

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

**ALGORITMO QUE PERMITE DETERMINAR LA TENDENCIA DE PERSONAS EN UN ENTORNO  
INDUSTRIAL ROBOTIZADO**

CAROLINA LÓPEZ GALARZO

SEBASTIÁN RESTREPO LARGO

INGENIERÍA DE SISTEMAS  
TRABAJO DE GRADO EN LABORATORIO

Director(es) del trabajo de grado

MAURICIO ARIAS CORREA

**INSTITUTO TECNOLÓGICO METROPOLITANO**

**2017**

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## RESUMEN

---

En el siguiente trabajo, se desarrolló un algoritmo usando el IDE de software matemático MATLAB y sus funciones del paquete de visión. Para la detección y seguimiento de las personas en un ambiente industrial robotizado para poder determinar la tendencia de movimiento de la misma, la metodología utilizada para el desarrollo del trabajo fue el siguiente:

1. Se investigó sobre el flujo óptico, concepto general, aplicaciones y técnicas utilizadas para su obtención.
  
2. Se consultó sobre frameworks, software o herramientas computacionales, que permitieran trabajar con flujo óptico.
  - 2.1 Se realizaron pruebas con algunas de las herramientas seleccionadas para determinar cuál es la más adecuada para trabajar.
  
3. Finalmente, como resultado, se realizó la construcción de un software que implementa el algoritmo desarrollado.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## RECONOCIMIENTOS

---

Se reconoce al Laboratorio de óptica, fotónica y visión artificial en cabeza del docente investigador Mauricio Arias Correa, que, gracias a sus conocimientos y orientación, se pudo lograr el objetivo con muy buenos resultados. Además de las enseñanzas para la vida personal y profesional.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## ACRÓNIMOS

---

MatLab : ***MAT**rix **LAB**oratory*

OpenCV : *Open Source Computer Vision*

GMM: *Modelo de Mezclas gaussianas (Gaussian mixture model)*

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## TABLA DE CONTENIDO

1. INTRODUCCIÓN .....	6
2. MARCO TEÓRICO .....	7
2.1 Que es el flujo óptico .....	7
2.2 Categorización de flujo óptico .....	7
2.3 Cálculos matemáticos para el flujo óptico.....	8
2.4 Descripción gráfica.....	10
2.5 Método de Gunnar Farneback.....	10
2.6 Detección de primer plano usando modelos de mezcla Gaussiana .....	11
2.7 Diagrama de flujo.....	12
3. METODOLOGÍA.....	13
4. RESULTADOS Y DISCUSIÓN.....	19
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO .....	26
5.1 CONCLUSIONES.....	26
5.2 RECOMENDACIONES.....	26
5.3 TRABAJO FUTURO .....	27
REFERENCIAS .....	28
APÉNDICE.....	29
APENDICE A.....	29

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# 1. INTRODUCCIÓN

---

En el área de visión por computadora, el seguimiento y la detección de personas es uno de los problemas que más interés y desarrollo generan en los investigadores. Dado esto se encuentran múltiples aplicaciones tales como sistemas de seguridad, análisis de comportamientos en personas, y hasta las cámaras fotográficas de los móviles.

Debido a esto el objetivo principal es poder desarrollar e implementar el algoritmo y como objetivo específico poder contribuir con trabajos relacionados en el área de la visión artificial.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2. MARCO TEÓRICO

---

### 2.1 Que es el flujo óptico

El concepto de flujo Óptico se relaciona con el de patrón Óptico y hace referencia a la estructura poseída por la luz en un punto de observación en movimiento. Más concretamente, mientras que hablar del patrón Óptico sería hacerlo respecto a las diferencias cuantitativas y/o cualitativas que caracterizan a la energía que converge en un punto del espacio (la luz que llega en una dirección es más o menos intensa y/o se acumula en longitudes de onda distintas a la que llega de otra), hablar del flujo óptico sería hacerlo respecto a las características del patrón Óptico si se asume un punto de observación en continuo movimiento. Si bien es cierto que existen múltiples formas de describir las características de este flujo, se basa en la descripción de las características que tendrían las imágenes recogidas por un conjunto de cámaras de video que se desplazan con el punto de observación en movimiento. Tal descripción, además, es similar en muchos aspectos a la que podría efectuarse partiendo de las características de la imagen retiniana. (Lillo, 1995)

### 2.2 Categorización de flujo óptico

Los algoritmos de flujo óptico pueden ser categorizados en:

- Métodos basados en regiones
- Métodos basados en energía
- Métodos basados en fase
- Métodos diferenciales

(Michelin, 2015)

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Los Métodos diferenciales los cuales se harán énfasis en el trabajo se dividen en algoritmos dispersos y algoritmos densos

Los algoritmos dispersos calculan el flujo óptico de una serie de puntos de interés de la imagen. Uno de los algoritmos más utilizados para el cálculo del flujo disperso es el algoritmo de Lucas y Kanade o alguna de sus variantes

Los algoritmos densos determinan el flujo óptico para todos los píxeles de la imagen. Estos algoritmos se basan en la solución de la restricción del flujo óptico junto con alguna condición extra que determina el problema. Uno de los algoritmos para la determinación del flujo óptico denso más populares es el algoritmo de Horn y Schunck (Jose Luis Hueso, 2013)

### 2.3 Cálculos matemáticos para el flujo óptico

Los métodos de flujo óptico tratan de calcular el movimiento entre dos imágenes que fueron tomadas en los momentos de tiempo  $t$  y  $t + \Delta t$ . Estos métodos utilizan derivadas parciales con respecto a las coordenadas espaciales y temporales.

Dado un punto asociado a un píxel  $(x,y)$  en la imagen tomada en el instante  $t$ , se define  $I(x,y,t)$  como la intensidad del píxel en esa imagen. El mismo punto en la imagen tomada en el instante  $t + \Delta t$ , tendrá asociado un píxel  $(x + \Delta x, y + \Delta y)$ , siendo  $\Delta x$  y  $\Delta y$  la variación de posición con respecto al píxel anterior, y la intensidad asociada a este nuevo píxel es igual a:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (2.1)$$

El problema del flujo óptico considera la siguiente restricción:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (2.2)$$



	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Desarrollando las series de Taylor sobre (2.1) se obtiene la siguiente expresión:

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x, y, t) + \partial I \partial x \Delta x + \partial I \partial y \Delta y + \partial I \partial t \Delta t \quad (2.3)$$

Y debido a la restricción (2.2) se tiene que:

$$\partial I \partial x \Delta x + \partial I \partial y \Delta y + \partial I \partial t \Delta t = 0 \quad (2.4)$$

Dado que el flujo óptico considera movimiento de objetos, se puede establecer la velocidad con que se mueve un punto como la relación existente entre la variación de su posición en ambos ejes con respecto a la variación de tiempo  $\Delta t$ . Entonces, se define

$$V_x = \Delta x / \Delta t \quad V_y = \Delta y / \Delta t \quad (2.5)$$

como la velocidad de ese punto en los ejes  $x$  e  $y$  respectivamente.

Dividiendo la expresión (2.4) entre la variación de tiempo  $\Delta t$  y aplicando (2.5) se tiene:

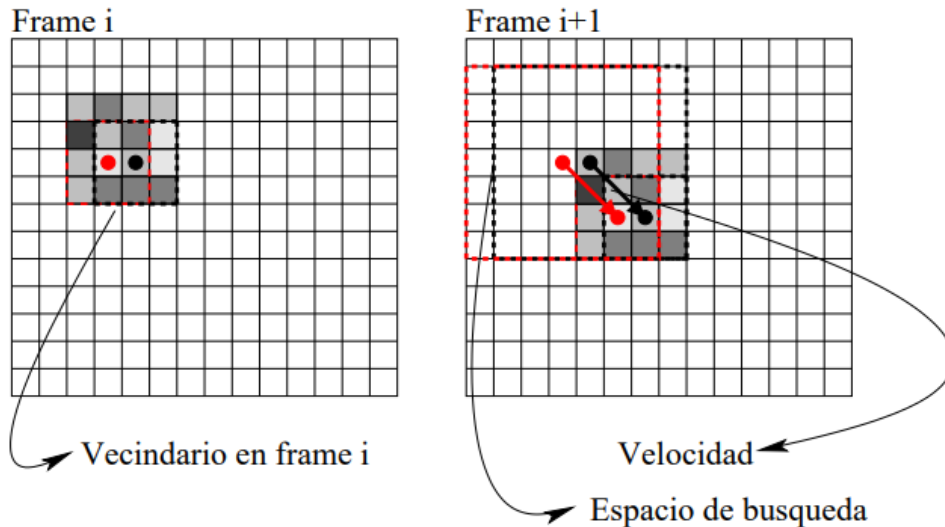
$$\partial I \partial x \Delta t \Delta t + \partial I \partial y \Delta y \Delta t + \partial I \partial t \Delta t \Delta t = \partial I \partial x V_x + \partial I \partial y V_y + \partial I \partial t = 0 \quad (2.6)$$

Renombrando las derivadas parciales como  $I_x = \partial I \partial x$ ,  $I_y = \partial I \partial y$ ,  $I_t = \partial I \partial t$ , se tiene finalmente la siguiente ecuación:

$$I_x V_x + I_y V_y = -I_t \quad (2.7)$$

Dado que esta ecuación contiene dos incógnitas,  $V_x$  y  $V_y$ , no podrá ser resuelta de forma directa. Para hallar una solución será necesario añadir alguna restricción adicional.

## 2.4 Descripción gráfica



**Figura 1 - (Adolfo Otero)**

## 2.5 Método de Gunnar Farneback

El método de Gunnar Farneback es un algoritmo de estimación de movimiento de dos cuadros. Gunnar Farneback usa polinomios cuadráticos para aproximar el movimiento entre los cuadros. Esto se puede hacer de manera eficiente utilizando la transformación de expansión polinómica.

En el caso de Gunnar Farneback, el punto de interés son los polinomios cuadráticos que producen el modelo de señal local expresado en un sistema de coordenadas local tal que

$$f(x) \sim x^T A x + b^T x + c \quad (1)$$

Donde **A** es una matriz simétrica, **b** un vector y **c** un escalar.

Gunnar Farneback es un algoritmo de flujo óptico denso porque calcula el flujo óptico para todos los píxeles en la imagen. (Jasper de Boer, Mathieu Kalksma, 2015) (Gunnar Farneback, 2003)

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2.6 Detección de primer plano usando modelos de mezcla Gaussiana

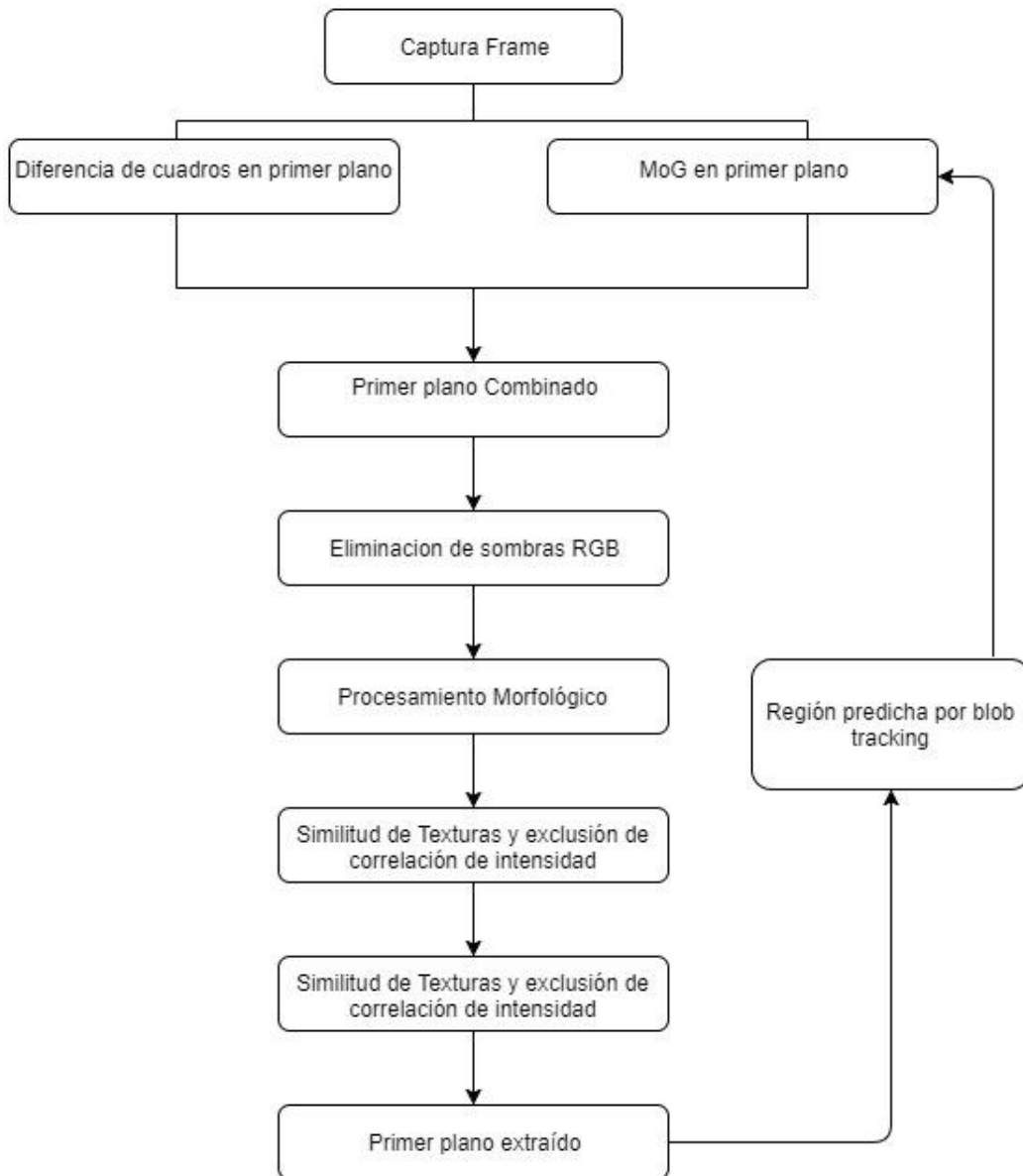
Los modelos de mezcla de Gauss son un modelo probabilístico para representar subpoblaciones distribuidas normalmente dentro de una población general. Los modelos de mezcla en general no requieren saber a qué subpoblación pertenece un punto de datos, lo que permite que el modelo aprenda automáticamente las subpoblaciones. Dado que la asignación de subpoblación no se conoce, esto constituye una forma de aprendizaje no supervisado.

Por ejemplo, al modelar los datos de altura humana, la altura se modela típicamente como una distribución normal para cada género con una media de aproximadamente 5'10 "para los hombres y 5'5" para las mujeres. Dado solo los datos de altura y no las asignaciones de género para cada punto de datos, la distribución de todas las alturas seguiría la suma de dos distribuciones normales a escala (diferente varianza) y desplazada (media diferente). Un modelo que hace esta suposición es un ejemplo de un modelo de mezcla de Gauss (GMM), aunque, en general, un GMM puede tener más de dos componentes. La estimación de los parámetros de los componentes de distribución normal individuales es un problema canónico en el modelado de datos con GMM.

Los GMM se han utilizado para la extracción de características de los datos de voz y también se han utilizado ampliamente en el seguimiento de objetos múltiples, donde el número de componentes de mezcla y sus medios predicen ubicaciones de objetos en cada cuadro en una secuencia de video.

(John McGonagle, Vincent Tembo, Alex Chumbley, s.f.)

## 2.7 Diagrama de flujo



**Figura 2** – Diagrama de Flujo (Li Yao, Miagogen Ling, 2014)

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3. METODOLOGÍA

Inicialmente se realizó un experimento en el cual se ubicó el Kinect en un ángulo elevado con visión hacia una calle, donde se pueden apreciar diferentes objetos como carros, personas, vegetación etc. Se probaron diferentes implementaciones del flujo óptico que incorpora OpenCV para validar su capacidad de procesamiento y detección. Los cuales será detallados a continuación:

1. Flujo óptico Lukas – Kanade con obtención de puntos de interés obtenidos con la técnica de Shi-Tomasi



**Figura 3**

Con este método se registran unos puntos iniciales que se obtienen utilizando la función propia de OpenCV `cv2.goodFeaturesToTrack()`. Se siguen iterativamente esos puntos usando el flujo óptico de Lucas-Kanade con la función `cv2.calcOpticalFlowPyrLK()` de OpenCV.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. Flujo óptico denso usando el algoritmo de Gunnar Farneback.



**Figura 4**

Con este método se puede determinar la dirección y velocidad de un objeto en movimiento utilizando la función de OpenCV `cv2.calcOpticalFlowFarneback()`.

Seguimiento de objetos utilizando Lukas-Kanade



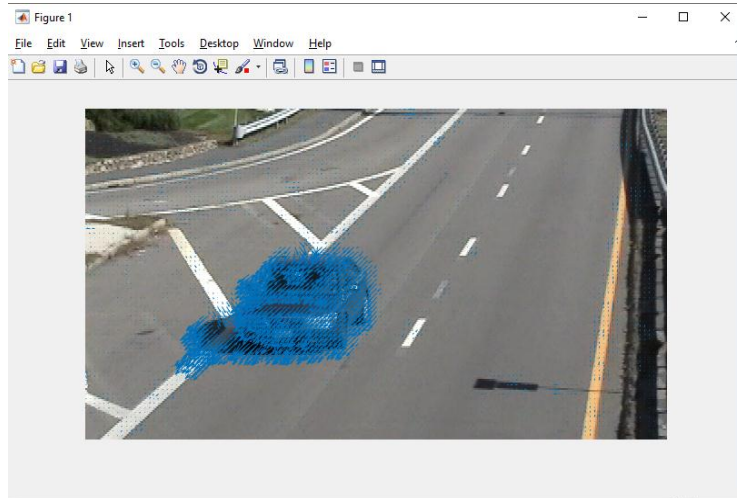
**Figura 5**

Método que de igual manera sigue iterativamente los puntos registrados con Lukas-Kanade

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

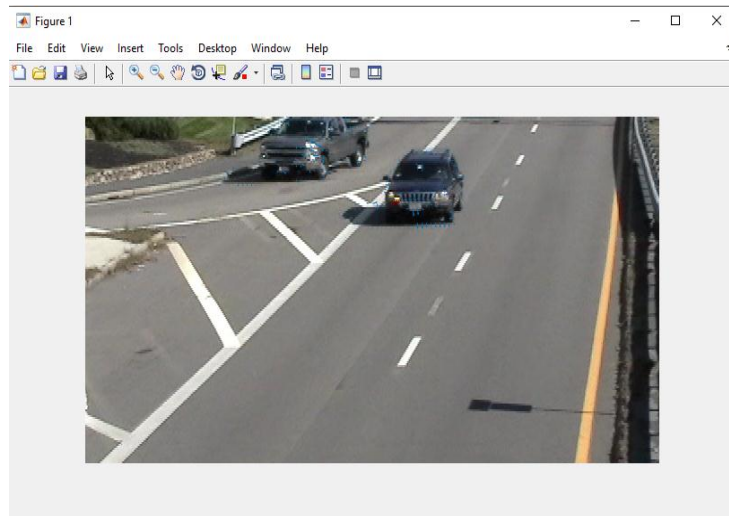
Después de haber probado todos los métodos que nos brinda las librerías de OpenCV, se procede a probar todas las funciones con las que cuenta MatLab para el tema de visión artificial y que sirva para la parte de detección, seguimiento y generar la estimación de flujo óptico.

Se empezó probando el método de flujo óptico con la técnica de Gunnar Farneback.



**Figura 6**

Posteriormente se prueba el método de flujo óptico de Lukas-Kanade arrojándonos el siguiente resultado.



**Figura 7**

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Se analizan los resultados arrojados por todos los metodos probados, y se selecciona primero en que lenguaje y con que librerias se va a trabajar. Se selecciona entonces el IDE MatLab con su lenguaje de programacion y el ToolBox de vision que lo incluye, por tener una documentacion mas estructurada, una comunidad mas amplia, la curva de aprendizaje mas rapida, todo esto con respecto a las librería de OpenCV en cualquiera de los lenguajes con los que se utilice, ya sea C++, java o python. Las 2 opciones nos brindan potencia y robustes, pero Matlab nos presenta las soluciones de una manera mas simplificada y trabajable, ya que no se requiere conocimientos previos muy amplios para poder utilizar estas funciones, cosa que no pasa con OpenCV ya que se debe tener basto conocimiento en el lenguaje con el que se vaya a programar y la documentacion esta muy fragmentada.

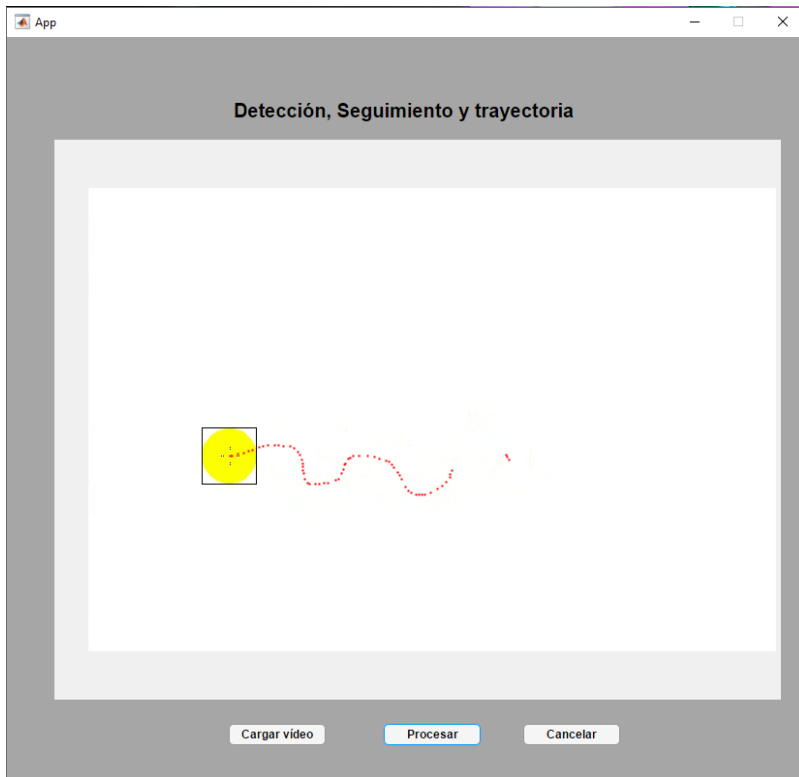
Luego de escoger la herramienta, se selecciona el metodo que se va utilizar para la estimacion de flujo optico de las 2 tecnicas en la que se enfoco la investigacion (Lukas-Kanade, Gunnar Farneback), comparando puntos como costo computacional y precisión, se decide utilizar la tecnica de Gunnar Franeback ya que como se puede observar en las figuras (2,4) este metodo calcula el flujo para todos los puntos de la imagen (flujo denso) obteniendo una mayor precisión y fidelidad del flujo calculado a comparacion del de Lukas-Kanade que lo hace para un conjunto de puntos definido (flujo disperso) y que puede llegar a fallar en la deteccion de todos los elementos que queramos seguir en una escena.

Posteriormente se hace el analisis para hacer la deteccion de la persona, y poder hacer un seguimiento que nos permita hacer el calculo del flujo optico solo de este objeto, y determinar la trayectoria del mismo. Para esto se usa un metodo basado en deteccion de primer plano que incluye el paquete de vision de MatLab llamado vision.ForegroundDetector, luego de esto se acota el objeto entre un rectangulo negro para hacerle el seguimiento fotograma a fotograma. Luego se hace el calculo del flujo optico por medio de la tecnica de Gunnar Farneback (flujo optico denso) y por ultimo se pinta la trayectoria del objeto en movimiento.

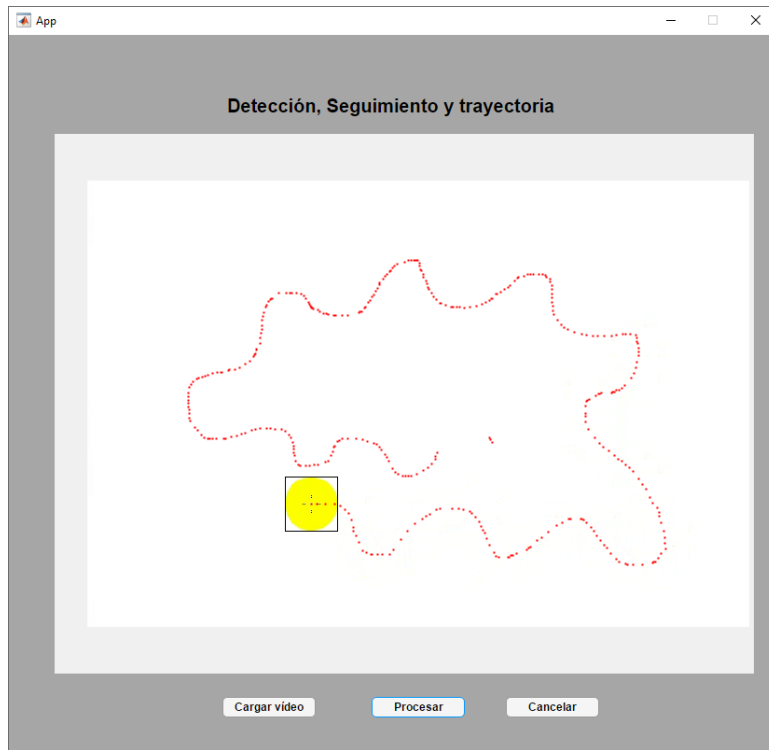


	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

A continuación se mostrara la simulacion de la escena en donde se mostrara un objeto en moviento la cual seria una persona, en la que se hace la deteccion, el seguimiento y la grafica de la trayectoria.



**Figura 8**



**Figura 9**

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 4. RESULTADOS Y DISCUSIÓN

---

### **CÓDIGO DEL ALGORITMO DESARROLLADO PARA LA DETECCIÓN, SEGUIMIENTO CON SU TRAYECTORIA Y CALCULO DEL FLUJO OPTICO**

Finalmente se obtiene el algoritmo en el que se plantean los calculos necesarios, que nos permitan determinar la tendencia de movimiento de las personas en un ambiente industrial robotizado. Esta tendencia permitira complementar un sistema futuro en el que pueden evitar colisiones en los ambientes compartidos Humano – Robot.

Codigo del algoritmo planteado en MatLab.

```

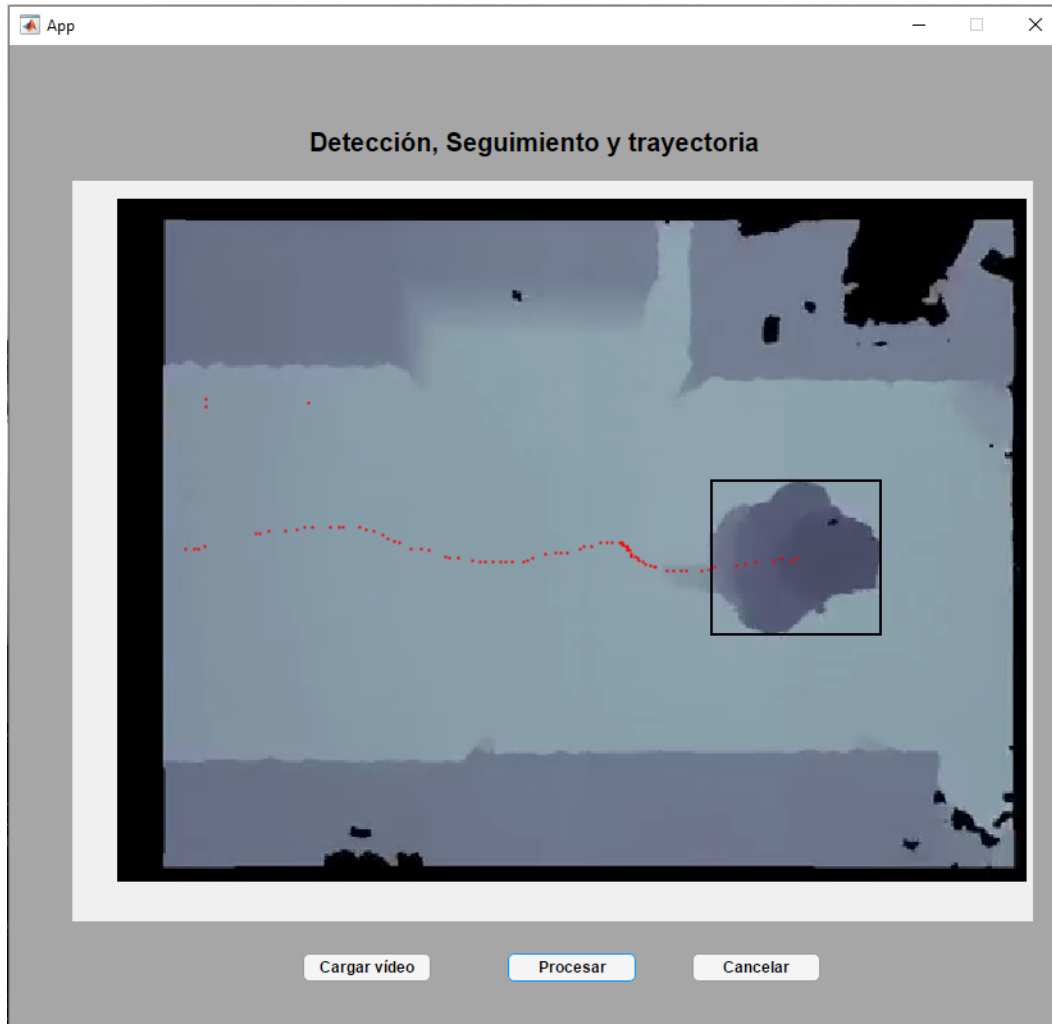
1  global fullPath;
2  % Leemos el video suministrado, donde sólo un objeto se mueve. Leemos el
3  % video en RGB. También es posible leerlo en escala de grises
4  vidReader = VideoReader(fullPath);
5  % Calculamos el número de fotogramas del video
6  n = ceil(vidReader.FrameRate*vidReader.Duration);
7  % Establecemos el número de fotogramas de entrenamiento a 250 y la
8  % desviación en píxeles como una matriz de 100 x 100
9  detector = vision.ForegroundDetector('NumTrainingFrames', 250,'InitialVariance', 100*100);
10 % Efectuamos una análisis blob donde nos interesa acotar el objeto en
11 % movimiento mediante un rectángulo. La demás propiedades se inactivan
12 blob = vision.BlobAnalysis('CentroidOutputPort', false,'AreaOutputPort', false
13     , 'BoundingBoxOutputPort', true,'MinimumBlobAreaSource', 'Property', 'MinimumBlobArea', 250);
14 % Tomamos un rectángulo de color negro para envolver el objeto que se mueve
15 shapeInserter = vision.ShapeInserter('BorderColor', 'Black');
16 % Iniciamos la detección de flujo óptico
17 opticFlow = opticalFlowFarneback;
18 % Mientras no se termine de leer todos los fotogramas se hace
19 % lo que esta en el cuerpo de definición del ciclo mientras.
20 k = 1;
21 x = zeros(n);
22 y = zeros(n);
23 while (k<=n)
24     % Leemos el fotograma actual
25     frame = readFrame(vidReader);
26     % Convertimos a escala de grises
27     frameGray = rgb2gray(frame);
28     % Determinamos el tamaño del frame
29     % s = double(size(frameGray));
30     % Leemos la máscara detectada
31     fgMask = step(detector, frame);
32     % Obtenemos el rectángulo acotador
33     bbox = step(blob, fgMask);
34     % Si tenemos un rectángulo acotador no vacío
35     if (~isempty(bbox))
36         % Estimamos el flujo óptico
37         flow = estimateFlow(opticFlow, frameGray);
38         % Calculamos la velocidad promedio en "x" del objeto en movimiento en el fotograma
39         MVx = mean(mean(flow.Vx(bbox(1,2):bbox(1,2)+bbox(1,4),bbox(1,1):bbox(1,1)+bbox(1,3))));
40         % Calculamos la velocidad promedio en "y" del objeto en movimiento en el fotograma
41         MVy = mean(mean(flow.Vy(bbox(1,2):bbox(1,2)+bbox(1,4),bbox(1,1):bbox(1,1)+bbox(1,3))));
42         % Calculamos la norma euclídea del vector velocidad promedio V = (Vx^2 + Vy^2)^0.5
43         Norm = norm([MVx,MVy], 2);
44         % Si hay movimiento, mostramos el vector velocidad promedio unitario en el plot
45         if (Norm>0)
46             %Velocidad promedio en "x" del objeto en movimiento
47             MVx = MVx/Norm;
48             %Velocidad promedio en "y" del objeto en movimiento
49             MVy = MVy/Norm;
50         end
51         s_ = {};
52         s_.Vx = flow.Vx(bbox(1,2):bbox(1,2)+bbox(1,4),bbox(1,1):bbox(1,1)+bbox(1,3));
53         s_.Vy = flow.Vy(bbox(1,2):bbox(1,2)+bbox(1,4),bbox(1,1):bbox(1,1)+bbox(1,3));
54         CM = double([(bbox(1,1)+(bbox(1,3)/2),bbox(1,2)+(bbox(1,4)/2))]');
55         % Se inserta el rectángulo de seguimiento en el fotograma
56         out = step(shapeInserter, frame, bbox);
57         % Obtenemos rows, cols y pages desde imagen
58         [rows,cols,pages] = size(out);
59         % Ancho en píxeles del fotograma
60         app.UIAxes.XLim = [1 cols];
61         % Alto en píxeles del fotograma
62         app.UIAxes.YLim = [1 rows];
63         % Mostramos el fotograma con el rectángulo de seguimiento
64         imshow(out, 'Parent', app.UIAxes);
65         % imshow('cameraman.tif', 'Parent', app.UIAxes)
66         hold(app.UIAxes, 'on');
67         x(k) = bbox(1)+bbox(3)/2;
68         y(k) = bbox(2)+bbox(4)/2;
69         % Pintamos la trayectoria o historial
70         % de movimiento en color rojo
71         plot(app.UIAxes,x(1:k),y(1:k), 'r. ');
72     end
73     k = k + 1;
74 end

```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Con este algoritmo se obtuvieron los siguientes resultados, probados con 3 videos en los cuales podemos ver el movimiento de personas, donde se les hace una detección, un seguimiento y grafica de trayectoria.

**1. Video Tomado de <http://vrai.dii.univpm.it/re-id-dataset>**



**Figura 10**

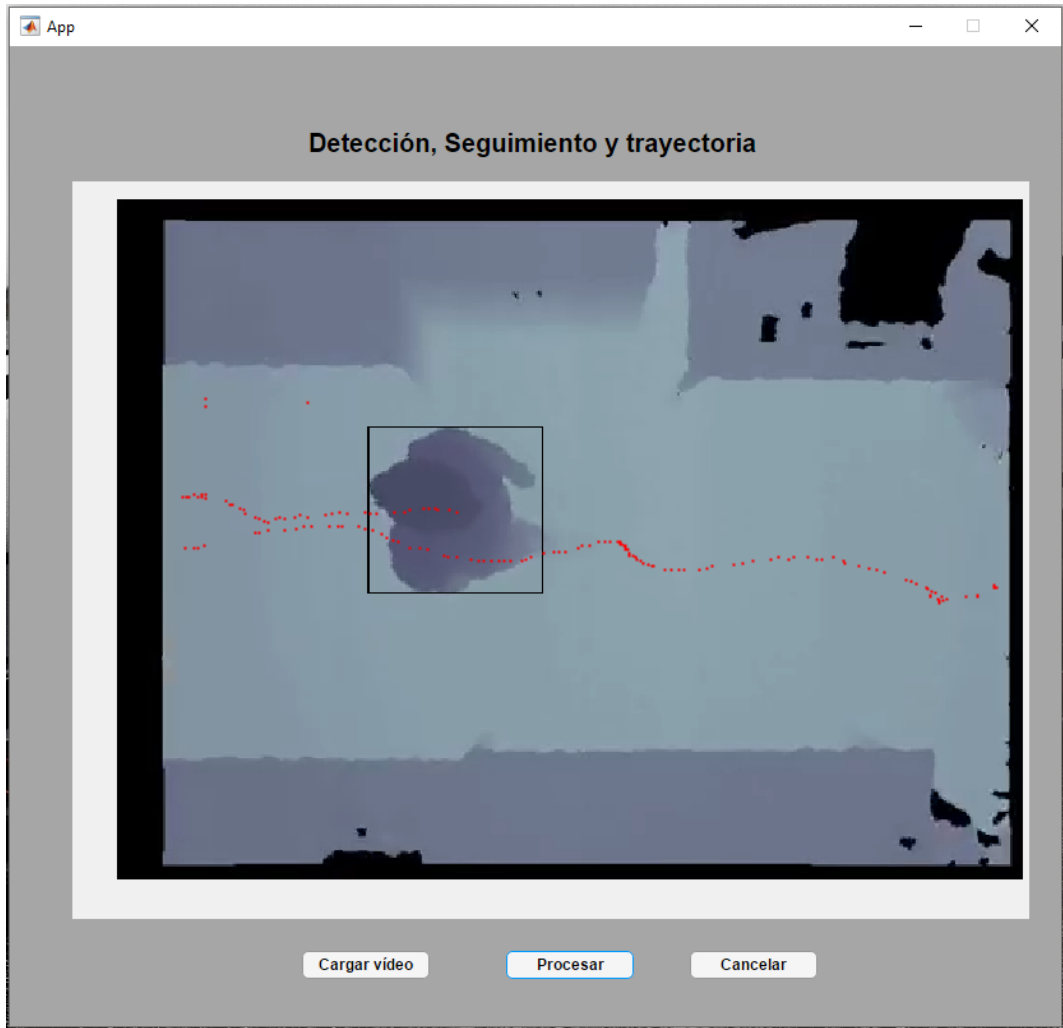
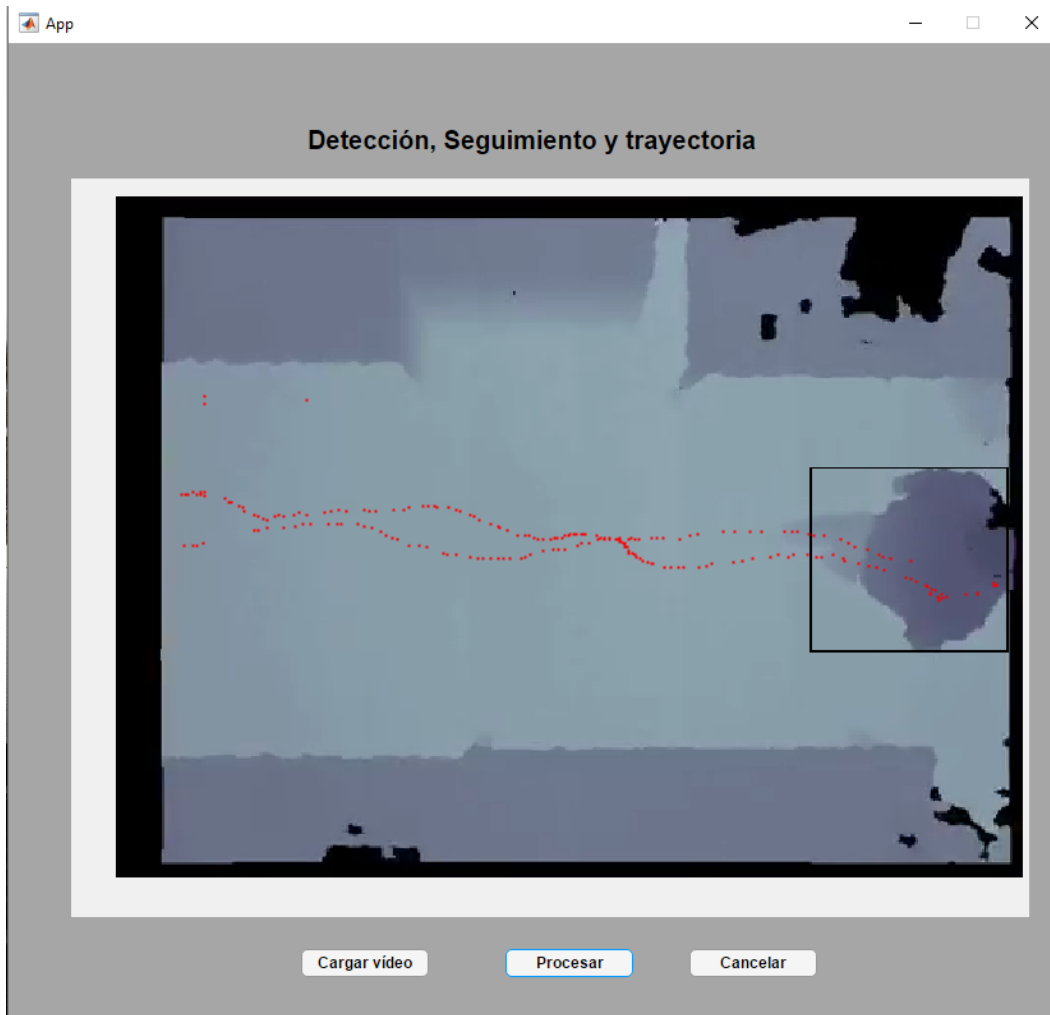


Figura 11

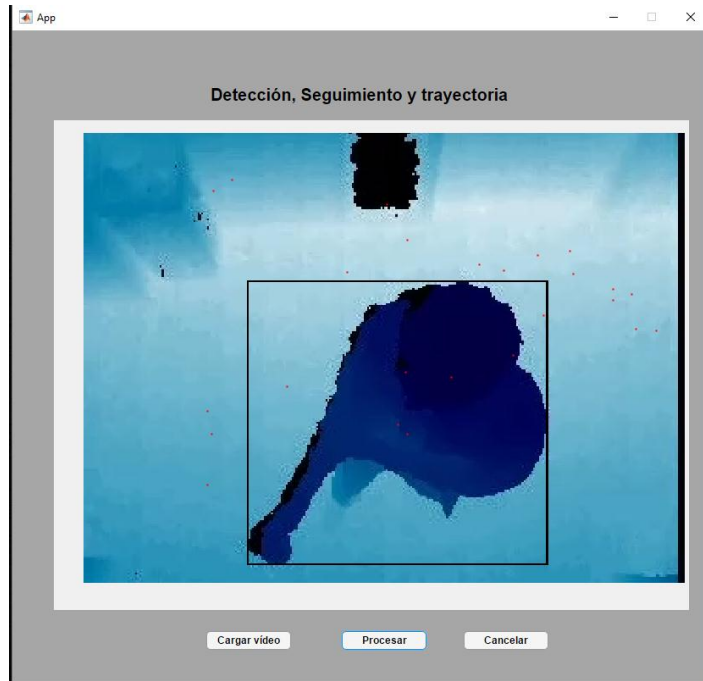


**Figura 12**

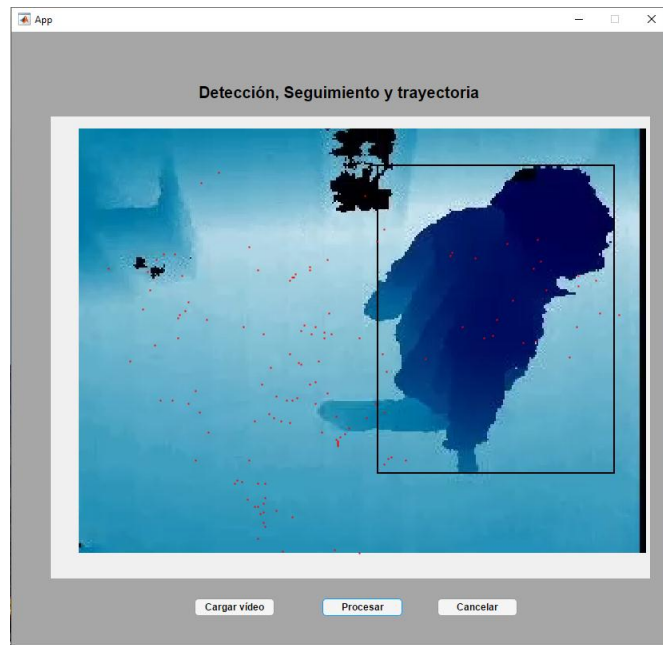
Se puede observar en el video anterior, como las personas pasan y con la camara ubicada en modo Top View se logra hacer la deteccion, y posteriormente la grafica de la trayectoria que serviran como insumos para determinar la tendencia de moviemtno de la persona en la escena.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. Video tomado de [https://www.albert.cm/projects/ram\\_person\\_id/](https://www.albert.cm/projects/ram_person_id/)



**Figura 13**



**Figura 14**



 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Se puede observar en este video la misma perspectiva, pero en un plano más cerca, que de igual forma permite detectar la persona y marcar su trayectoria, aunque como está demasiado cerca, no se logra apreciar tan clara la trayectoria ya que son recorridos muy cortos los que realiza la persona.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

---

### 5.1 CONCLUSIONES

- El algoritmo desarrollado cumple y presenta los cálculos necesarios para determinar la tendencia de movimiento de una persona en un ambiente industrial robotizado, sin embargo, se deben de considerar algunas recomendaciones para que las detecciones y los cálculos sean efectivos.
- Matlab cuenta con diferentes metodologías para desarrollar el algoritmo de tendencia de movimiento, el seleccionado fue ForegroundDetector, por su capacidad de entrenamiento y velocidad en la detección.
- La experimentación sobre la determinación de la herramienta para desarrollar el algoritmo, por medio de otros trabajos realizados, dio un amplio conocimiento sobre la habilidad de investigación, análisis y desarrollo.

### 5.2 RECOMENDACIONES

- Si el fotograma tiene un tamaño muy grande no muestra los resultados intermedios.
- Tener en cuenta que el movimiento del objeto debe ser suave, no puede tener un cambio de movimiento muy brusco.
- El objeto que se mueve debe estar totalmente contenido en el fotograma, es decir, en la escena.
- El rectángulo acotador debe estar totalmente en la escena.
- La intensidad lumínica de fotograma a fotograma debe permanecer constante.
- Como el procesamiento digital de imágenes es complejo y requiere muchos recursos de cómputo, es recomendable trabajar con un video pequeño en el sentido que el fotograma no tenga demasiados píxeles de alto ni demasiados píxeles de ancho.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Posiblemente trabajar con imagen en escalas de Grises (RGB genero un alto consumo de recursos de la maquina), o cámaras con sensores de profundidad, que nos permiten hacer una segmentación de la imagen y así y evitar ruido en la imagen, con cambios de luz bruscos que es lo que más puede afectar la detección.

### **5.3 TRABAJO FUTURO**

La aplicación a trabajos futuros de este algoritmo comprende la utilización en diferentes escenarios en los que se pueda requerir hacer un seguimiento para determinar tendencias de movimiento de persona u otros objetos. Además de apoyar los desarrollos futuros que se hagan en el Laboratorio de óptica, fotónica y visión artificial del ITM o cualquier otro desarrollo institucional en el que pueda ser de utilidad.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## REFERENCIAS

---

- Adolfo Otero, J. O. (s.f.). *Flujo óptico*.
- Arranz Aranda, F. -L.-L.-M. (2011 - 2012). *Interacción persona-computador basada en el reconocimiento visual de manos*.
- Gunnar Farneback. (2003). *Two-Frame Motion Estimation Based on Polynomial Expansion*.
- Jasper de Boer, Mathieu Kalksma. (2015). *Choosing between optical flow algorithms for UAV position change*.
- John McGonagle, Vincent Tembo, Alex Chumbley. (s.f.). <https://brilliant.org/wiki/gaussian-mixture-model/>.
- Jose Luis Hueso, J. R. (2013). *El flujo optico como herramienta para el video-analisis de fenomenos fisicos*.
- Li Yao, Miagogen Ling. (2014). *An Improved Mixture-of-Gaussians Background Model with Frame Difference and Blob Tracking in Video Stream*.
- Lillo, J. (1995). *Flujo óptico y conducción: características generales y velocidad percibida*.
- Michelin, B. M.-R. (2015). *Deteccion de movimiento mediante el uso de flujo optico*.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# APÉNDICE

---

## APENDICE A

### DEFINICION DE FUNCIONES IMPLEMENTADAS EN EL CÓDIGO DEL ALGORITMO

**vision.VideoFileReader:** Es el objeto que lee marcos de video, imágenes y muestras de audio de un archivo de video. El objeto también puede leer archivos de imagen.

**vision.ForegroundDetector:** Es el objeto que compara un cuadro de video de color o escala de grises con un modelo de fondo para determinar si los píxeles individuales son parte del fondo o el primer plano. Luego calcula una máscara de primer plano. Al usar la sustracción de fondo, puede detectar objetos en primer plano en una imagen tomada desde una cámara estacionaria.

**vision.BlobAnalysis:** Es el objeto que calcula estadísticas para regiones conectadas en una imagen binaria.

Usa la sintaxis de paso siguiente con la entrada de imagen binaria, *BW*, objeto de análisis de blobs, *H* y cualquier propiedad opcional. El método de paso calcula y devuelve estadísticas de la imagen binaria de entrada en función de los valores de propiedad especificados. El orden de los valores devueltos cuando hay varias salidas está en el orden en que se describen a continuación:

[*AREA*, *CENTROID*, *BBOX*] = paso (*H*, *BW*) devuelve el área, el centroide y el cuadro delimitador de los blobs cuando las propiedades *AreaOutputPort*, *CentroidOutputPort* y *BoundingBoxOutputPort* están establecidas en verdadero. Estas son las únicas propiedades que se establecen en verdadero de manera predeterminada. Si establece propiedades adicionales en verdadero, las salidas correspondientes siguen las salidas *AREA*, *CENTROID* y *BBOX*.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

[\_\_\_, MAJORAXIS] = paso (H, BW) calcula la MAJORAXIS de la longitud del eje principal de los blobs encontrados en la imagen binaria de entrada BW cuando la propiedad MajorAxisLengthOutputPort se establece en verdadero.

[\_\_\_, MINORAXIS] = paso (H, BW) calcula la longitud del eje menor MINORAXIS de los blobs encontrados en la imagen binaria de entrada BW cuando la propiedad MinorAxisLengthOutputPort se establece en verdadero.

[\_\_\_, ORIENTACIÓN] = paso (H, BW) calcula la ORIENTACIÓN de los blobs encontrados en la imagen binaria de entrada BW cuando la propiedad OrientationOutputPort está establecida en verdadero.

[\_\_\_, ECCENTRICITY] = paso (H, BW) calcula la ECCENTRICIDAD de los blobs encontrados en la imagen binaria de entrada BW cuando la propiedad EccentricityOutputPort se establece en verdadero.

[\_\_\_, EQDIASQ] = paso (H, BW) calcula el EQDIASQ cuadrado equivalente de los blobs encontrados en la imagen binaria de entrada BW cuando la propiedad EquivalentDiameterSquaredOutputPort se establece en verdadero.

[\_\_\_, EXTENT] = paso (H, BW) calcula la EXTENSIÓN de los blobs encontrados en la imagen binaria de entrada BW cuando la propiedad ExtentOutputPort se establece en verdadero.

[\_\_\_, PERÍMETRO] = paso (H, BW) calcula el PERÍMETRO de los blobs encontrados en la imagen binaria de entrada BW cuando la propiedad PerimeterOutputPort se establece en verdadero.

[\_\_\_, LABEL] = paso (H, BW) devuelve una matriz de etiquetas LABEL de los blobs encontrados en la entrada de la imagen binaria BW cuando la propiedad LabelMatrixOutputPort se establece en verdadero.

**vision.VideoPlayer:** El objeto VideoPlayer puede reproducir un video o mostrar secuencias de imágenes.

**opticalFlowFarneback:** Estima la dirección y la velocidad de un objeto en movimiento desde una imagen o cuadro de video a otro usando el método de Farneback.

FIRMA ESTUDIANTES: Sebastián D.  
Gerardo López G.

FIRMA ASESOR: 

FECHA ENTREGA: 16/11/2017

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD \_\_\_\_\_

RECHAZADO \_\_\_      ACEPTADO \_\_\_      ACEPTADO CON MODIFICACIONES \_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_

FIRMA CONSEJO DE FACULTAD \_\_\_\_\_

ACTA NO. \_\_\_\_\_

FECHA ENTREGA: \_\_\_\_\_