
Proyecto Final

AntSwarm

DanielGoldberg - José Indalecio Liendro

Instituto Tecnológico de Buenos Aires

Ingeniería Informática – Agosto 2012

Tutor:

Dr. Daniel Parisi

Índice

1. Introducción al problema	1
1.1 Objetivos.....	1
1.2 Especificación de requerimientos.....	2
1.2.1 Requerimientos funcionales	2
1.2.2 Requerimientos no funcionales.....	4
1.2.3 Requerimientos no incluidos dentro del alcance	4
1.3 Análisis funcional	4
1.3.1 Actor	4
1.3.2 Casos de uso	5
2. Arquitectura y diseño.....	6
2.1 Arquitectura.....	6
2.2 Configuración	9
3. Transformación de la información.....	15
3.1 Procesamiento de imágenes.....	15
3.1.1 white_acrylic_background.m.....	17
3.1.2 white_acrylic_background_fast.m	18
3.1.3 silver_background.m	18
3.1.4 subtract_with_empty_t.m.....	18
3.1.5 power_metal_150.m.....	20
3.2 Procesamiento de videos	20
4. Aprendizaje automático	22
4.1 Detección de hormigas.....	22
4.2 Resolución de colisiones	23
4.3 Aprendizaje de tamaño de los agentes	24
5. Algoritmos utilizados.....	26
5.1 Clusterización por adyacencia.....	27
5.2 Algoritmo bwconncomp	27
5.3 Algoritmo Flood-Fill.....	27
5.4 Ejemplo de funcionamiento	28

5.5 Fuzzy c mean(fcm)	29
5.6 Conteo de hormigas	30
5.7 Seguimiento de hormigas	30
6. GUI – Interfaces visuales	31
6.1 Ant Swarm	32
6.2 Process center	33
6.3 AntParameters	34
7. Archivos de salida	36
7.1 Centro de masa	36
7.2 Masa total	36
7.3 Trayectoria de hormigas	37
7.4 Cluster de hormigas	37
7.5 Uso de la información	38
8. Indicadores	39
8.1 Centros de masa	40
8.2 Masa total	41
8.3 Trayectoria de hormigas	41
8.4 Black out del seguimiento	42
8.5 Velocidad	43
8.6 Espacio transitado	44
8.7 Indicadores de la interfaz	45
8.7.1 Tamaño de los agentes	45
8.7.2 Trayectoria individualizada	46
9. Guía del Usuario	47
9.1 Carga de parámetros para poder seleccionar hormigas a seguir	47
9.1.1 Abrir el video	47
9.1.2 – Elegir frame de comienzo	47
9.1.3 – Seleccionar máscara	48
9.1.4 – Analizar tamaño de hormigas	49
9.1.5 – Ingresar parámetros de evaluación de tamaño de hormiga	50
9.1.6 – Tamaño mínimo, máximo y media	50
9.1.7 – Guarda parámetros	51
9.2 Tracking de hormigas	51

9.2.1 – Seleccionar hormigas.....	51
9.2.2 – Ver camino de hormigas.....	52
9.3 Reproducción	52
9.3.1 – Configurar parámetros de comienzo y fin	52
9.3.2 – Configurar salidas deseadas	52
9.3.3 – Controles de video.....	53
9.4 Funciones de umbralizado	53
9.5 Detalles de UI	54
9.5.1 – Porcentaje de análisis completado.....	54
9.5.2 –Cantidad de hormigas en el frame actual	54
9.5.3 – Controles para hacer zoom in y zoom out de la región seleccionada	55
9.5.4 – Configurar el nombre de la especie analizada.....	55
10. Posibles extensiones y mejoras	57
11. Conclusiones	58
12.1 Papers	60
12.2 Funciones utilizadas.....	60
12.3 Algoritmos	60
Agradecimientos	61

1. Introducción al problema

Resulta de gran importancia el estudio del egreso de multitudes en estado de pánico, debido a que una adecuada caracterización de este sistema complejo permitirá el correcto diseño de edificios y vías de escape, lo que facilitará salvar vidas en caso de evacuaciones de emergencia.

Hasta el momento se ha encarado este problema mediante simulaciones computacionales, las cuales no se encuentran adecuadamente validadas con datos de la realidad. Esto se debe a la dificultad de obtener datos de multitudes en pánico bajo condiciones de laboratorio.

Con la finalidad de obtener datos de sistemas similares, se propuso realizar experimentos con agentes biológicos. En particular se eligieron insectos sociales: hormigas, las cuales son sometidas a distintos grados de alarma mediante la presencia de un estímulo. Los procesos de evacuación de las hormigas a través de una salida angosta se registran con cámara digital.

1.1 Objetivos

El objetivo general del presente trabajo final es aplicar y desarrollar las herramientas de procesamiento de imágenes y video necesarias para poder analizar los registros adquiridos en los experimentos particulares.

Los objetivos son los siguientes:

- Distinguir si un pixel dado representa el fondo o a una hormiga, utilizando técnicas de clasificación automática por niveles de gris analizando posibles errores de clasificación.
- Medir velocidades promedio del movimiento de las hormigas, analizando vectores de movimiento cuadro a cuadro.
- Seguimiento automático de las hormigas durante el proceso de evacuación y gráfico de trayectorias.

1.2 Especificación de requerimientos

1.2.1 Requerimientos funcionales

RF 1 - Interacción con videos

Relevancia	Alta
Solicitante	Cliente
Especificación	El sistema ofrecerá una interfaz para interactuar con los videos de estudio, tanto para su reproducción así como la interacción con los mismos, selección de áreas de interés, agentes y trayectorias e intervalo de análisis.

RF 2 - Extracción de métricas de interés

Relevancia	Alta
Solicitante	Cliente
Especificación	El sistema ofrecerá la posibilidad de extraer información en archivos de texto de distintos indicadores de interés.

RF 3 - Identificación de agentes

Relevancia	Alta
Solicitante	Cliente
Especificación	El sistema debe poder identificar qué pixel es un agente y cuál no. Además de poder discernir entre distintos agentes e identificarlos.

RF 4 - Resolución de colisiones

Relevancia	Alta
Solicitante	Cliente
Especificación	El sistema debe poder diferenciar agentes incluso cuando estos

	colisionen.
--	-------------

RF 5 - Seguimiento de agentes

Relevancia	Alta
Solicitante	Cliente
Especificación	El sistema ofrecerá la posibilidad de seguir a los agentes en el video, ya sea de forma individualizada o colectiva.

RF 6 - Guardar configuraciones

Relevancia	Media
Solicitante	Cliente
Especificación	Será posible en todo momento guardar la configuración que el usuario considere adecuada para el análisis de los videos.

RF 7 - Aprendizaje de agentes

Relevancia	Alta
Solicitante	Cliente
Especificación	El sistema deberá poder aprender qué objeto es un agente y qué no lo es.

RF 8 - Procesamiento iterativo de videos

Relevancia	Media
Solicitante	Cliente
Especificación	Una vez configurados los parámetros adecuados para cada uno de los videos, se espera poder procesar todos dichos videos de forma iterativa sin supervisión del usuario.

1. 2.2 Requerimientos no funcionales

RNF 1 – Sistema usable

Tipo	Usabilidad
Complejidad	Alta
RF	Todos
Especificación	La aplicación debe tener un alto grado de usabilidad para cualquiera de las funcionalidades ofrecidas. Dado el proceso de seguimiento es importante tener un feedback constante del tipo de procesamiento obtenido hasta el momento. Además de interactuar directamente con los frames del video, ya sea para seleccionar áreas, agentes o verificar trayectorias.

1.2.3 Requerimientos no incluidos dentro del alcance

1. Señales de alerta ante cierto comportamiento crítico de los agentes
2. Procesamiento en tiempo real de los videos analizados por el sistema.
3. Permitir obtener salidas en formato de video off line del proceso de análisis.
4. Compresión automática de los videos de entrada.
5. Diferenciar entre distintos tipos de agentes. Es decir, por ejemplo, en el caso de las hormigas, cuáles son obreras, cuál es la reina, etc. o bien diferenciar entre agentes de distintas especies.

1.3 Análisis funcional

1.3.1 Actor

Hay un único usuario de la aplicación

1.3.2 Casos de uso

Nivel de complejidad

- Simple (S)
- Media (M)
- Compleja (C)

ID CU	Nombre	Complejidad	IR Req.
CU01	Agregar nuevo videos al sistema	S	RF1
CU02	Reproducir un video	M	RF1
CU03	Seleccionar un área de interés	M	RF1
CU04	Seleccionar un agente o seguir a todos	C	RF3,RF4, RF5
CU05	Seleccionar un rango de frames de interés	S	RF1
CU06	Especificar el nombre de la especie de los agentes	S	RF6
CU07	Seleccionar cuáles son las salidas deseadas	M	RF2
CU08	Tomar una foto del frame actual como herramienta de debug	S	RF1
CU09	Elegir entre modos de visualización del video (pantalla completa, p reducida)	M	RF1
CU10	Elegir entre distintas funciones de umbralización	M	RF1
CU11	Guardar la configuración sobre el video actual	S	RF6
CU12	Contar agentes en el frame	C	RF1
CU13	Tener información relevante en una consola auxiliar	S	RF1
CU14	Estudiar el tamaño de agente para el video actual	C	RF7
CU15	Especificar parámetros de estudio de agentes	S	RF7
CU16	Sugerir al usuario un intervalo estimado	M	RF7
CU17	Guardar toda la información relevante al estudio de agentes	S	RF7
CU18	Procesar de forma iterativa todos los videos distponibles	M	RF8

2. Arquitectura y diseño

La aplicación que se construye para cubrir los objetivos del proyecto se denomina AntSwarm (AS), y en el presente informe los términos aplicación, programa, SWA refieren a la misma.

SWA se realiza en Matlab versión 2011a. Esto se debe a las facilidades que presenta con funciones para el procesamiento de imágenes a través de matrices de forma eficiente y la gran diversidad de herramientas de aprendizaje estadístico para poder extraer información de las entradas de los videos.

Todos los resultados a partir del análisis de videos presentes en este informe han sido realizados en dos arquitecturas de computadoras y sistemas operativos diferentes.

- Intel Core 2 GHz, 4 GB Ram - Windows 7
- Intel i7 2,2 GHz, 8GB Ram – MacOS X Lion 10.7.4

El código fuente de todo el proyecto se presenta adjunto a este informe.

2.1 Arquitectura

El estilo de programación tiene un modelo híbrido donde predomina la estructura imperativa y por otro lado el manejo de pseudo objetos en forma de estructuras con propiedades.

La aplicación tiene un diseño desktop stand alone, y se alimenta principalmente de los videos a analizar y archivos de configuración.

La interacción con el usuario es a través de una GUI – Interfaces visuales que se constituyen en distintas pantallas que se describirán en la sección 6.

Toda la información extraída se persiste en archivos de texto con extensión txt. Además, se persisten archivos de configuración en formato JSON e imágenes que pueden ser de interés para un análisis puntual o como posibles elementos para facilitar procesos de umbralización de los cuadros de los videos.

El desarrollo se distribuye en un árbol de carpetas para dar un orden al código fuente de AS, además de ordenar las salidas y archivos de configuración. La siguiente descripción corresponde a una configuración por defecto aunque esta puede adaptarse como se explica en el apartado de configuración.

La decisión sobre el anidamiento de las carpetas o dónde están los archivos del código fuente es arbitraria, salvo para los archivos de umbralización. Es decir que si se mueven los archivos fuentes de carpetas, AS seguirá funcionando sin cambio alguno. Por el contrario los archivos de configuración y salida sí podrían afectar el correcto funcionamiento de la aplicación.

La nomenclatura que se utiliza para describir los nombres de los archivos consiste en expresar valores variables entre signos de menor y mayor, lista de valores posibles prefijados entre corchetes y separados por una barra vertical, y por último texto fijo de formato fijo. Por ejemplo `<nombre de video>-<numero de frame>-[otsu|main].png` representa archivos que comienzan con el nombre de un video, seguido de un guion, luego un número, seguido de un guión, continuado por la palabra `otsu` o `main` y terminado por `.png`. Así una instancia de este tipo de nombre de archivo puede ser `video1.avi-1234-main.png`.

El proyecto cuenta con las siguientes carpetas

- Doc

Archivos de documentación tales como Informe final, presentación, guía de usuario y readme.

- Gui

Archivos de la interfaz visual (`.fig`) y su respectiva programación (`.m`)

- Images

Imágenes usadas para las interfaces visuales.

- Output

Aquí se guardan los archivos de salida y archivos de configuración

- `videoFrames` - Imágenes de frames arbitrarios que han sido guardadas por el usuario, así como también frames vacíos para un proceso de umbralización. En primer lugar los frames arbitrarios se denominan con la siguiente convención `<nombre de video><numero de frame>-[otsu|main].png`, mientras que los segundos `<nombre de video>empty.png`. A diferencia de los primeros las imágenes vacías sin hormigas son únicas, es decir que si el usuario decide guardar una segunda imagen, la misma reemplaza a la anterior.

- videoProcess
 - antParameters - archivos de configuración para el proceso de aprendizaje del tamaño estimado de una hormiga. Se utiliza la convención *<nombre de video>.stats.txt*
 - mass¢ermass - archivos de salida del proceso de análisis de la masa total de hormigas y la posición del centro de masa de la masa total en función de tiempo. Se utiliza la convención *<nombre de video>.[Masa|CentroMasa].txt*
 - clusterAnts – archivos de salida del proceso de análisis de cada cluster mayor al área de una hormiga. Se utiliza la convención *<nombre de video>.-antsCluster.txt*
 - paths - archivos con las trayectorias de las hormigas de un video en particular. Se utiliza la convención *<nombre de video>.path.txt*
 - images – en esta carpeta se ubican las imágenes que se han obtenido a partir del análisis de los resultados a partir del script *seepathtest.m*.
- videoSettings - archivos de configuración para cada video en particular relacionados con las opciones de reproducción y análisis, máscara utilizada, tamaño de hormiga aprendido, etc. Se utiliza la convención *<nombre de video>.js*

■ Src

En esta carpeta se encuentra el código fuente del proyecto.

- io - funciones para dar formato a la información y persistirla. Así como también alguna función para mostrar mensajes dentro de la interfaz.
- jsonlab - librería utilizada para el manejo de archivos en formato json.
- jsDAO - funciones para obtener información de archivos json.
- processAnts - funciones de análisis de hormigas, aprendizaje, seguimiento, conteo, selección
 - statistic - aprendizaje de tamaño de hormigas a través del estudio de los últimos frames del video.
- processImage - funciones auxiliares para el procesamiento de las imágenes cuadro por cuadro, como por ejemplo recortar la imagen a una zona de interés y umbralizar, además de dar un formato visual de los frames.
- umbralizado - distintas funciones de umbralización. A diferencia del resto de las carpetas que están a un mismo nivel es importante que las mismas no cambien de lugar.
- / en la raíz de esta carpeta se encuentran distintas funciones ligadas al funcionamiento de la interfaz. En particular se encuentra un archivo llamado umbralizar que cumple de nexo entre la interfaz y las funciones de umbralización,

es decir se encarga de que la interfaz use la función de umbralización adecuada según la elección del usuario.

- Test

Funciones auxiliares que fueron utilizadas para probar algunas funciones sobre videos y/o imágenes en el transcurso de todo el proyecto.

- /

En la raíz del proyecto se encuentra un archivo llamado settings.js que se encarga de contener los parámetros generales de la configuración general de la aplicación.

2.2 Configuración

AS permite manejar la configuración global de la aplicación a través de un archivo llamado settings.js ubicado en la raíz del proyecto. Así como también almacenar la configuración particular para el análisis de cada uno de los videos.

Configuración global - Settings.js

En settings.js se almacenan las rutas de las carpetas utilizadas para destinar los archivos de salida, las imágenes del sistema y los videos de entrada. Del mismo modo cómo parámetros estadísticos por defecto para el proceso de aprendizaje del tamaño de las hormigas. Este archivo tiene formato json.

Dependiendo el sistema operativo en el que se use AS varía la sintaxis del path, como por ejemplo en MAC es necesario anteceder los path con ./ antes de la ruta. En el apartado settings se hallan las rutas a las principales carpetas de interés

- VIDEOFILES: videos para analizar
- VIDEOSETTINGS: configuración particular para cada video
- VIDEOFRAME: imágenes de frames arbitrarios
- VIDEOPROCESS: archivos de masa y centro de masa
- VIDEOPATHS: archivos de trayectorias
- VIDEOPARAMETERS: parámetros estadísticos específicos para cada video

- IMAGES: imágenes del sistema

En defaultscan los parámetros estadísticos de aprendizaje predefinidos

- BINSHISTOGRAM: cantidad de bins (intervalos) para elaborar el histograma
- LASTFRAME: cantidad de frames para el análisis de los tamaños de hormigas comenzando desde el último.
- LASTFRAMESTEP: intervalo en el que el intervalo LASTFRAME efectivamente extrae datos de las imágenes del video.
- MINCLUSTERSIZE: tamaño mínimo de cluster de una sola hormiga. Debajo de este valor se consideras impurezas del proceso de umbralización.
- MAXCLUSTERSIZE: tamaño máximo de cluster de una sola hormiga. Arriba de este valor se consideran todas manchas o colisiones de hormigas.
- ANTSPECIES: cantidad de normales que se quiere estimar a partir del histograma obtenido.
- CONFIDENCE: nivel de confianza estadístico para hallar el intervalo de confianza en la normal más alta. Es decir el intervalo del tamaño de lo que se considera una hormiga con un nivel de confianza CONFIDENCE.

Y en species se definen los nombre de las especies en estudio, los cuales están relacionados con una imagen homónima de extensión jpg en la carpeta images/species.

- NAMES: lista de nombre de las especies en estudio

Ejemplo de archivo Settings.js

```
{  
  "settings": {  
    "VIDEOFILES":      "../examples/",  
    "VIDEOSETTINGS":  "output/videoSettings/",  
    "VIDEOFRAME":     "output/videoFrames/",  
    "VIDEOPROCESS":   "output/videoProcess/mass&centermass/",  
    "VIDEOPATHS":     "output/videoProcess/Paths/",  
    "VIDEOPARAMETERS": "output/videoProcess/antParameters/",  
  }  
}
```

```

        "IMAGES":          "images/"
    },
    "defaultscan":{
        "BINSHISTOGRAM":   100,
        "LASTFRAME":      2000,
        "LASTFRAMESTEP":  20,
        "MINCLUSTERSIZE":  150,
        "MAXCLUSTERSIZE":  1000,
        "ANTSPECIES":     1,
        "CONFIDENCE":     0.999
    },
    "species":{
        "NAMES": ["Generic species", "Linepithema_humile", "Camponotus_mus"]
    }
}

```

Configuración particular - <nombre de video>.js

Para el análisis de los videos el usuario puede guardar la configuración que ha ajustado para obtener resultados más precisos. De esta forma puede almacenar la siguiente información en formato json

- filename: nombre del video.
- regionMask: lista de puntos que conforma la máscara que enmarca la región de interés.
- framesize: tamaño original de los frames de video.
- rect: mínimo rectángulo que engloba regionMask (boundingBox)
- frames: [primer frame de interés, último frame de interés, frame actual en la reproducción]
- outfileOpt: vector de unos y ceros que representa los checkboxes de la interfaz para definir qué salida se desea para el análisis del video, como por ejemplo la masa, centro de masa, trayectorias, etc.

- `binaryFnc`: nombre de la función de umbralización utilizada para la detección de hormigas.
- `antvalues`: tamaño mínimo, medio y máximo de lo que se considera una hormiga para ese video. Valores utilizados para discriminar que cluster de puntos adyacentes es o no una hormiga y cuantas hormigas hay dentro de una zona mayor al tamaño máximo de una de ellas.

Ejemplo de archivo <nombre de video>.js

```
{  
  "settings": {  
    "filename": "nuevo3.avi",  
    "regionMask": [  
      [597.9725275,312.0384615],  
      [594.0164835,379.2912088],  
      [791.8186813,648.3021978],  
      [1207.203297,652.2582418],  
      [1205.225275,52.91758242],  
      [795.7747253,46.98351648],  
      [597.9725275,312.0384615]  
    ],  
    "framesize": [720,1280],  
    "rect": [595,47,612,605],  
    "frames": [297,3285,281],  
    "outfileOpt": [0,0,0,0],  
    "binaryFnc": "subtract_with_empty",  
    "antvalues": {  
      "min": 36.9079,  
      "mean": 137.032,  
      "max": 260  
    }  
  }  
}
```

Configuración particular - <nombre de video>.stats.txt

En caso de que no basten los parámetros de aprendizaje estadístico por defecto comentados en el apartado defaultscan del archivo settings.js. El usuario puede guardar los parámetros que considere más adecuados así como también los resultados del proceso de aprendizaje.

Así contiene los parámetros de defaultscan además de los resultados del proceso de aprendizaje, es decir la información procesada para la construcción del histograma así como de las normales para, tener acceso inmediato a los resultados sin tener que reprocesar los últimos frames del video para ver esta información.

De este modo suma los campos

- data: información de los intervalos del histograma
- normals: parámetros de las normales estimadas a partir del histograma.

Ejemplo de archivo <nombre de video>.stats.txt

```
{
  "stats": {
    "ANTSPECIES": 2,
    "CONFIDENCE": 0.9999,
    "MINCLUSTERSIZE": 20,
    "MAXCLUSTERSIZE": 300,
    "BINSHISTOGRAM": 100,
    "LASTFRAME": 3000,
    "LASTFRAMESTEP": 30,
    "data": [53,37,35,33,119,133,27,65,32,...,190],
    "normals": {
      "meanV": [
        [111.224897],
        [137.0316531]
      ],
      "stdV": [
        [58.86423628],
        [26.9221042]
      ],
      "xx": [7.15,8.258,9.366,...,311.85]
    }
  }
}
```

3. Transformación de la información

Las entradas que alimentan al sistema son una serie de videos con agentes biológicos, puntualmente hormigas, que se desplazan en un espacio pre configurado. La principal tarea de AS consiste en obtener información de estos agentes, como por ejemplo las trayectorias de estos insectos, sus velocidades, el comportamiento del centro de masa total, etc. Para ello es necesario analizar el video y en este sentido el escenario base de análisis se centra en el procesamiento de imágenes que conforman un video.

A partir del análisis visual de los videos se observa que las condiciones en la que transcurren los videos es dinámica, por ejemplo las condiciones lumínicas pueden variar, la fuente de emisión de luz así como su intensidad no es constante, durante el transcurso de los videos existe la posibilidad de la aparición de sombras producidas por los encargados de realizar los videos, así como también la aparición repentina de objetos sólidos como pinceles, dedos humanos. Estas condiciones variantes y aleatorias complejizan la tarea del análisis de las imágenes y debido a esto se implementan distintas estrategias según el contexto del video luego de un análisis visual preliminar.

Destacamos que métodos de umbralización automáticos tradicionales como el método de Otsu, no son eficaces para el reconocimiento de los agentes debido a las variaciones mencionadas.

3.1 Procesamiento de imágenes

Dada la imagen de un frame resulta de interés determinar qué pixel de la imagen corresponde a una hormiga y cuál no. De este modo la primera transformación de la información consiste en pasar de una imagen en colores proveniente del video a una imagen binaria que muestre una primera diferenciación, es decir y por ejemplo 0 representa el cuerpo de un posible objeto y 1 el fondo de la escena. A este primer proceso lo llamaremos umbralización.

Debido a las condiciones dinámicas del contexto de los video, cada uno de ellos puede tener asociada una función de umbralización específica para obtener datos con el mayor nivel de precisión posible. Así se definen tres funciones predeterminadas principales, con la posibilidad de que el usuario agregue nuevas funciones implementando un tipo de función que reciba una imagen en colores y retorne una binaria.

Sintaxis de la función de umbralización:

function bw = binaryFunction(img)

Las funciones de umbralización reciben una imagen ya recortada sobre el área de interés del video. Todas consideran que el contexto del video es mucho más claro que el cuerpo de las hormigas, aunque cada función puede tener cualquier otro tipo de consideración al respecto.

Las funciones de umbralización predefinidas son:

- `white_acrylic_background.m`
- `white_acrylic_background_fast.m`
- `silver_background.m`
- `substract_with_empty_t<cota de umbralización>.m`
- `power_metal_150.m`

3.1.1 white_acrylic_background.m

Esta función toma en cuenta un fondo claro con agentes oscuros, soporta variación aleatoria de las condiciones de iluminación, así como degrada el cuerpo de los agentes para quitar detalles morfológicos de los mismos como los extremidades (patitas de las hormigas). Hecho que facilita el cómputo del proceso de aprendizaje posterior a la umbralización.

Pseudo código

- 1) pasar la imagen a escala de grises
- 2) aplicar un filtro topológico para eliminar la irregularidad de la luz
- 3) aumentar el contraste de la imagen
- 4) aplicar umbralización según el método de Otsu
- 5) aplicar la máscara parcial obtenida a la imagen original y quitar falsos positivos
- 6) quitar extremidades de los agentes

Con respecto a (5), lo que se hace es aplicar la imagen umbralizada hasta el momento, a la imagen original y eliminar puntos negros de la umbralización que no corresponden a un agente.

En relación a la performance, el hecho de utilizar un filtro topológico (2), aumenta significativamente al doble o tripe del tiempo de procesamiento que utiliza la función sin el filtro.

Algoritmo

```
function answer = white_acrylic_background( img )
    r=img(:,:,1);
    rev=255-r;

    % regularizar la iluminación
    i2=imtophat(rev,strel('disk',30));

    % ecualizar
    i3=imadjust(i2);
    % umbralizar con otsu
    threshold=graythresh(i3);
    i4=im2bw(i3,threshold);
    % primer filtro
    target= i4>0;
    img(target)=255;
    % segundo filtro con rojo
    target= img(:,:,1)==255 & img(:,:,2)<85 & img(:,:,3)<70;
    answer = zeros(size(i4));
    answer(target) = 1;
    answer = ~answer;
    answer = breakLegs(answer);
end
```

El proceso del filtro topológico es una implementación llamada `imtophat` y la eliminación de detalles de las hormigas es una implementación desarrollada por nosotros que se denomina `breakLegs`. Estas dos implementaciones constituyen parte vital del algoritmo.

La primera `imtophat` consiste en extraer el fondo de la imagen aislada sin agentes, lo cual dado un frame en el cual varía la condición de iluminación se obtiene por separado un fondo de degradé en escala de grises claro, que representa cómo varía la iluminación de una zona del video a otra. De este modo al restar ese fondo a la imagen original resulta evidente cual es el cuerpo de un objeto y todos los falsos positivos posibles son descartados en la resta del fondo aislado obtenido en el proceso del filtro.

Sintaxis:

```
IM2 = imtophat(IM,SE)
```

La segunda `breakLegs` consiste en detectar los bordes de todas las regiones de pixeles negros adyacentes y luego restar esos bordes a la imagen original, de esta manera, las extremidades al no tener una grosor significativo, es decir que suele tener un ancho de uno o dos pixeles, las extremidades de los agentes en la imagen resultante son eliminadas.

```
function bw = breakLegs( bw )  
    b=~bw;  
    bw=~(b-bwperim(b,8));  
end
```

La función `bwperim` en su primer parámetro recibe la imagen original y en el segundo, la cantidad de vecinos que considera para la detección de bordes, en este caso, 8 vecinos.

3.1.2 `white_acrylic_background_fast.m`

Es igual a `white_acrylic_background` pero no hace uso del filtro topológico, es decir que no se adapta a cambios de luz y por esta misma razón usa menos tiempo para realizar el proceso de umbralización.

3.1.3 `silver_background.m`

Es una evolución de `white_acrylic_background` su diferencia consiste en utilizar los filtros top-hat y bottom-hat para corregir el contraste de los frames del video. Esto es útil para videos con fondo oscuro.

3.1.4 `subtract_with_empty_t.m`

Esta función a diferencia de las anteriores no hace uso de los filtros topológicos. Aunque se basa en su comportamiento y es una técnica tradicional en el procesamiento de videos.

La misma consiste en tomar una imagen de la escena vacía de agentes biológicos y restarla en cada frame del video, de manera de obtener únicamente información respecto de cuerpos

que presenten variaciones respecto del fondo de la escena. Una vez que se obtiene esta información en escalas de grises se procede a realizar un proceso de umbralización de acuerdo a un cota, la cual viene a estar representada por el sufijo en el nombre de la función, es decir si esta se llama `subtract_with_empty_t20`, significa que una vez que resta la imagen vacía al frame actual, luego umbraliza la imagen con la cota 20.

Para esto el usuario puede tomar una imagen del video que se presente vacía y guardarla a través de un botón de la interfaz llamado `Takeemptypic!`. El mismo toma una imagen de la escena recortada según el área de interés seleccionada sin el formato de la representación del fondo en celeste.

En caso de no existir un frame vacío, es decir sin agentes, se han obtenido muy buenos resultados tomando un frame casi vacío y borrar las hormigas con una herramienta muy básica de procesamiento de imagen como Paintde Microsoft Windows.

Estas imágenes se almacenan en la carpeta `output/videoFrames` y tienen el sufijo `.empty.png`. La misma tiene la características de ser única para cada video, es decir que si el usuario toca reiteradas veces el botón mencionado sustituye esta imagen.

Desde el lado de la interfaz visual para hacer uso de este filtro se selecciona de la lista de `Binaryimagefunction` la función de umbralización denominada `subtract_with_empty_t<cota de umbralización>`. La misma de encontrarse la imagen sin agentes la utiliza para realizar el proceso de umbralización, de lo contrario no realiza ningún tipo de cambio sobre la imagen.

Las ventajas de esta función de umbralización son principalmente su característica genérica, ya que no importa las características del contexto de la escena, la misma es eliminada por el proceso de resta de imágenes, y por otro lado al ser una operación de resta de matrices y no hacer uso de ningún tipo de filtro topológico, el tiempo de procesamiento es significativamente menor.

Por otra parte es recomendable este tipo de función para videos donde no cambie dinámicamente las condiciones de iluminación durante el mismo, ya que la iluminación se considera constante dado que la misma es registrada en la imagen sin agentes y es eliminada en cada frame tomando como premisa que las condiciones de iluminación o encuadre no varían.

Algoritmo

```
functionbw = subtract_with_empty( img )

    globalemptyFrame;

    if(~isempty(emptyFrame))
        b=img;
```

```

        c=emptyFrame-b;
        ants=c>20;                % cota de umbralización
        bw=ones(size(c))*255;
        bw(ants)=0;
        bw=im2bw(bw);
    else
        bw=img;
    end

end
end

```

Una vez que se restan las imágenes, el proceso de umbralización es a partir de un umbral determinado empíricamente, en este caso es 20, y está comentado en la función como cota de umbralización.

La salida del proceso de umbralización ejemplificado en estas tres funciones constituye la entrada de un segundo proceso de discriminación de información, dado que la umbralización ante la adyacencia de hormigas o ante imperfecciones propias del proceso de umbralización no hace diferencia entre estos casos. Esta segunda etapa es cubierta en la sección 4.

3.1.5 power_metal_150.m

Se basa en `subtract with empty` pero antes de umbralizar ecualiza la imagen obtenida.

A partir del uso de la aplicación con dos clases de videos uno con la especie *Linepithema humile* (hormigas marrones más pequeñas) y otros con la *camponotus mus* (hormigas más grandes negras), se han obtenido los mejores resultados con `subtract_with_empty_t35` y `white_acrylic_background` y respectivamente.

3.2 Procesamiento de videos

Ahora bien supongamos que a partir del proceso de umbralización y producido el proceso de aprendizaje detallado en la sección 4 se logra determinar qué región en la imagen es una hormiga, algo que podría deducirse de esta información es el centro de masa de estos cuerpos y tomarlo como indicador principal de la posición de la hormiga en el frame actual. Es decir, una segunda transformación de información.

Luego entre dos frames consecutivos podrían compararse con algún tipo de métrica qué puntos que representan los centros de masa de las hormigas corresponden al mismo agente por la cercanía entre frames consecutivos, considerando para esto una región máxima en la cuál una hormiga puede desplazarse. Esta región máxima se ha tomado como el 8 % del tamaño medio de una hormiga.

De esta forma se utiliza la distancia eucladiana como métrica de esta correspondencia de centros de masa en frames consecutivos para deducir la trayectoria de un agente dado. Además se enumeran a estos agentes de modo de almacenar estos centros de masa como elemento constitutivo de sus trayectorias a lo largo del video. Y a esta tercer transformación se la denomina proceso de seguimiento o tracking.

En este proceso se tiene en cuenta que un agente puede ingresar o salir de la zona de estudio, con lo cual ante la aparición de un nuevo agente comienza una nueva trayectoria con una etiqueta nueva o bien puede finalizar el seguimiento hasta el momento.

Este algoritmo de seguimiento sirve tanto para el seguimiento de algunos o todos los agentes en el video. Para esto, a través de la interfaz visual el usuario puede seleccionar las hormigas de interés individualmente o bien activar un checkbox para que se detecten todas las nuevas hormigas y se realice el proceso de trackeo.

En el código fuente de la aplicación este proceso se especifica en el archivo trackAnts.m.

4. Aprendizaje automático

Luego del proceso de umbralización resulta de interés determinar con cierta precisión qué objetos de los reconocidos son agentes de interés o no.

Para esto se utilizan diversas técnicas y estrategias que son complementarias y que no se ejecutan simultáneamente, para lograr discriminar esta información.

De esta forma se utilizan herramientas de clusterización en primer lugar para determinar que regiones adyacentes de pixeles contiguos que forman posibles cuerpos y luego una reclusterización de esas regiones si estas exceden el tamaño máximo de lo que se considera una hormiga.

Por otro lado se hace un estudio estadístico del tamaño de estas regiones contiguas para extraer el intervalo de tamaño válido de una hormiga en ese video.

4.1 Detección de hormigas

El proceso de detección de hormigas consiste en una combinación de técnicas de clustering. En primer lugar una clusterización de las regiones adyacentes dada una imagen umbralizada, y luego una reclusterización en el caso de que las regiones posean más de un agente.

La primer a través de `bwconcomp` y `regionprops`, y la segunda con una versión adaptada de k-means denominada fuzzy-c-means (`fcm`).

El primer proceso está más ligado al tratamiento de imágenes y consisten en a partir de una imagen umbralizada obtener regiones contiguas de pixeles negros, etiquetarlas y obtener información de esas regiones, como por ejemplo su perímetro, la cantidad de pixeles que componen la región, es decir su área, del mismo modo su centro de masa, características morfológicas de la región, lista de pixeles que componen la región, etc.

Para esto se hace uso de `bwconncomp` y `regionprop`, la primera de ellas es una posibilidad entre tantas otras funciones como por ejemplo `bwlabel`, pero a diferencia de ésta `bwconncomp` es más eficiente porque no es un método iterativo y analiza la imagen una única vez. La segunda es la encargada de obtener información de estas regiones como por ejemplo su área, centro de masa, y la lista de pixeles que conforman las regiones.

Supongamos que se conoce un intervalo de tamaño fiable en el cual, todo aquello que esté dentro de ese intervalo es con certeza una hormiga y el resto no. Aquello que es menor representa imperfecciones del primer proceso de umbralización y toda región mayor es una aglomeración de agentes que se encuentra adyacente, es decir una colisión de agentes.

Con este intervalo se puede discriminar las regiones que identifican a un agente, y registrar sus datos en el sistema, mientras que a partir de esta diferenciación se puede conocer además cuáles son las regiones que representan colisiones entre agentes.

4.2 Resolución de colisiones

Para depurar la información provista por la adyacencia de agentes se utiliza fuzzy c means, que se basa en k-means, pero que permite a un mismo punto, es decir un pixel, tener un grado de pertenencia a distintos clusters y de esta forma obtener mayor precisión a la hora de computar el centro de masa de los agentes en la colisión.

El método fcm es un método iterativo, que converge a los centros de masa que representan los centroids de los agentes colisionados y por lo tanto costoso computacionalmente.

Para obtener la cantidad estimada de agentes lo que se realiza es dividir el área total del área con agentes colisionados por el tamaño promedio de un agente. Este valor se obtiene por un proceso estadístico que es anterior al de clusterización descrito en la sección 4.3.

Así con la lista de puntos que integra una región de colisión de hormigas y la cantidad estimada de agentes en la misma, se reclusteriza esa zona con el método fcm.

Dado el costo de este proceso de reclusterización se toman las siguientes decisiones de diseño:

1. evitar en la medida de lo posible la reclusterización
2. degradar la cantidad de pixeles que forman la entrada de f_{cm} .

La primera decisión se ve implementada en una de las funciones de umbralización `white_acrilic_background`, que una vez umbralizada la imagen de entrada, degrada los bordes de las regiones eliminando las extremidades de los agentes y de esta forma disminuyendo los puntos de contacto, es decir evitando la colisión por el contacto de extremidades cuando los agentes están próximos.

La segunda tiene que ver con disminuir el tiempo de procesamiento del proceso de reclusterización, ya que es proporcional al tamaño de la entrada. Para esto lo que se hace es elegir una cierta cantidad de pixeles, un 5 % del total de pixeles que conforman la región y eliminarlos de la misma, asumiendo que su ausencia no afecta la morfología de los miembros adyacentes.

Para esto se eligen un 5% de los pixeles totales, se selecciona igual cantidad de números aleatorios con distribución uniforme a la cantidad de pixeles y se descartan aquellos que tenga una probabilidad de ocurrencia menor al 5 %.

En el código fuente de la aplicación este proceso se especifica en el archivo `countAnts.m`

4.3 Aprendizaje de tamaño de los agentes

A partir del análisis visual de los videos, y en especial el primer conjunto de videos en los que se utiliza citronela para repeler a las hormigas Ref.[1], se observa que a medida que transcurre el video las hormigas van desalojando la zona, y de esta manera la cantidad de colisiones es menor dado que las hormigas tienen mayor espacio para desplazarse.

Con esta observación la estrategia que se toma para el aprendizaje del tamaño de una hormiga es analizar los últimos n frames del video en los cuales las hormigas se presentan disjuntas mayormente y analizar esos frames con el primer proceso de clusterización explicado en la sección 4.1.

Una vez obtenida esa información de las regiones encontradas en los últimos frames del video se elabora un histograma para hallar un intervalo adecuado que represente con certeza al tamaño característico de una hormiga en ese video.

Para esto el usuario cuenta con una interfaz específica donde puede especificar tamaños mínimos y máximos de las regiones de interés para acotar la búsqueda, así como también la cantidad de frames, y el paso entre cada frame del video para obtener esta información.

Estos parámetros varían de acuerdo a cada video, y es responsabilidad de quien esté utilizando AS de poner parámetros adecuados. Los mismos pueden ser almacenados en archivos de configuración para ser recobrados en otros análisis o bien para ser estudiados de forma aislada.

A raíz de este análisis surge evidencia clara de que el tamaño de los agentes tiene una distribución normal, con lo cual se utilizan técnicas de estimación de normales denominadas EM (ExpectationMaximization) para estimar los valores de los tamaños de los agentes. Los parámetros que puede modificar el usuario para obtener los intervalos de tamaños adecuados, además de los ya mencionados, son la cantidad de normales que se podrían extraer del histograma así como el nivel de confianza para obtener finalmente cotas máximas, mínimas y la media del tamaño del cuerpo de una hormiga.

En el proyecto las librerías están en `test/em_1dim.m`

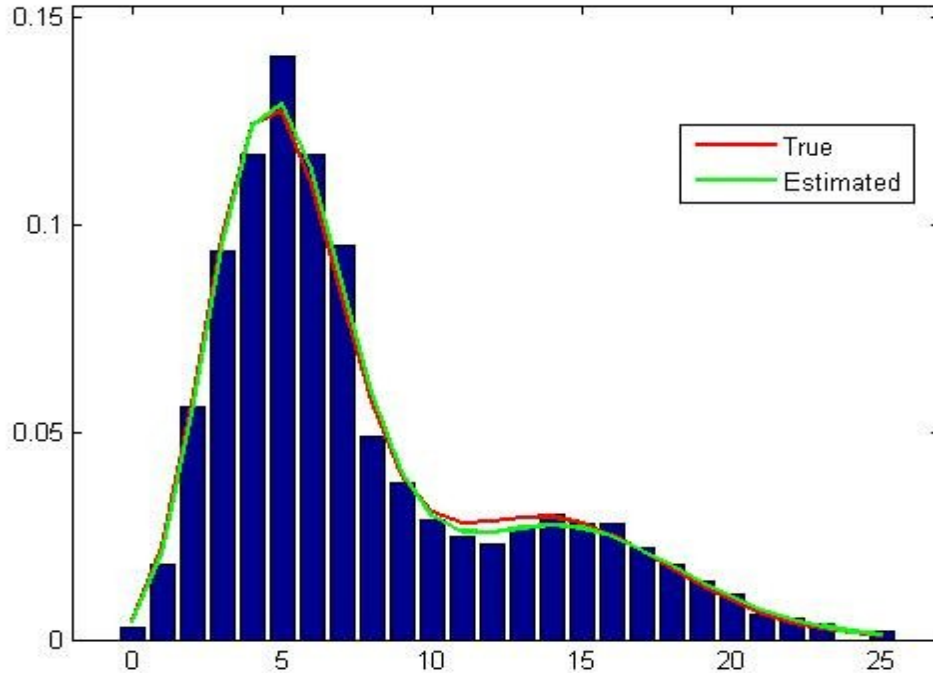


Figura 1 - Funcionamiento del método EM

La estimación de las normales es una primera aproximación y que el usuario puede ajustar estos valores de una forma supervisada con la interfaz. A través de herramientas de análisis de gráficos para escoger valores tal vez más adecuados según su criterio.

De esta forma el usuario investigador puede obtener un intervalo de confianza par poder discriminar la información provista a partir de la umbralización.

En el código fuente de la aplicación este proceso se especifica en la carpeta `src/processAnts/statistic`.

5. Algoritmos utilizados

Se desarrollaron algoritmos propios para el proceso de tracking haciendo uso de funciones ya implementadas para manejo de imágenes, videos y herramientas de estudio estadístico.

Lo más significativos serán cubiertos en esta sección.

1. Implementaciones
 - a. Clusterización por adyacencia
 - b. Fuzzy c means

2. Desarrollo propio
 - a. Conteo de hormigas
 - b. Seguimiento de hormigas

5.1 Clusterización por adyacencia

Se comentó en la sección llamada Detección de hormigas el uso de estos algoritmos, esta vez se explica el modo en el cual funcionan `bwconncomp` y `regionprop`.

Se clasifica `abwconncomp` dentro del paquete `MorphologicalOperations` y a `regionprop` dentro de `ImageAnalysis and Statistics`.

5.2 Algoritmo `bwconncomp`

1. Buscar el próximo pixel desetiquetado, `p`.
2. Usar el algoritmo flood-fill para etiquetar todos los pixeles directamente conectados que contienen `p`.
3. Repetir los pasos 1 y 2 hasta que todos los pixeles hayan sido etiquetados.

5.3 Algoritmo Flood-Fill

Sintaxis: `Flood-fill(node, color objetivo, color de reemplazo)`

1. Si el color del nodo no es igual al color objetivo, retornar.
2. Poner el color al nodo del color de reemplazo
3. Flood-fill (un paso a la izquierda del nodo, color objetivo, color de reemplazo)
4. Flood-fill (un paso a la izquierda del nodo, color objetivo, color de reemplazo)
5. Flood-fill (un paso a la izquierda del nodo, color objetivo, color de reemplazo)
6. Flood-fill (un paso a la izquierda del nodo, color objetivo, color de reemplazo)
7. Retornar

5.4 Ejemplo de funcionamiento

Sea la imagen binaria BW

```
BW = logical ([1 1 1 0 0 0 0 0
               1 1 1 0 1 1 0 0
               1 1 1 0 1 1 0 0
               1 1 1 0 0 0 1 0
               1 1 1 0 0 0 1 0
               1 1 1 0 0 0 1 0
               1 1 1 0 0 1 1 0
               1 1 1 0 0 0 0 0]);
```

```
L = bwconncomp(BW)
```

L =

```
1 1 1 0 0 0 0 0
1 1 1 0 2 2 0 0
1 1 1 0 2 2 0 0
1 1 1 0 0 0 3 0
1 1 1 0 0 0 3 0
1 1 1 0 0 0 3 0
1 1 1 0 0 3 3 0
1 1 1 0 0 0 0 0
```

Por otra parte `regionprops` a partir de la información obtenida de `bwconncomp` obtiene métricas de las regiones encontradas.

Shape Measurements

'Area'	'EulerNumber'	'Orientation'
'BoundingBox'	'Extent'	'Perimeter'
'Centroid'	'Extrema'	'PixelIdxList'
'ConvexArea'	'FilledArea'	'PixelList'
'ConvexHull'	'FilledImage'	'Solidity'
'ConvexImage'	'Image'	'SubarrayIdx'
'Eccentricity'	'MajorAxisLength'	
'EquivDiameter'	'MinorAxisLength'	

Tabla 1 - Posibles métricas que puede obtenerse a partir de `regionprop`

5.5 Fuzzy c mean (fcm)

Este algoritmo es una extensión de k-means, como tal es un algoritmo iterativo al cual hay que definirle una condición de corte.

Su diferencia con k-means es que cada punto puede pertenecer a más de un cluster, es decir que puede tener un grado de pertenencia a cada cluster y como consecuencia tener mayor precisión en la obtención de los centros de masa ante una colisión porque en esos casos algunos pixeles pueden tener este comportamiento, es decir ser pixeles que representan la superposición de una sección del cuerpo entre más de una hormiga. Por el otro lado este cómputo degrada la performance de ejecución en comparación con k-means donde cada punto pertenece únicamente a un cluster.

Algoritmo

1. Inicializar la matriz $U = [u_{ij}]$, $U_{(0)}$
2. En k-paso: Calcular los vectores de centros $C(k)=[c_j]$ with $U_{(k)}$
3. actualizar $U_{(k)}$, $U_{(k+1)}$
4. Si $|| U_{(k+1)} - U_{(k)} || < \xi$ entonces FIN; en otro caso volver a paso 2.

fcm comparte la función objetivo de k-means con la diferencia del coeficiente u_{ij} que es el grado de pertenencia de un punto a un cluster.

$|| * ||$ puede ser cualquier norma que exprese la similaridad entre los puntos y los centros.

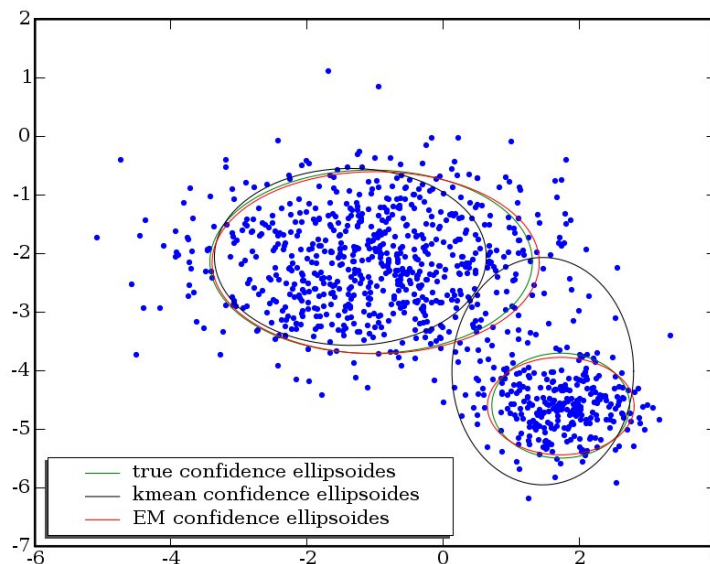


Figura 4 - Ejemplo de funcionamiento de fcm sobre datos arbitrarios

5.6 Conteo de hormigas

La función de conteo de agentes tiene la siguiente definición:

```
function [qAnt swarm antCluster]=countAnts(bw,antvalues,handles)
```

Recibe como parámetro:

- Bw - una imagen umbralizada
- Antvalues - los valores estimados del tamaño de la hormiga
- Handles - handler de la interfaz visual

Retorna:

- qAnt - cantidad de hormigas en la imagen
- swarm - conjunto de esas hormigas y sus propiedades
- antCluster - conjunto de clusters de más de una hormigas

Pseudo código:

1. Filtrar la imagen según los tamaños estimados de agente
2. Para cada uno de ellos
 - a. obtener propiedades de interés (masa, centro de masa)
 - b. agregarlos a swarm
3. Para los cluster que exedan el tamaño de un agente
 - a. agregarlos antCluster
 - b. degradar la cantidad de puntos totales a reclusterizar
 - c. reclusterizar los bloques según el tamaño medio de una hormiga
 - d. agregar estos agentes a swarm
 - e. estimar valores de masa

Asignar a todos los miembros de swarm id=0

Asignar la cantidad de agentes a qAnt

5.7 Seguimiento de hormigas

La función de seguimiento de agentes tiene la siguiente definición:

```
function [swarm lostantantCluster,qAnt]=  
trackAnts(selected,bw,dna,handles,currentFrame,antvalues)
```

recibe como parámetros

- selected - colección de hormigas de interés
- bw - imagen del frame actual umbralizada
- dna - flag (Detect new ants)
- handles - handler de la interfaz visual
- currentFrame - número de frame actual

- antvalues - valores estimados del tamaño de agente

retorna

- swarm - conjunto de agentes que están en el frame actual
- lostant - agentes que han desaparecido en este frame
- antCluster - conjunto de más de un agente
- qAnt - cantidad de agentes en el frame

pseudo código:

1. contar las hormigas en el frame actual
2. para cada una de las hormigas de interés
 - a. buscar su correspondiente en el frame actual en base a su cercanía (distancia eucladiana)
 - b. en caso de encontrar una correspondencia
 - i. agregar su centro de masa como próximo paso en su trayectoria histórica
 - ii. conservar el id histórico
 - c. si no el agente ya no está en este frame
 - i. agregarlo a lostAnt
3. sidnaestáactivado
 - a. para cada una de los agentes que no son de interés inicializar su seguimiento con un id nuevo

6. GUI – Interfaces visuales

El usuario puede interactuar con el sistema de varias formas. A través de archivos de configuración, archivos de salida, sin embargo para poder obtener esta información primero debe entrar en contacto con una interfaz visual tipo desktop, stand alone elaboradas para el presente proyecto.

De esta forma se cuenta con tres secciones principales:

- Ant Swarm(ver figura 5)
- Process Center (ver figura 6)
- AntParameters (ver figura 8)

Para acceder a cualquiera de estas interfaces el procedimiento es abrir el archivo de extensión .m correspondiente a la sección y luego hacer click en el botón Run, señalado

con botón típico de play de color verde. Todos estos archivos se encuentran en la carpeta denominada `gui`.

6.1 Ant Swarm

Descripción: sección principal para el análisis interactivo de los videos de las hormigas.

Funcionalidad principal:

- Abrir y reproducir algún video
- Selección de un área de interés.
- Selección de hormigas.
- Establecer parámetros de configuración y almacenarlos a disco.
- Provee de una consola para obtener información de las hormigas que egresan de la zona de estudio.

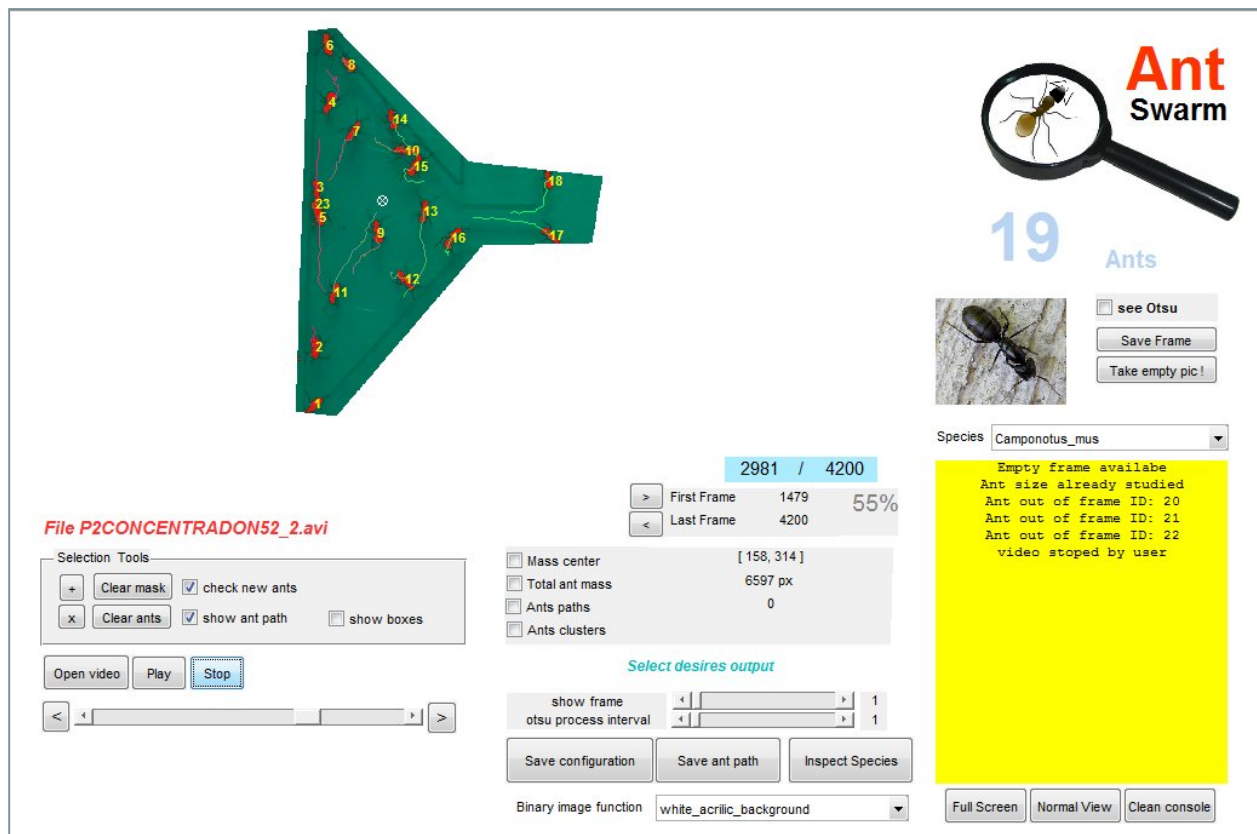


Figura 5 – Ant Swarm – interfaz principal

6.2 Process center

Descripción: una vez que se establecen los parámetros de configuración adecuados para cada uno de los videos, sería de esperar poder ejecutar el proceso de análisis de forma iterativa por cada uno de los videos que se hayan configurado, de la misma forma de tener una representación visual, del estado de los mismos. Es decir si están listos para ser analizados ya que se cuenta con un archivo de configuración disponible con la nomenclatura correcta, o bien si ya se cuenta alguna información extraída del mismo.

Por esta razón existe esta interfaz para cumplir con esas necesidades.

Los estados disponibles son:

- - No se ha configurado el video
- READY Se cuenta con un archivo de configuración
- DONE Se dispone de algún tipo de análisis ya realizado

Funcionalidad principal:

- ver todos los videos disponibles para analizar y conocer su estado en el sistema
- ejecución del procesamiento iterativo para un conjunto de videos seleccionados

Archivo: ProcessCenter.m

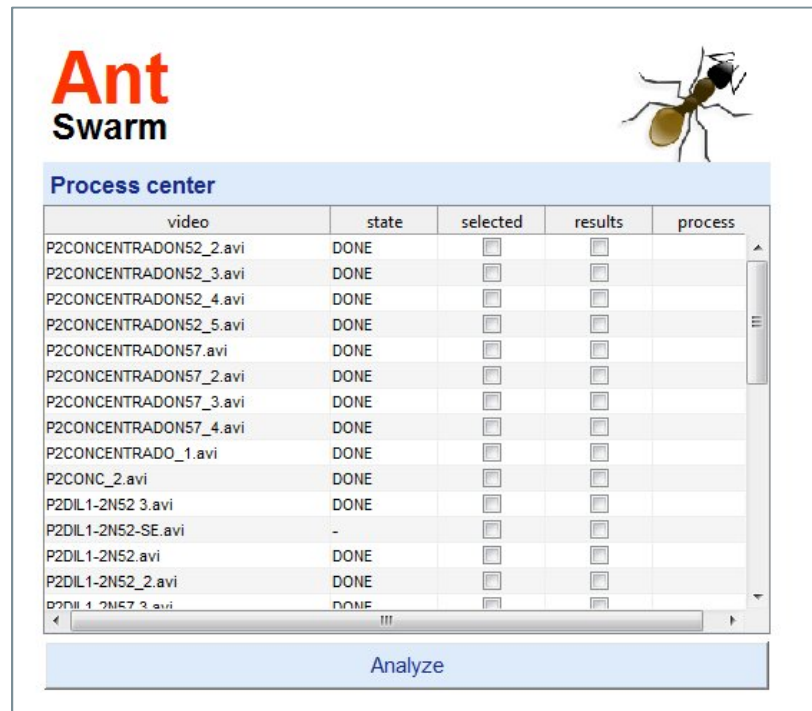


Figura 6 - Process center

6.3 AntParameters

Descripción: dada la necesidad de estudiar, parametrizar y ser precisos a la hora de elegir un intervalo adecuado para el tamaño de lo que se interpreta como una hormiga. Se desarrolla una tercer interfaz, que tiene como finalidad facilitar el estudio de su tamaño.

Funcionalidad principal:

- definir valores adecuados para investigar el intervalo adecuado de hormiga
- guardar esa información para futuros análisis
- guardar los resultados del análisis
- sugerir al usuario un intervalo adecuado
- que el usuario puede definir manualmente el intervalo que considere adecuado

En segundo lugar dado que el proceso de recolección de información puede ser costoso en el sentido de analizar una porción representativa del video, es que se cuenta con una barra de estado del proceso para informar al usuario el avance del mismo. Por esta misma razón es que el usuario puede guardar los resultados del análisis para ejecutarlos una única vez.

Archivo: `guitest.m`

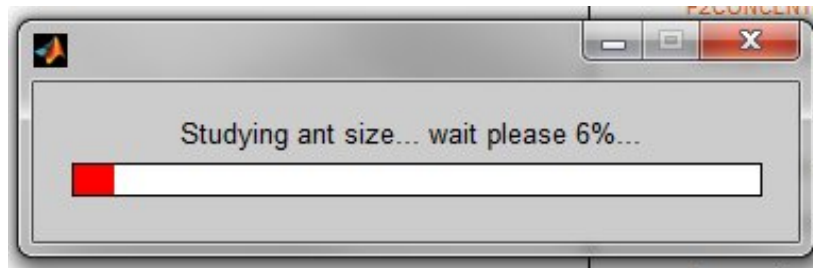


Figura 7 - Proceso de recolección de información

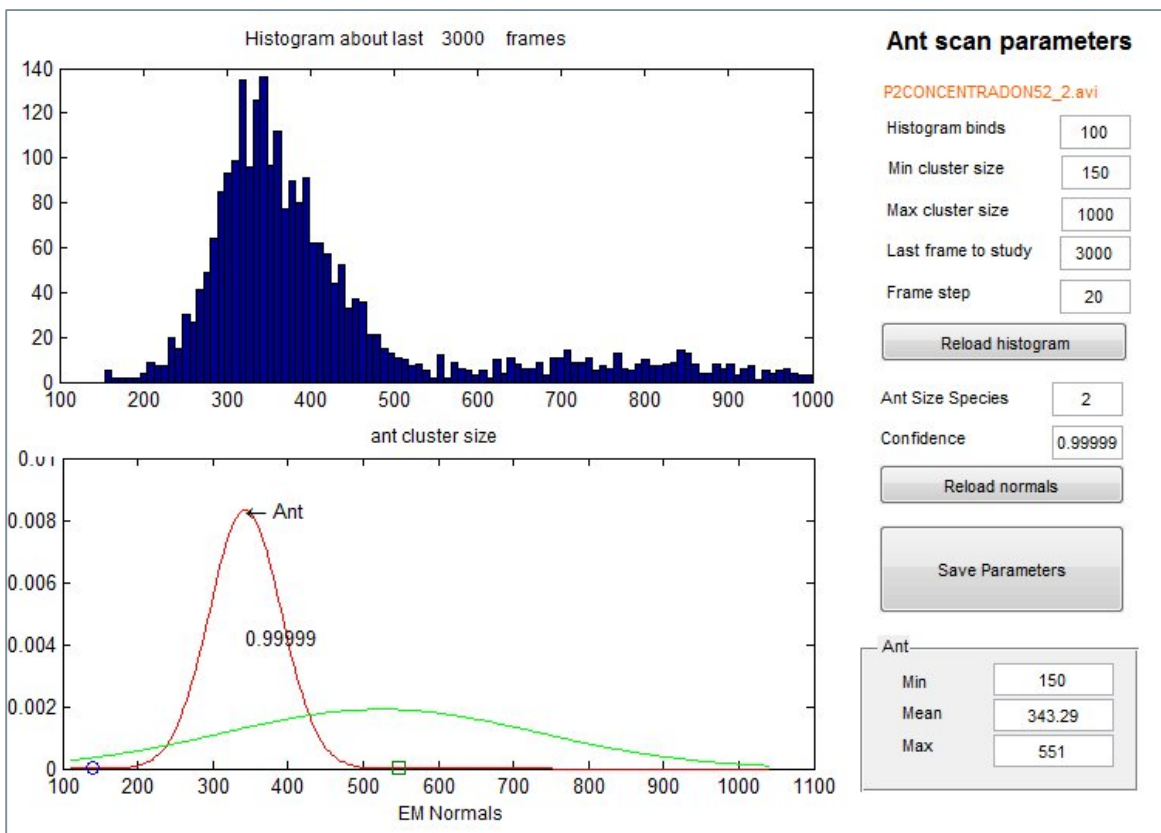


Figura 8 -AntParameters

7. Archivos de salida

El usuario es el encargado de seleccionar los tipos de datos a extraer de cada video, para esto puede elegir entre:

- Masa total
- Centro de Masa
- Trayectoria de Hormigas
- Clusters de hormigas

Los formatos de estos archivos son ASCII de texto plano, para facilitar el análisis y la independencia de estos datos respecto de la aplicación.

A continuación se describe la información suministrada así como el formato de salida de cada uno de los archivos mencionados.

7.1 Centro de masa

Descripción	Coordenadas del centro de masa del conjunto completo de hormigas en función de tiempo. El archivo es una secuencia de líneas que tienen el formato especificado más abajo. Si bien las coordenadas en una imagen son coordenadas de pixeles y por lo tanto discretas, los valores informados en los archivos de salida son número reales dado el cálculo del centro de masa, que no necesariamente pertenece al ámbito discreto.
Ubicación	<code>output/videoProcess/mass&centermass</code>
Formato	<code><número de frame>, <x>, <y></code>

7.2 Masa total

Descripción	Masa total de hormigas de la escena en función de los frames. Los valores de la masa total son la cantidad de pixeles negros que se identifican como hormigas. La cantidad estimada de hormigas se calcula según la descripción de la
-------------	---

	<p>sección de detección de hormigas.</p> <p>El archivo se compone de una secuencia de líneas que tienen el formato abajo mencionado.</p>
Ubicación	output/videoProcess/mass¢ermass
Formato	<número de frame>, <cantidad de pixeles totales>, <cantidad estimada de hormigas>

7.3 Trayectoria de hormigas

Descripción	<p>Recorrido de cada una de las hormigas detectadas en el video. A diferencia del centro de masa esta vez las coordenadas son aproximadas a valores enteros con la función round.</p> <p>El área de cada hormiga también es un valor discreto, sin embargo para las hormigas que provienen de una colisión, se asigna el valor medio calculado en el proceso de aprendizaje de tamaño, que generalmente es continuo.</p> <p>Esto ayuda a tener una segunda lectura de la información del área, es decir conocer la procedencia de la hormiga en el archivo de salida.</p>
Ubicación	output/videoProcess/paths
Formato	<id hormiga>, <número de frame>, <x>, <y>, <cantidad de pixeles del cuerpo de la hormiga>

7.4 Cluster de hormigas

Descripción	Clusters mayores a una hormiga en función del tiempo.
Ubicación:	output/videoProcess/mass¢ermass

Formato	<número de frame>, <cantidad total de pixeles>, <x>, <y>, <cantidad estimada de hormigas> (por cada cluster)
---------	---

7.5 Uso de la información

Para hacer uso de la información almacenada en estos archivos desde la consola se puede utilizar el comando `load`. Por ejemplo el siguiente comando realiza y muestra un gráfico de la evolución de la masa total de las hormigas en función del tiempo.

```
m=load('output\videoProcess\paths\P2CONCENTRADON52_2.avi.path.txt');
plot(m(:,1),m(:,2),'--b');
```

En el caso de las imágenes se puede hacer uso del comando `imread` e `imshow`. Por ejemplo para abrir una imagen y verla, se puede utilizar el siguiente comando.

```
i=imread('umb2\932-main.png');
imshow(i);
```

8. Indicadores

Los archivos de salida son los que tienen correspondencia con los indicadores para el estudio del comportamiento de los agentes. Aunque también hay indicadores como por ejemplo el de blackout que ayuda a depurar la información en función de la identidad de las trayectorias, espacio transitado, o velocidad que se deducen de los indicadores solicitados en los objetivos del proyecto.

A partir de unas primeras salidas de la aplicación se puede verificar la precisión de la muestra de los indicadores a través de un análisis visual de los mismos.

Es significativo mencionar que para el análisis visual de los resultados resulta de interés evaluarlos en una imagen que posea las mismas dimensiones de escenario de estudio y no en un gráfico convencional, dado que el último pierde la relación de proporción y distorsiona la información. Así por ejemplo para el análisis de las trayectorias hay dos tipos de gráficos uno que refleja la proporción real del escenario como en las imágenes de la Figura 9, y gráficos (plots) que ayudan a individualizar las trayectorias de los agentes de un modo más flexible, como por ejemplo colorando las trayectorias por agente e indicando referencias como en el caso de la Figura 11, aunque hay que tener en cuenta que este tipo de representación con facilidad puede perder la proporción real del escenario de estudio.

Si bien en la actualidad la interfaz visual no permite extraerlos de un modo interactivo, existe un script semi automático que los provee. El mismo se llama `seepathstest.m` y se encuentra en la carpeta `test`. Para conseguir los gráficos a partir de los archivos de salida de la aplicación se debe abrir el archivo `seepathstest.m` y modificar manualmente el nombre del video de interés. Luego correr el script con el botón Run y si no hubo problemas intermedios los gráficos de salida se almacenarán en la carpeta `output/videoProcess/images` donde cada uno de los gráficos de salida tendrán de prefijo el nombre del video y como sufijo una mínima descripción del contenido.

Así los posibles archivos tienen la siguiente definición:

`<nombre del video>.V<número de versión>.`

`[cm_only | cm_paths | color_path | color_path_histo | paths_length | total_ants | total_mass | velocity]`

`.png`

8.1 Centros de masa

Indicador de cómo se desplaza en conjunto de agentes en su totalidad.

Durante la ejecución del video este indicador se representa a través de un círculo con una cruz en el medio de color blanco.

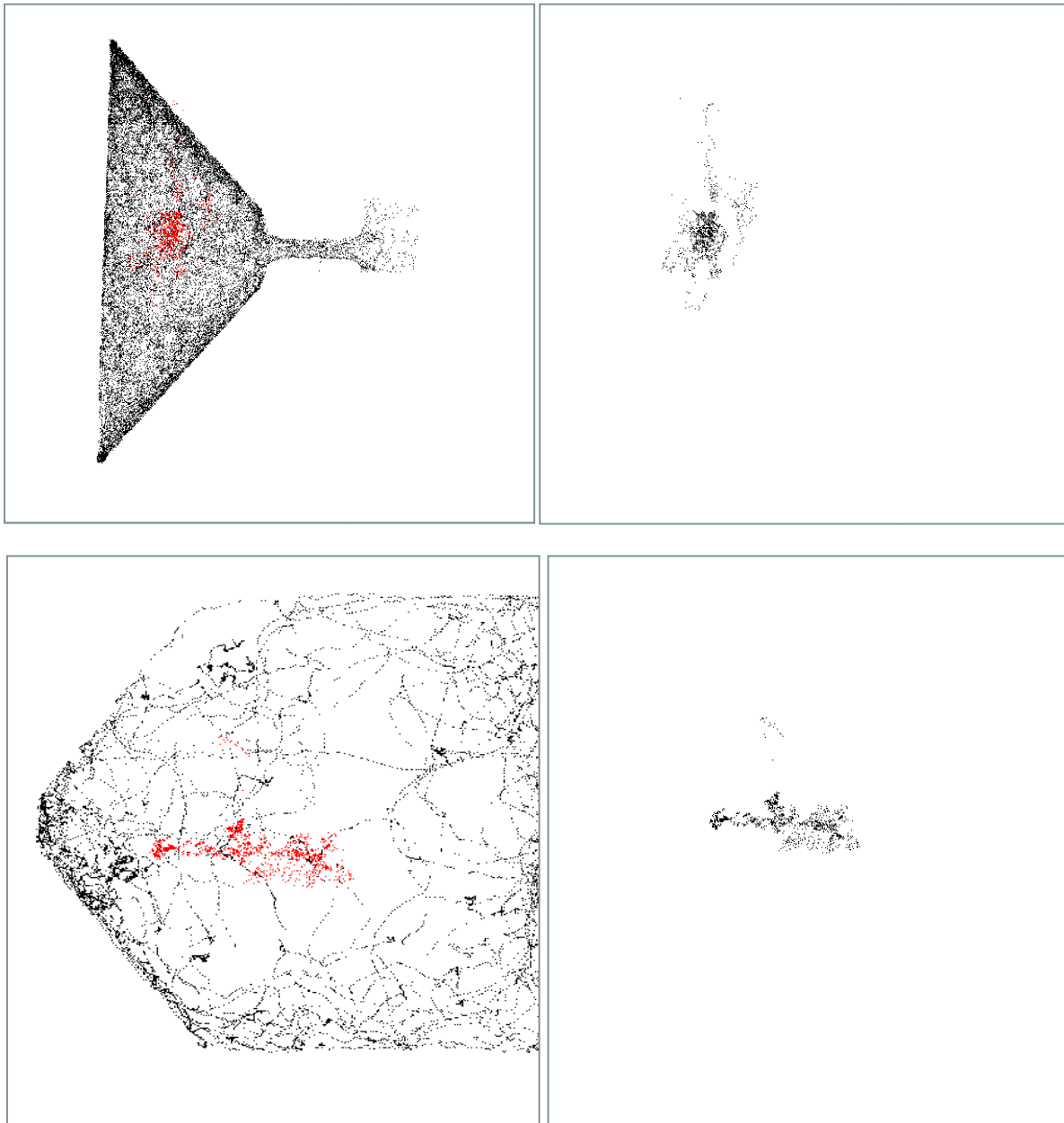


Figura 9- Ejemplos del indicadores de centro de masa.

Arriba videos con hormigas más grandes y abajo con unas más pequeñas

A la izquierda en negro la trayectoria de todas las hormigas y en rojo la trayectoria del centro de masa, a la derecha sólo la trayectoria del centro de masa

8.2 Masa total

Indicador de cómo se desplaza el conjunto de agentes en su totalidad, por ejemplo como evacua una determinada zona.

Esta métrica ayuda tanto a reflejar el ritmo en el cual los agentes evacuan una zona así como a ver la calidad en el que se extrae la información de los mismos. Es decir que el primer proceso de umbralización al menos tiene poco ruido. Y que por otra parte durante la ejecución de un video no hubo situaciones patológicas como por ejemplo la aparición de un objeto extraño o una sombra que altere la masa total. Como se evidencia en la Figura 10.

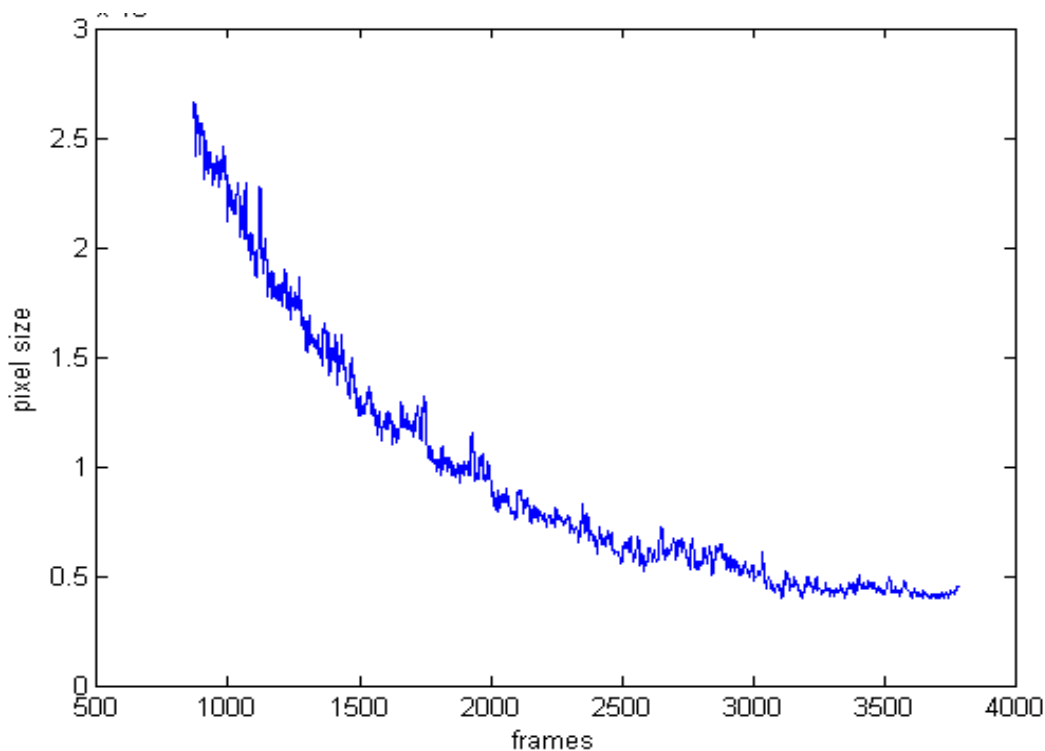


Figura 10 - Ejemplo del indicador de masa en una situación de evacuación

8.3 Trayectoria de hormigas

Una vez obtenida las trayectorias de los agentes por la aplicación es posible visualizarlas de un modo discriminado a través de gráficos con referencias. Como por ejemplo en la Figura 11. Y a partir de los mismos tener una primera impresión sobre el desplazamiento de los mismos.

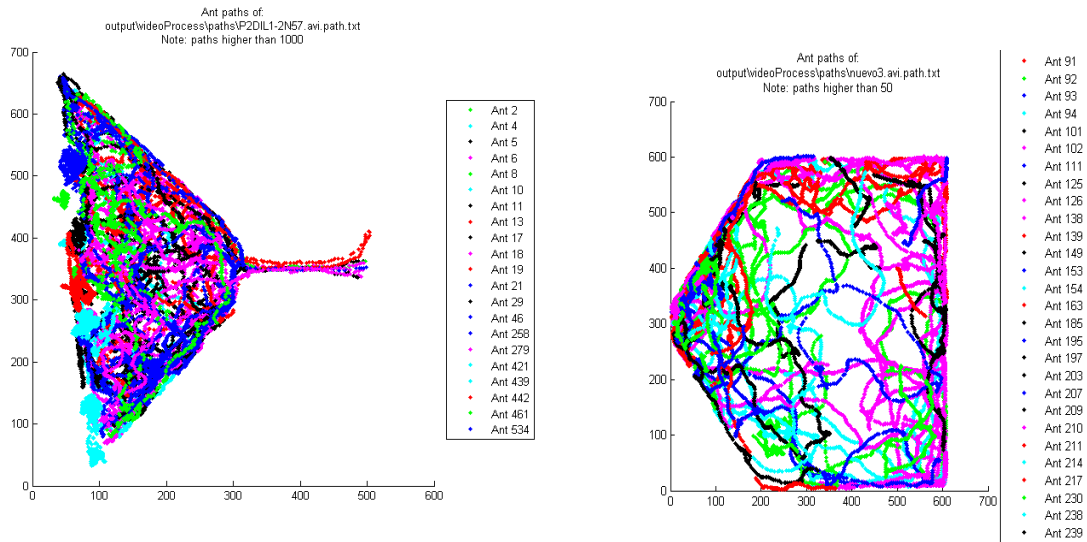


Figura 11 - Trayectorias individualizadas por hormiga para videos con especies diferentes. A la izquierda hormigas grandes y a la derecha hormigas de menor tamaño.

8.4 Black out del seguimiento

Durante el análisis del seguimiento de los agentes dada las condiciones en las que fueron filmados los videos, las irrupciones de objetos extraños, sombras, y además las aproximaciones de los agentes en base a su tamaño existen imprecisiones y un fenómeno que denominamos blackout en el cual se pierde la identidad de los agentes.

Este fenómeno produce un cierre de la trayectoria del agente hasta el momento y ante la detección de un nuevo insecto ya sea porque ingresa repentinamente a la escena o es el mismo agente que por la irrupción de una condición de blackout se perdió su identidad, se asigna una nueva numeración como si fuese uno nuevo.

La condición de blackout corresponde a la perdida de información de una hormiga en particular y no a un frame del video en particular, es decir que si en una misma escena hay agentes que conservan información respecto de su trayectoria las mismas son almacenadas, mientras que las que se vean afectadas por algún factor son discontinuadas.

El mismo comportamiento puede suceder en el caso de que se divida un cluster varios agentes con un número mayor de agentes de los que realmente existen en el cluster, ya sea por una mala configuración o porque hay mucha diferencia entre los valores límites de una hormiga y su media. En estos casos sucede que ante una colisión de dos hormigas se pueda generar tres números de identificación, sin embargo este tercer identificador que representa un falso positivo tendrá una vida corta en cantidad de frames, de la misma manera que ante un blackout.

Esta información se puede analizar con posterioridad a la obtención de las trayectorias, por ejemplo graficando la longitud de información de las trayectorias para cada uno de los agentes detectados, como se muestra en la figura a continuación.

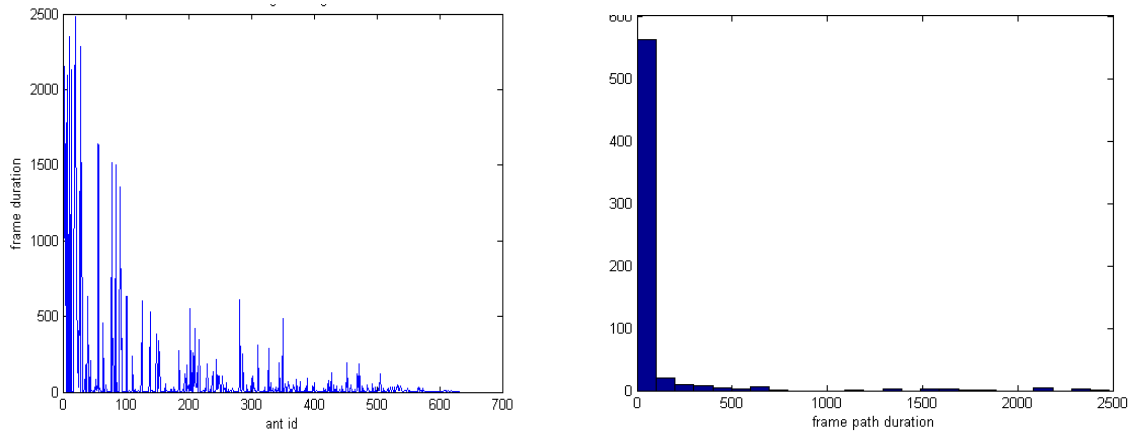


Figura 12-Largo de caminos obtenidos. A la izquierda longitud de datos en función de ID del agente y a la derecha un histograma de las longitudes

De esta manera todas las trayectorias que superen un umbral que considere el usuario investigador tendrán o bien mayor relevancia o bien serán las únicas tenidas en cuenta. Esto en cuanto a información que brinde identidad de agentes, ya que si bien estos datos pueden estar en trayectorias que no corresponden a distintas trayectorias breves, no dejan de ser datos de trayectorias válidas, porque corresponden a centros de masa de agentes válidos.

Si bien una trayectoria puede ser más extensa que la otra, esto no significa necesariamente que el agente se haya desplazado, probablemente el agente haya estado casi estático brindando la información de un mismo punto a lo largo de la mayoría de los frames.

8.5 Velocidad

A partir de la información obtenida en las trayectorias de los agentes resulta posible obtener la velocidad de los mismos a través del cálculo de su derivada discreta, frame a frame. Un resultado que ejemplifica esto es la Figura 13, en la cual se observa un histograma de todas las velocidades a lo largo de un video.

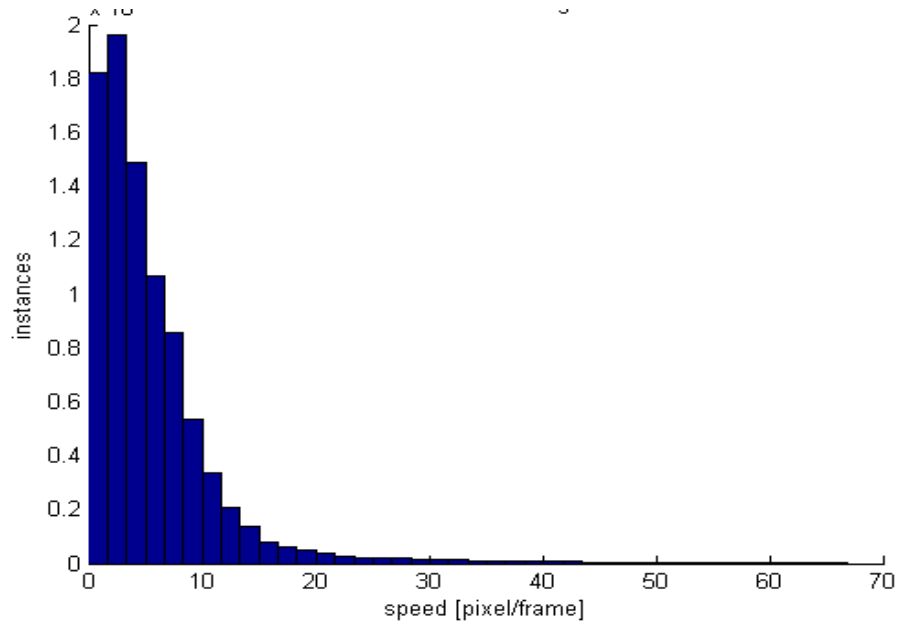


Figura 13 - Histogramas de la velocidades de todas las hormigas en el video P2CONCENTADO52_2.avi

8.6 Espacio transitado

Al representar la totalidad de las trayectorias en una imagen resulta evidente que hay zonas que no han sido recorridas por los agentes así como otras que son preferidas o sobre transitadas.

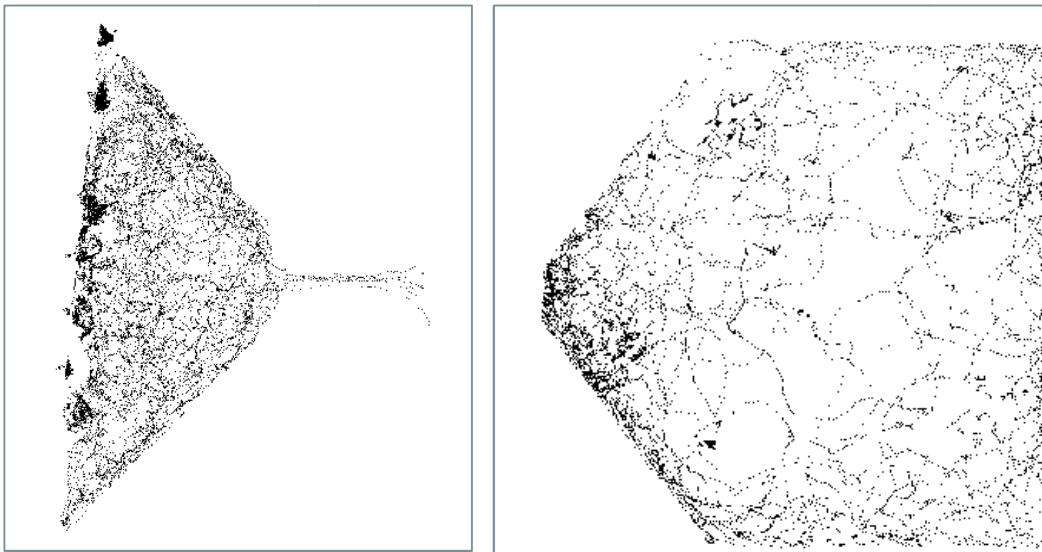


Figura 14-Espacio transitado y no transitado por las hormigas en videos con especies diferentes y en escenarios distintos, a la izquierda hormigas negras grandes y a la derecha especies de menor tamaño

8.7 Indicadores de la interfaz

Del mismo modo en la interfaz visual se puede supervisar distintos indicadores respecto del tamaño de los agentes en estudio así como supervisar el proceso de seguimiento ya sea de una forma individual o grupal de los mismos.

8.7.1 Tamaño de los agentes

A partir del estudio del tamaño de los agentes se puede observar que en algunos casos como por ejemplo en expuesto en la Figura 15, los mismos tienen un tamaño con distribución normal.

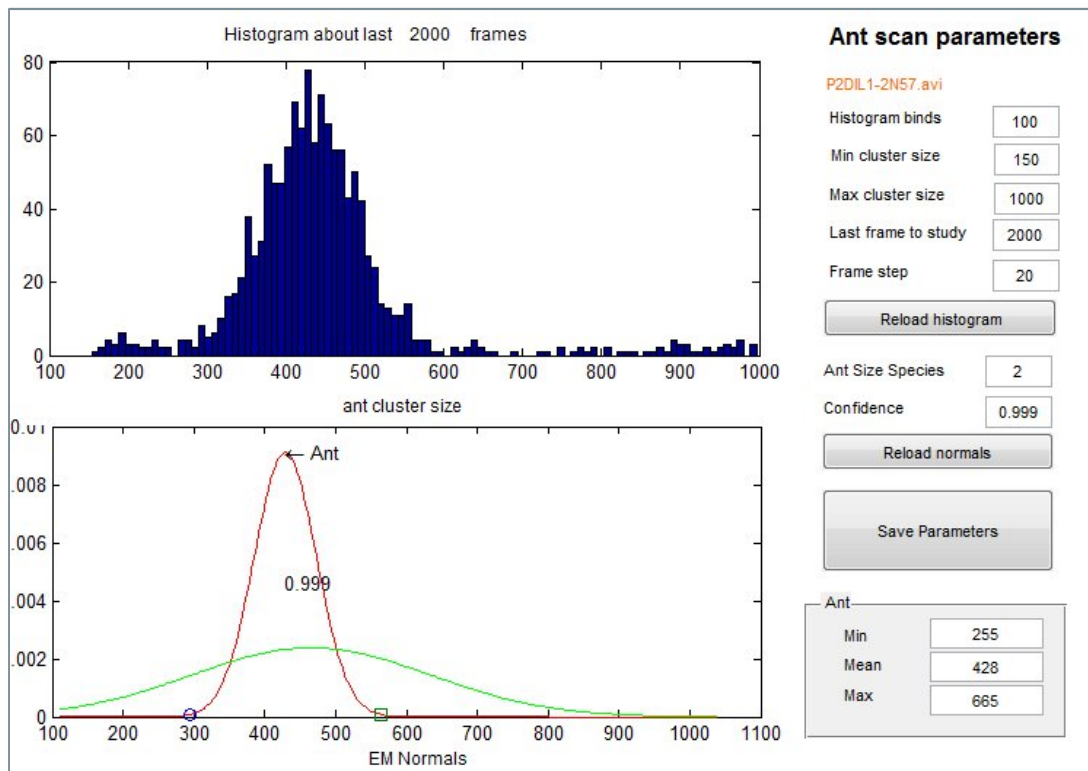


Figura 15 - Análisis de la distribución del tamaño de hormiga en el video P2DIL1-2N57.avi

8.7.2 Trayectoria individualizada

Las trayectorias individualizadas son un indicador que ayuda a verificar que el proceso de seguimiento se ejecute de un modo correcto, es decir que el agente dibuje con su desplazamiento una línea que se corresponda con el mismo.

Por otro lado ayuda a acotar el estudio de agentes de una forma individualizada. Tal vez por alguna razón sólo interese sacar métricas parciales de los recorridos al discriminar agentes de interés.

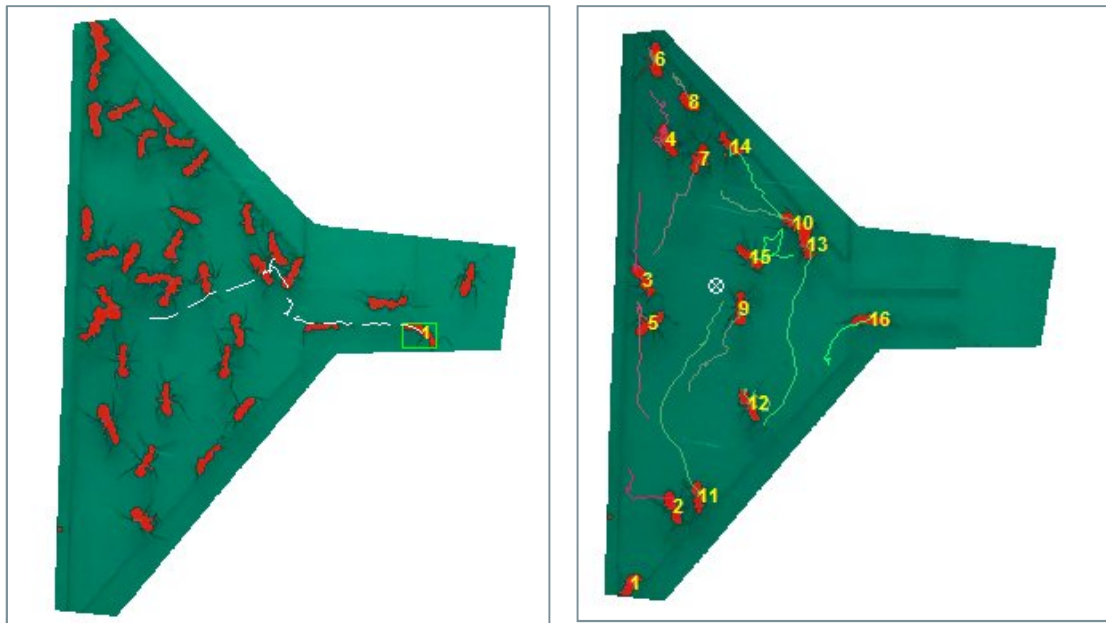


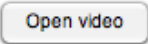
Figura 16 - Seguimiento de una hormiga en particular o muchas con distintos colores y numeración cómo identificación

Por otro lado para facilitar la lectura de las trayectorias de los agentes, se seleccionan colores aleatorios para cada uno de los agentes.

9. Guía del Usuario

9.1 Carga de parámetros para poder seleccionar hormigas a seguir.

9.1.1 Abrir el video

Para abrir el video, se debe presionar el botón  y luego buscar la carpeta correspondiente en donde tendremos que seleccionar el video a analizar.

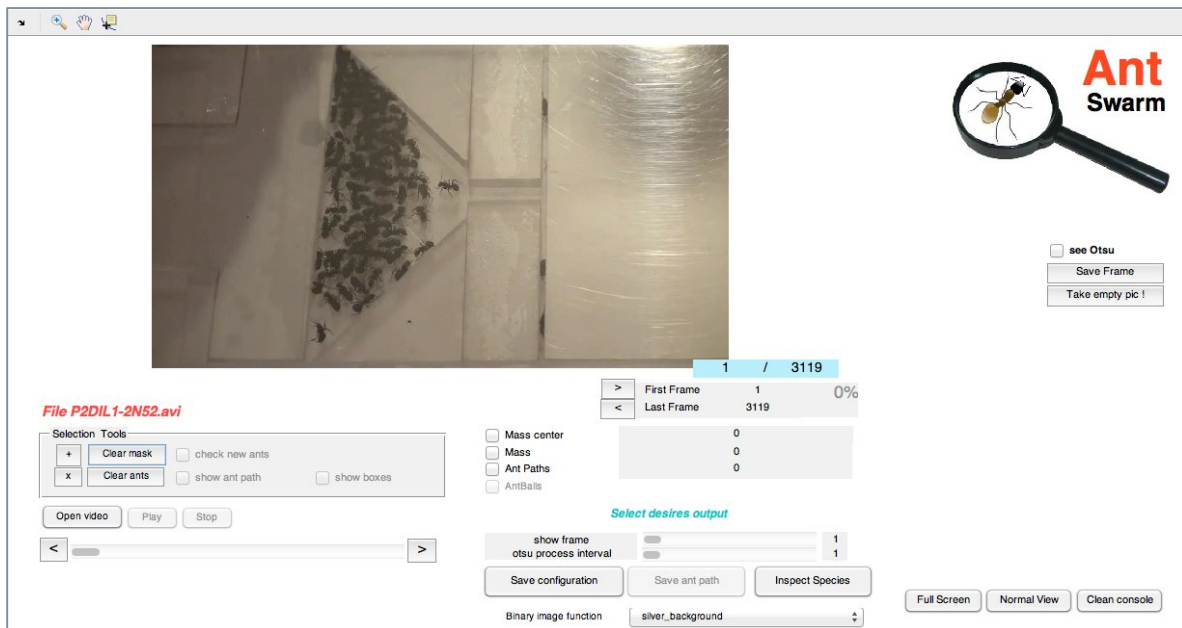



Figura 17 – Abrir el video

9.1.2 – Elegir frame de comienzo

Movernos hasta el frame del video en el cual se quiera empezar a analizar el mismo.

9.1.3 – Seleccionar máscara

Presionar el botón  para seleccionar la región de interés. Notar que una vez apretado el botón si ponemos el mouse sobre la imagen del video, el cursor es un signo mas(+). Cada vez que se presiona el botón izquierdo del mouse, aparecerá un punto azul.

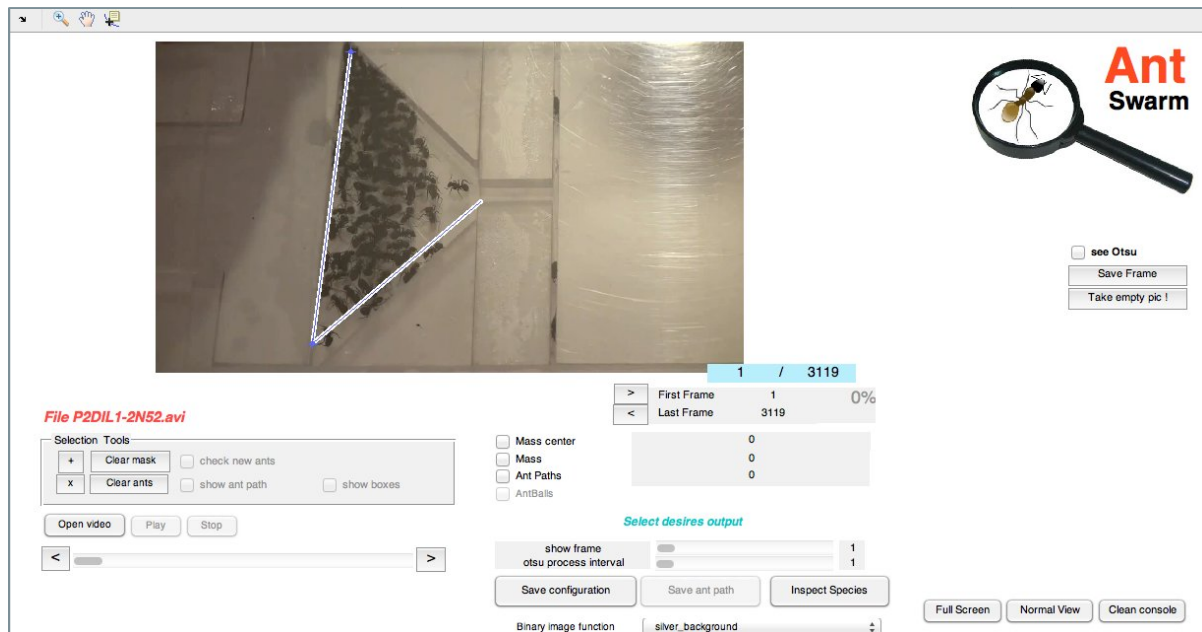


Figura 18 – Proceso de selección de una zona en el video

La región se cerrará al realizar un click sobre un punto azul. Una vez cerrada la región, al hacer doble click izquierdo sobre la región seleccionada se recortará la imagen para un mejor análisis.

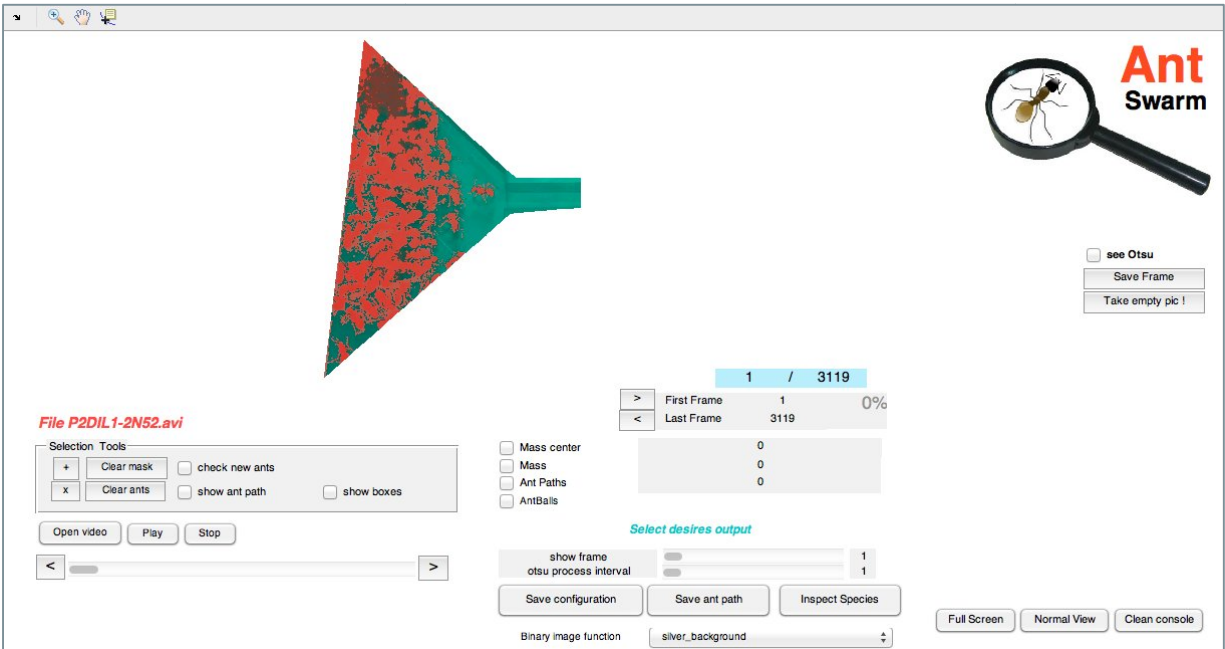
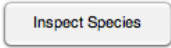


Figura 19 – Zona ya seleccionada

Una vez seleccionada el área e intervalo de frames de interés para el análisis del video, guardar la configuración con el botón llamado Save Configuración.

9.1.4 – Analizar tamaño de hormigas

El próximo paso es analizar el tamaño de las hormigas en el video en cuestión. Para esto presionaremos el botón  y tendremos que analizar los frames del video en la siguiente pantalla.

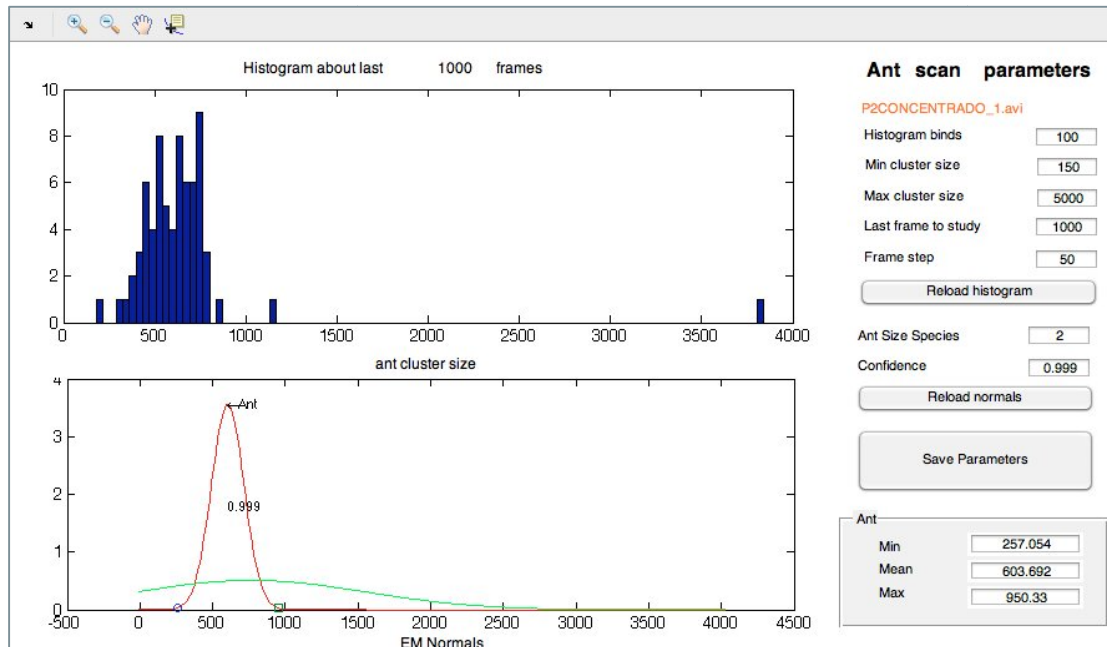
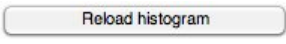
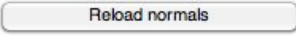



Figura 20 – Análisis del tamaño de las hormigas

9.1.5 – Ingresar parámetros de evaluación de tamaño de hormiga

Para poder estimar el tamaño promedio real de la hormiga en el video, debemos primero ingresar la cantidad de intervalos (Histogrambins), luego el tamaño mínimo y máximo del cluster, la cantidad de frames a analizar y el paso en que se van tomando los frames. Luego tendremos que presionar el botón .

9.1.6 – Tamaño mínimo, máximo y media.

El próximo paso es seleccionar cuantas normales se quiere sacar en el histograma y con que nivel de confianza obtener el intervalo de la normal más alta. Presionar el botón . Por último, guardar estos parámetros obtenidos presionando el botón .

9.1.7 – Guarda parámetros

Una vez encontrado los parámetros para el video, se deberá presionar el botón




y cerrar esta ventana.

9.2 Tracking de hormigas

Para realizar el tracking de hormigas es indispensable haber realizado la carga de parámetros anteriormente. Una vez finalizada la carga, tendremos que seguir los siguientes pasos para lograr el tracking de una o mas hormigas.

9.2.1 – Seleccionar hormigas

Presionar el botón  y elegir la/s hormiga/s a seguir. El seleccionado será correcto si podemos ver en la imagen una elipse alrededor de la hormiga en cuestión.

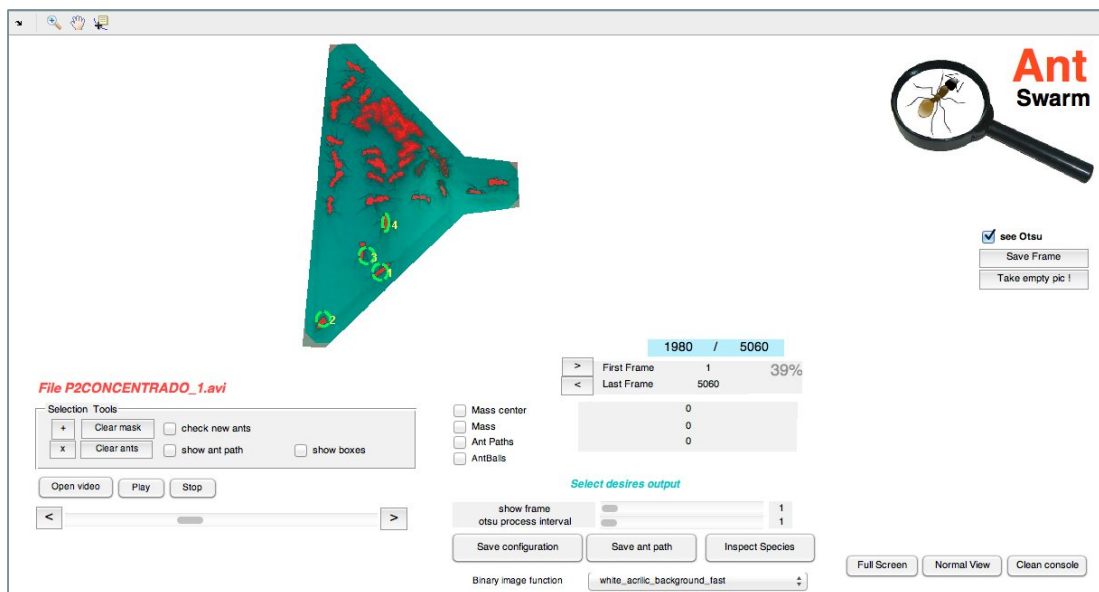


Figura 21 – Proceso de selección de agentes sobre el video

Si se desea seguir a todos los agentes tildar la opción Check New Ants.

9.2.2 – Ver camino de hormigas

Si se quiere ver el camino de la hormiga en la imagen se debe marcar el recuadro show ant path y una vez que se ejecuta el video, se podrá observar el camino que hace la hormiga.

9.3 Reproducción

9.3.1 – Configurar parámetros de comienzo y fin

Para poder reproducir y analizar los videos, luego de haber configurado el tamaño de la hormiga y la región a analizar, debemos configurar desde que frame y hasta que frame del video haremos los análisis. Para esto debemos ir al frame que deseamos que sea el primer frame del análisis y presionar el botón . Lo mismo haremos con para seleccionar el último frame de análisis y presionaremos el botón .

9.3.2 – Configurar salidas deseadas

Para determinar que salida se desea obtener, se deben tildar las checkbox que determinan las distintas salidas. Estas son: Centro de masa, masa total, recorridos de las hormigas e información sobre los clusters mayores a una hormiga.

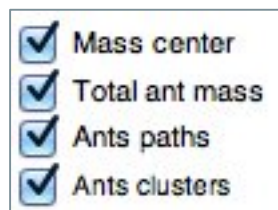


Figura 22 – Opciones sobre las salidas deseadas por el usuario

9.3.3 – Controles de video

Para poder avanzar o retroceder los videos y poder empezar a verlo y frenarlo se cuenta con diferentes botones para lograr esto.

Se cuenta con un slider para poder avanzar y retroceder más rápido. Si se desea más precisión se puede utilizar los botones (<) y (>).

Para reproducir el video se debe apretar el botón (Play) y para frenar la reproducción el botón (Stop).



Figura 23 – Controles de reproducción de video

9.4 Funciones de umbralizado

Para lograr mejores resultados se debe tener en cuenta que cada video podría necesitar una función de umbralizado particular. Esto varía según la calidad del video, agentes externos que afectan al video, luces, sombras entre otras cosas.

Se cuenta con las funciones necesarias para analizar los videos entregados hasta la fecha de elaboración de este informe. Para seleccionar que función utilizar solo hay que seleccionarla del combo.



Figura 24 – Combo de selección para las funciones de umbralización

9.5 Detalles de UI

Otros detalles de la interfaz permiten utilizar la aplicación de una manera más óptima. Estas mejoras se detallan a continuación.

9.5.1 – Porcentaje de análisis completado



Figura 25 – Porcentaje de reproducción en función del intervalo de interés definido por el usuario

9.5.2 – Cantidad de hormigas en el frame actual

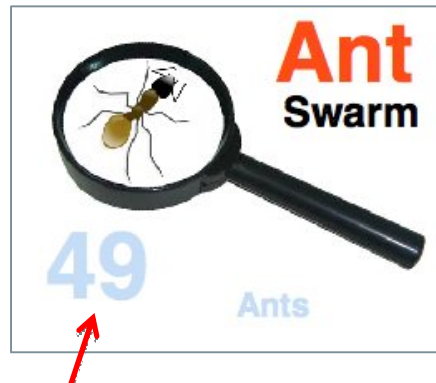


Figura 26 – Contador de agentes

9.5.3 – Controles para hacer zoom in y zoom out de la región seleccionada



Figura 27 – Botones para el control de modo de visualización

9.5.4 – Configurar el nombre de la especie analizada

Abajo del contador de hormigas se encuentra un combo para guardar el nombre de la especie en estudio.

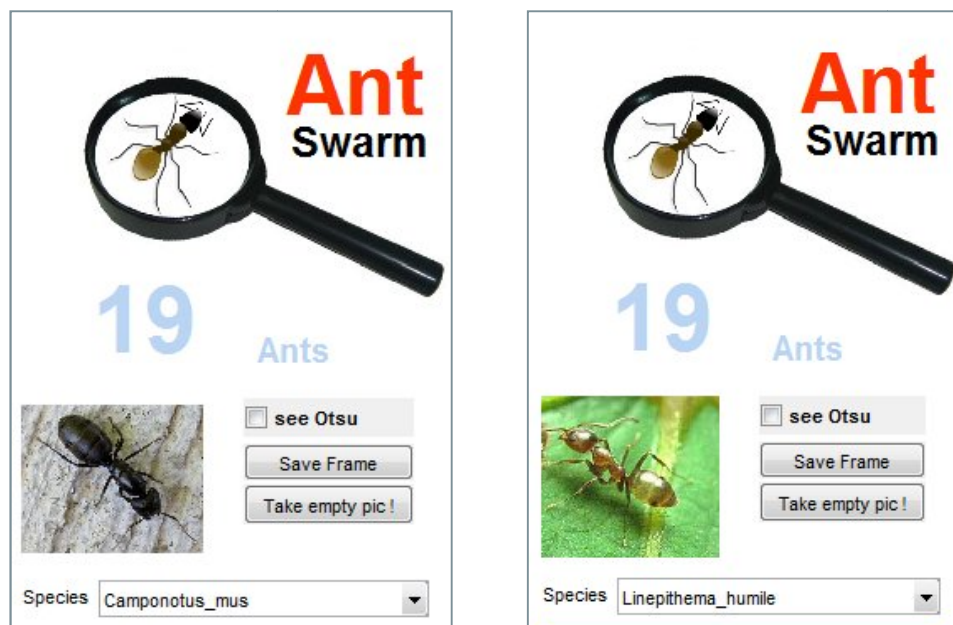


Figura 28 – Selección del tipo de especie de estudio acompañado por una imagen representativa de la misma

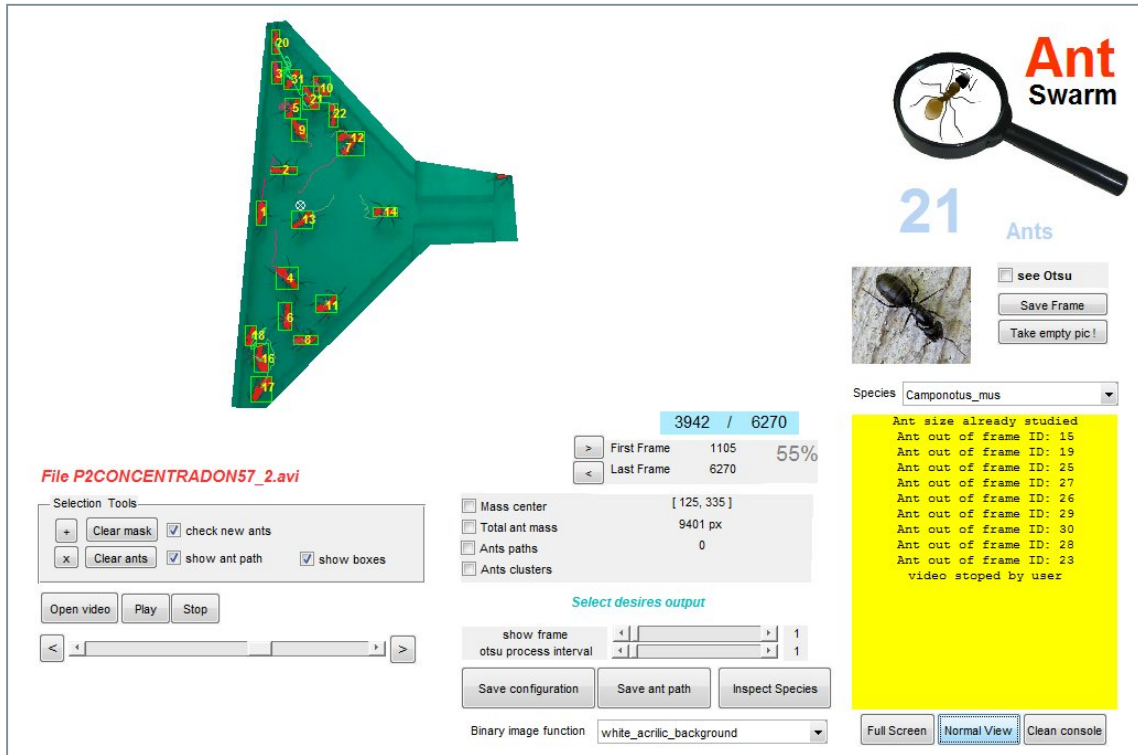


Figura 29 – Interfaz modo normal view

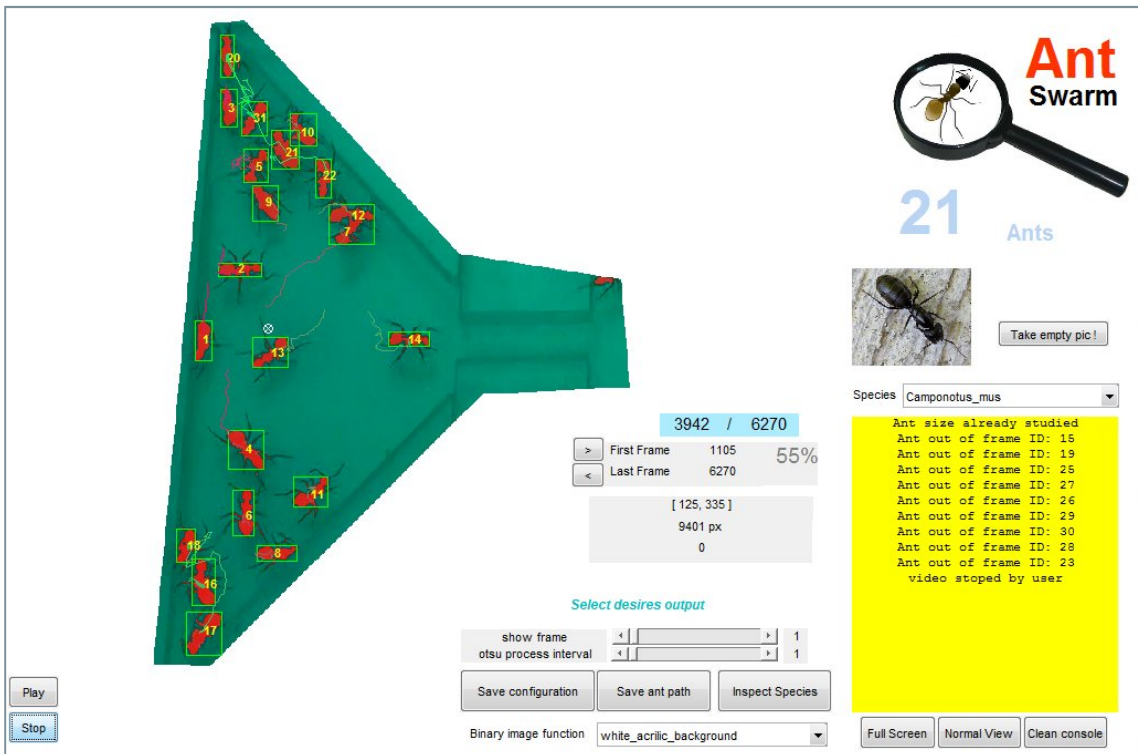


Figura 30 – Interfaz modo full screen

10. Posibles extensiones y mejoras

- Mejoras en la performance en el algoritmo de tracking para que realice un procesamiento matricial.
- Mejoras de diseño de la aplicación. Disminuir el uso de variables globales en el interfaz visual.
- Alertas durante la ejecución del video respecto de la interpretación de algún indicador. Por ejemplo que una hormiga superó una cierta velocidad.
- Probarlo con otras especies de agentes.
- Aspectos a tener en cuenta para la realización de futuros videos
 - Iluminación constante y homogénea
 - Procurar una fuente de luz perpendicular al plano del experimento, de modo de evitar la aparición de sombras de agentes.
 - No proyectar sombras arbitrarias durante la grabación de los videos
 - Evitar la aparición de objetos extraños
 - Regular previamente la nitidez del foco antes de comenzar la grabación
 - Colorar las hormigas para mejorar su identificación
 - Una vez configurado el dispositivo que realiza las grabaciones sistematizar la toma del escenario del experimento sin agentes, es decir vacío, para utilizarlo en el proceso de umbralización por sustracción del fondo vacío.

11. Conclusiones

El proyecto ha sido un desafío desde varios puntos de vista. Tanto por la complejidad técnica del problema como por la construcción de una aplicación en un entorno desconocido. Luego, el trasfondo teórico para hacer uso correcto de las herramientas mencionadas, las decisiones de diseño y las estratégicas definidas para conseguir los requerimientos del proyecto.

Durante el desarrollo del mismo, en todas las etapas se manejó un balance delicado entre obtener datos con el mayor nivel de precisión posible y la discretización o transformación de la información a partir de un video con entornos dinámicos, complejos de analizar. Y durante este último proceso estimar valores de la realidad de modo de poder darle un proceso computacional.

A pesar de no entrar dentro del ámbito del proyecto el hecho de analizar la información obtenida a través de la aplicación. Se han podido establecer hipótesis, definir indicadores respecto de los datos obtenidos de los agentes así como de la calidad de los mismos. Además de primeras conclusiones sobre los datos obtenidos.

Entre ellas se encuentran.:

- El tamaño de las hormigas en los videos tiene una distribución normal, en base a una supervisión visual de los histogramas obtenidos. (Figura 15)
- Hay zonas en los videos que nunca son transitadas por las hormigas. O bien son preferidas por las mismas para desplazarse en el espacio. Los gráficos sobre las trayectorias son evidencia de esta observación. Como por ejemplo la Figura 14.
- Se han podido extraer con una muy buena precisión las trayectorias individualizadas de las hormigas que era uno de los objetivos más ambiciosos del proyecto. Como ilustran las Figuras 11 y 16.
- Se puede con las trayectorias analizar las velocidades de las hormigas. Que a través de un análisis visual de los histogramas de agentes en situación de evacuación, se observa que podrían tener una distribución exponencial decreciente. Figura 13
- Se puede estudiar el desplazamiento de agentes en particular, otro de los objetivos ambiciosos, así como supervisar en tiempo de ejecución las trayectorias obtenidas hasta el momento. Como puede observarse en la Figura 16.

- Si bien la aplicación y el objetivo del proyecto tuvo como objeto de estudio videos de hormigas, la misma puede ser utilizada para cualquier otro tipo de especie (agente), dado las etapas del proceso de reconocimiento y de seguimiento.

Resultó de gran satisfacción poder hacer uso de herramientas vistas en materias optativas de la carrera como análisis y tratamiento de imágenes, así como aprendizaje automático en un problema científico del ámbito académico. La primera dictada por la Dra. Juliana Gambini y la segunda por la Dra. Julia Villar. Significó una gran ayuda el conocimiento adquirido en las mismas, así como las herramientas adquiridas a lo largo de la formación ingenieril, y también la metodología de trabajo provista por la cátedra para encarar la resolución de esta problemática complicada a cargo del Dr. Daniel Parisi.

Con todo, a pesar de la complejidad del proyecto, resultó divertido darle una solución.

12. Referencias

12.1 Papers

[1] S. A. Soria, R. Josens y D. R. Parisi. "Experimental Evidence of the 'FasterisSlower' Effect in theEvacuation of Ants".Safety Science, Vol. 50, pp. 1584–1588, (2012).

[2] Los videos nuevos fueron realizados en el marco de las materias Laboratorio 6 y 7 de la carrera Lic. en Cs. Físicas de la FCEN-UBA por el alumno Santiago Boari, bajo la dirección de Daniel R. Parisi y Roxana Josens en el "Grupo de Estudio de Insectos Sociales, Dto. de Biodiversidad y Biología Experimental, FCEN-UBA ".

12.2 Funciones utilizadas

http://homepages.cwi.nl/~pauwels/matlab_library/progs/clustering/cluster_1_dim/em_1_dim.xml

<http://www.eecs.ucf.edu/~dcm/Teaching/EngeneeringAnalysis/ClassNotes/NASA.pdf>

12.3 Algoritmos

12.3.1 Adyacencia

<http://www.mathworks.com/help/toolbox/images/ref/bwlabel.html>

<http://www.mathworks.com/help/toolbox/images/ref/bwconncomp.html>

<http://www.mathworks.com/help/toolbox/images/ref/regionprops.html>

http://www.codecadex.com/wiki/Implementing_the_flood_fill_algorithm#Pseudocode

12.3.2 Em

<http://www.aquaphoenix.com/lecture/matlab10/page2.html>

<http://www.mathworks.com/matlabcentral/fileexchange/10956>

<http://www.mathworks.com/matlabcentral/fileexchange/26184-em-algorithm-for-gaussian-mixture-model>

<http://www.mathworks.com/matlabcentral/fileexchange/20712-em-for-hmm-multivariate-gaussian-processes>

<http://www.mathworks.com/matlabcentral/fileexchange/24867-gaussianmixturemodel-m>

12.33 Fcm

<http://www.mathworks.com/help/toolbox/fuzzy/fcm.html>

<http://www.mathworks.com/help/toolbox/fuzzy/fp310.html>

Agradecimientos

Daniel:

Les agradezco a todas las personas que me ayudaron a lo largo de toda la carrera, en especial a mis padres y a mi hermana que me soportaron en los momentos más difíciles de la carrera.

A Juliana Gambini por brindarnos su conocimiento en temas de procesamiento de imágenes y a Daniel Parisi por confiar en las ideas de dos estudiantes y por su cálido trato.

José Indalecio:

Al Instituto Tecnológico de Buenos Aires por haberme becado durante mi estadía. Por todo lo que he podido aprender de las personas que lo conforman directivos, profesores, ayudantes, amigos y compañeros.

En memoria de mi querida abuela Ofelia que juntaba las moneditas para comprarme galletitas en el almacén y que se esforzó mucho en que su descendencia tuviera las posibilidades que ella no tuvo.

A mi madre, quien sin ella poco hubiera sido de mi.

Y a todos los que me conocen ☺. Y a Valen que lo quiero mucho.