establecidos en un principio se han cumplido, la aplicación puede ser útil a los miembros del laboratorio del CSIC, pero ésta se podría mejorar para que fuese aún más útil y fiable para los miembros del CSIC.

## 8.2. Líneas futuras

En esta sección se van a proponer diversas formas para mejorar el funcionamiento de la aplicación. A continuación se listan las posibles mejoras a realizar:

- Se puede perfeccionar el método de la localización células y su identificación ya que tienen limitación en cuanto a distinguir el tipo de célula cuando hay más de un grupo en el mismo vídeo.
- Realizar un nuevo entrenamiento del Adaboost de más tiempo y en un ordenador superior al que se realizó, ya que puede tardar muchísimo tiempo el entrenamiento en un PC convencional
- En la obtención de puntos se podría establecer un umbral automático adaptable ya que ahora el que hay es uno fijo que es el que mejor se adapta a las muestras que hay.
- Perfeccionar la aplicación final, añadiendo la opción de exportar a hoja de cálculo y el análisis completo de un directorio en lugar de tener que seleccionar vídeo a vídeo

# Apéndice A

## Manual de usuario

Este programa a sido diseñado exclusivamente para el análisis de células.

### A.1. Instalación

No es necesaria ninguna instalación, para usar la aplicación sólo requiere copiar el archivo ejecutable en un PC con sistema operativo Windows™. También asegúrese de tener instalados los códecs de vídeo necesarios, para comprobar si los tiene, pruebe visualizar el vídeo que se desea analizar, si lo visualiza correctamente puede ejecutar la aplicación sin problema.

### A.2. Funcionamiento

Una vez abierta la aplicación saldrá una ventana como muestra la figura A.1. Para empezar a usar la aplicación, primero seleccione entre las opciones que muestra el cuadro “*Opciones a imprimir en fichero*”, a continuación viene una breve descripción de lo que mostrará las opciones en el fichero de salida al ser seleccionadas:

- Intensidad: muestra la intensidad máxima de los puntos luminosos.
- Tiempo residencia: muestra el tiempo de residencia de los puntos.

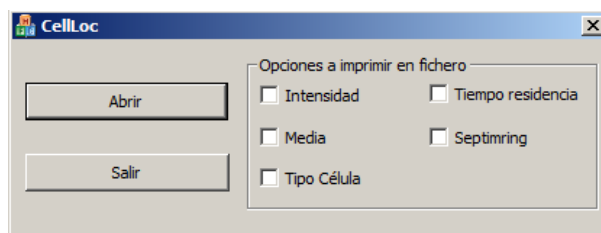


Figura A.1: Interfaz gráfica de la aplicación.

- Media: muestra la media de todos los puntos de de las opciones anteriores.
- Septim ring: muestra si el punto se encuentra en la zona del septim ring o no.
- Tipo célula: muestra el estadio celular en que se encuentra la célula que contiene el punto y si es madre o hija.

Una vez seleccionadas las opciones que se desean, haciendo clic en el botón abrir se abrirá una ventana de exploración que le permitirá seleccionar el vídeo que se desea analizar. Una vez se le de a abrir el fichero, la aplicación analizará el vídeo automáticamente, guardando los resultados en un fichero de texto con el mismo nombre que el archivo de vídeo analizado. El archivo de texto será substituido si ya existe uno con el mismo nombre, asegúrese que si no desea perder este fichero, cambiar el nombre al que ya existe.

### A.3. Formato del fichero de salida

En el fichero de salida, se muestra frame a frame los puntos hallados en el análisis. Los puntos se identifican mediante coordenadas  $X, Y$ . Los puntos se muestran en el último frame que se encontró ese punto, es decir, si un punto estuvo en los frames 3, 4 y 5, el punto se mostrará en el frame 5.

En la opción de intensidad, la intensidad que se muestra es a la escala de 0 a 255. El tiempo de residencia que se muestra, son el número de frames en que

apareció el punto. La media se muestra en la última línea del fichero. El septim ring, si se encuentra el punto en esta zona, saldrá *SR* en el fichero, sino saldrá *NSR*.

Para el tipo de célula se encuentran las siguientes etiquetas:

- *SBm*: Small Budded Mother.
- *SBD*: Small Budded Daughter.
- *BBm*: Big Budded Mother.
- *BBd*: Big Budded Daughter.
- *B*: Budded.
- *IND*: sin identificar.

## A.4. Terminar aplicación

Cuando se desee terminar la aplicación puede cerrarla apretando al botón de salir o a la “X” que aparece en la parte superior izquierda de la ventana.



# Bibliografía

- [AdDa] Adriana Dapena, “Técnicas de procesado de imagen”, 2006  
<[http://www.des.udc.es/~adriana/TercerCiclo/CursoImagen/curso/web/Filtrado\\_Espacial.html/](http://www.des.udc.es/~adriana/TercerCiclo/CursoImagen/curso/web/Filtrado_Espacial.html/)>
- [AIT] AIT Computer Vision Wiki, “HaarTraining for Pedestrian Detection”, 2003  
<<https://webeng.cs.ait.ac.th/cvwiki/opencv:tutorial:haartraining/>>
- [ATI] ATI, Dynamic Glossy Environment Reflections y Thorsten Scheuermann, “Summed-Area Tables“, 2005  
<[ati.amd.com/developer/gdc/GDC2005\\_SATEnvironmentReflections.pdf](http://ati.amd.com/developer/gdc/GDC2005_SATEnvironmentReflections.pdf)>
- [CaSh] A. E. Carlsson, A. D. Shah, D. Elking, T. S. Karpova y J. A. Cooper, “Quantitative Analysis of Actin Patch Movement in Yeast”, Mayo 2002
- [DaMi] David Millán, “OpenCV I - Highgui”, Abril 2005  
<<http://www.artresnet.com/david/tutorial.jsp?id=4/>>
- [Flo] Florian Adolf, “OpenCV’s Rapid Object Detection”, Septiembre 2003
- [GIMP] GIMP, “The GIMP Documentation Team”, 2007  
<<http://docs.gimp.org/es/plug-in-convmatrix.html/>>



- [Int] Intel, “Learning vision: Face detection”,2005  
<[http://www.intel.com/technology/itj/2005/volume09issue02/art03\\_learning\\_vision/p04\\_face\\_detection.htm](http://www.intel.com/technology/itj/2005/volume09issue02/art03_learning_vision/p04_face_detection.htm)>
- [JeLl] Jesús Llor, “Configurar OpenCV para usar en Visual Studio 2005”,Abril 2008  
<<http://www.jesusllor.es/?p=24/>>
- [Mat] Matlab  
<<http://www.mathworks.com/>>
- [MeZa] Mehran R. Zaini, Gary J. Forest y David D. Loshek, “Image processing for IMRT QA dosimetry”, Octubre 2004  
<<http://www.jacmp.org/index.php/jacmp/article/viewArticle/2071/1228/>>
- [Nao] Naotoshi Seo, “Tutorial: OpenCV haartraining”, Julio 2007  
<<http://note.sonots.com/SciSoftware/haartraining.html>>
- [OpCV] OpenCV  
<<http://www.intel.com/technology/computing/opencv/>>
- [Ralp] Ralph W. Pike, “CLASSICAL THEORY OF MAXIMA AND MINIMA”, 2001  
<<http://www.mpri.lsu.edu/textbook/Chapter2-a.htm/>>
- [SVM] LS SVM LAB  
<<http://www.esat.kuleuven.ac.be/sista/lssvmlab/>>
- [SDC] SDC Information Systems, “mmdCells,Extract blood cells and separate them”, 2008  
<<http://www.mmorph.com/cppmorph/morph/mmdemos/mmdCells.html/>>
- [TyIg] Ty C. Voss, Ignacio A. Demarco y Richard N. Day, “Quantitative imaging of protein interactions in the cell nucleus”, Septiembre 2005  
<<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1237115/>>

- [VioJo] P. Viola, M. Jones y D. Snow, “Detecting pedestrians using patterns of motion and appearance”, IEEE International Conference of Computer Vision (ICCV), 2003.
  
- [YoGe] I.T. Young, J.J. Gerbrands y L.J. Van Vliet, “Image Processing Fundamentals”, 1998  
<<http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Smoothin.html>>

---

Firmado: Leandro Jiménez Rodríguez  
Bellaterra, Septiembre de 2008

## **Resum**

Aquest projecte s'ha desenvolupat a petició del CSIC. El projecte consisteix en crear una aplicació que automatitzi la localització i obtenció de la intensitat de uns punts lluminosos en cèl·lules que apareixen en el vídeo. També es desitja identificar en què estadi cel·lular es troba les cèl·lules y si els punts trobats estan a la zona del septim ring. Per això, se ha fet un estudi i provat 2 mètodes per a la localització dels punts y s'ha realitzat un filtre mitjançant Adaboost per a la localització de les cèl·lules. També s'ha realitzat una interfície gràfica per a l'usuari final.

## **Resumen**

Este proyecto se ha desarrollado a petición del CSIC. El proyecto consiste en crear una aplicación que automatice la localización y obtención de la intensidad de unos puntos luminosos en las células que aparecen en el vídeo. También se desea identificar en qué estadio celular se encuentran las células y si los puntos hallados están en la zona del septim ring. Para ello, se ha hecho un estudio y probado 2 métodos para la localización de los puntos y se ha realizado un filtro mediante Adaboost para la localización de las células. También se ha realizado una interfaz gráfica para el usuario final.

## **Abstract**

This project has been developed by request of the CSIC. The project consists of creating an application that automates the location of luminous points inside the cells that appear in the video and obtaining it's intensity. Also it is wanted to identify in which cellular stadium the cells are and if the points are in the septim ring zone. For it, a study has been done and tried 2 methods for the location of the points and a filter has been realized by Adaboost for the location of the cells. Also it has been made a interface graph for the final user.