Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte de Proyecto Terminal

Estimación de pose de una mano a través de captura monocular

Coria Flores José Luis

Matricula: 202314318

_____

Trimestre 12 Primavera

Agosto de 2012

Risto Rangel-Kuoppa

Dr. en Ciencias de la Computación

Departamento de Sistemas

_____

Acerca de este manual:

En este manual se indicara el uso así como la instalación de la aplicación llamada cam1 de este proyecto.

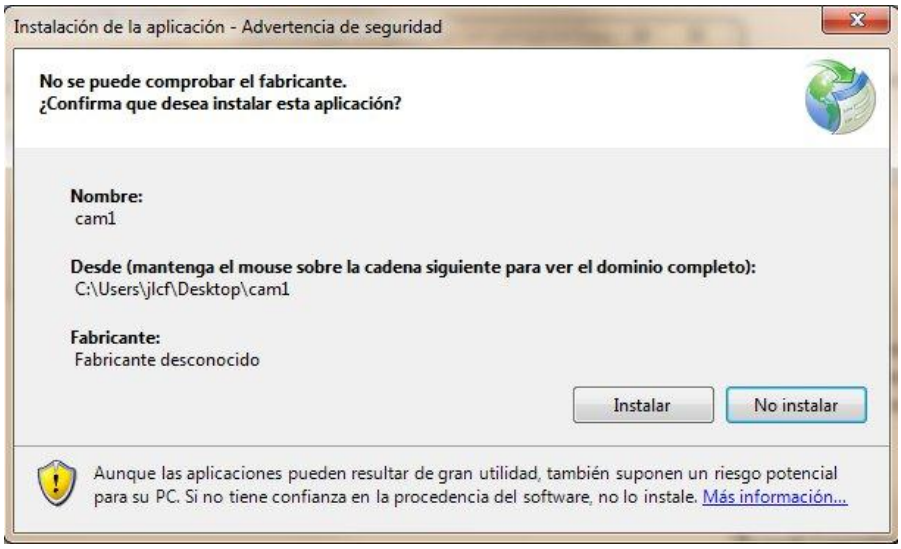
Instalación

- Es necesario para su funcionamiento tener instalado Microsoft .NET Framework 4, de lo contrario la aplicación no se ejecutara.

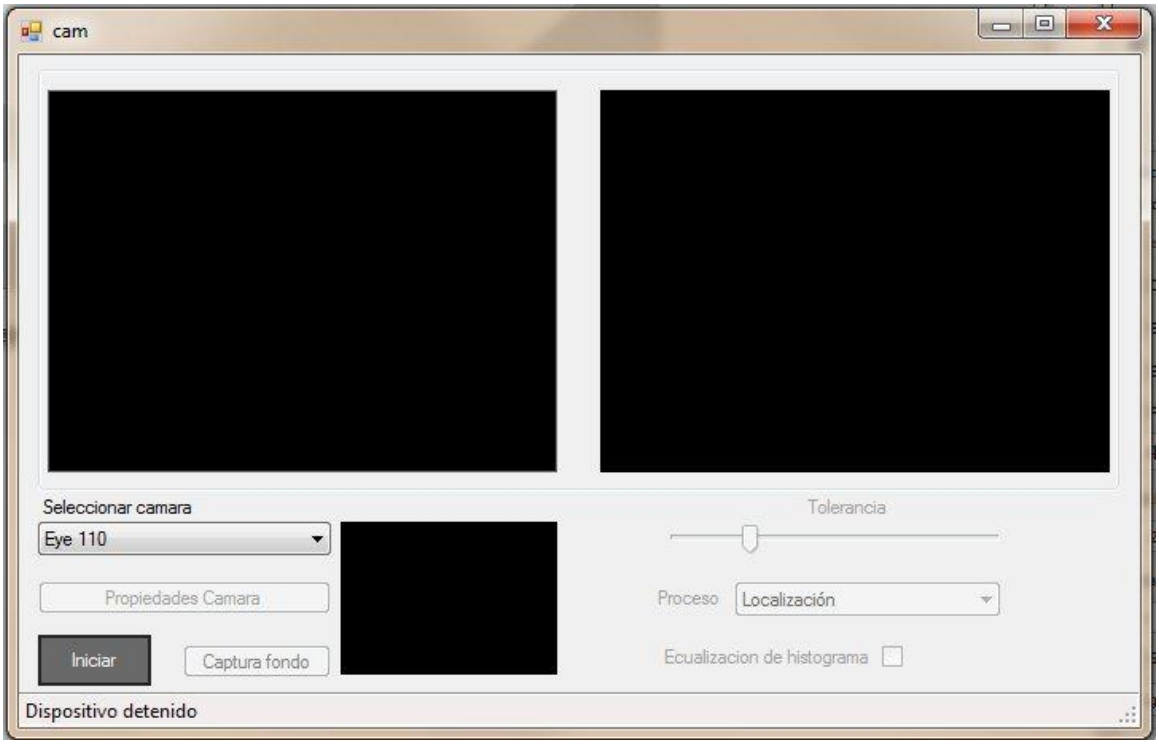
- Para descargarlo ir a: http://www.microsoft.com/en-us/download/details.aspx?id=17718 o simplemente ingresando en algún buscador: framework 4 .

- Después de descargado e instalado el framework 4, se podrá instalar la aplicación cam1, el programa de instalación consta de tres elementos: Setup.exe, cam1.application y una carpeta Application Files, ver la siguiente figura.

- Dar doble clic al icono de setup para comenzar la instalación.
- Se mostraran en pantalla la siguiente la siguiente imagen.



- Al dar clic en siguiente se instalara la aplicación al termino de esto se ejecutara la aplicación. Y se mostrara la siguiente imagen.



Con esto esta lista la instalación para empezar a usar la aplicación.

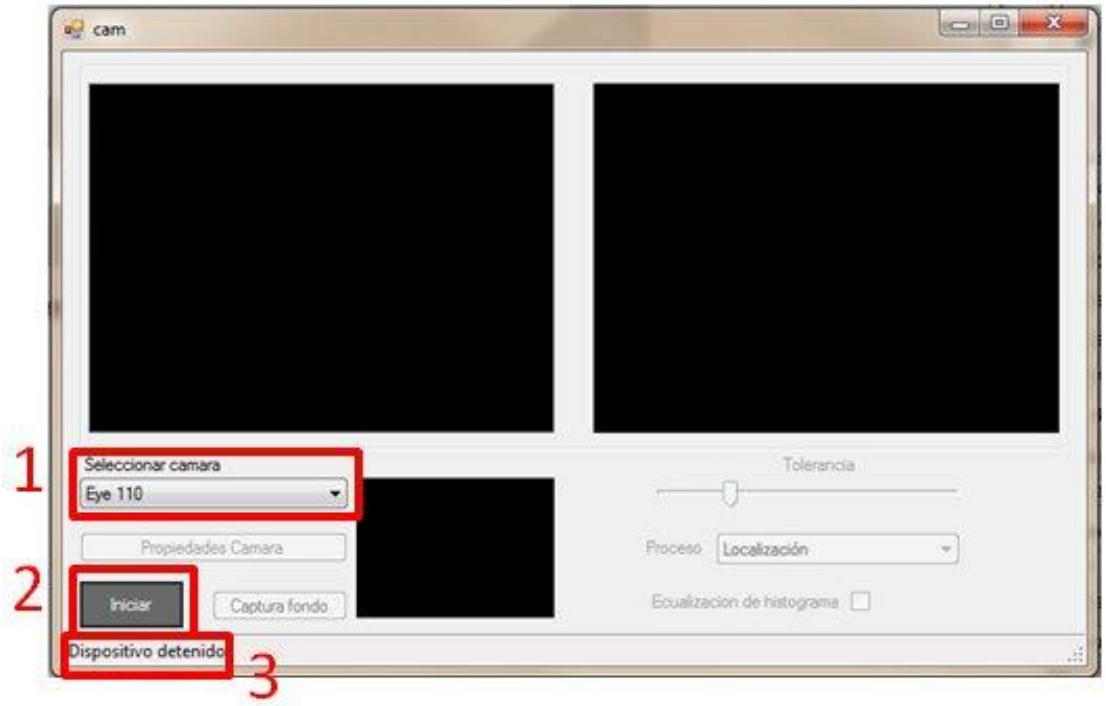
Manual de usuario

Controles de la aplicación y su uso:

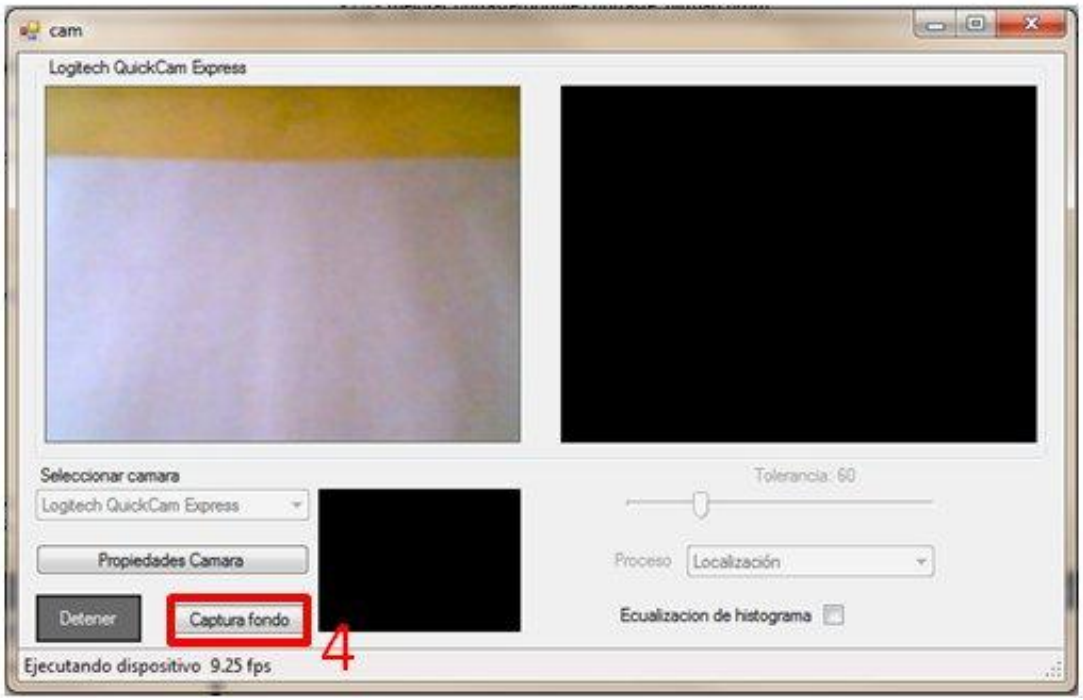
1. Seleccionar cámara: en este combo box se muestran los dispositivos detectados, es necesario que los dispositivos sean conectados antes de iniciar la aplicación, si se conectan después se requiere cerrar y volver a ejecutar para que detecte los nuevos dispositivos.

2. Iniciar: con este botón se inicia el stream de video de la cámara seleccionada en el punto anterior, o si ya se encuentra una captura de video, este botón indicara detener.

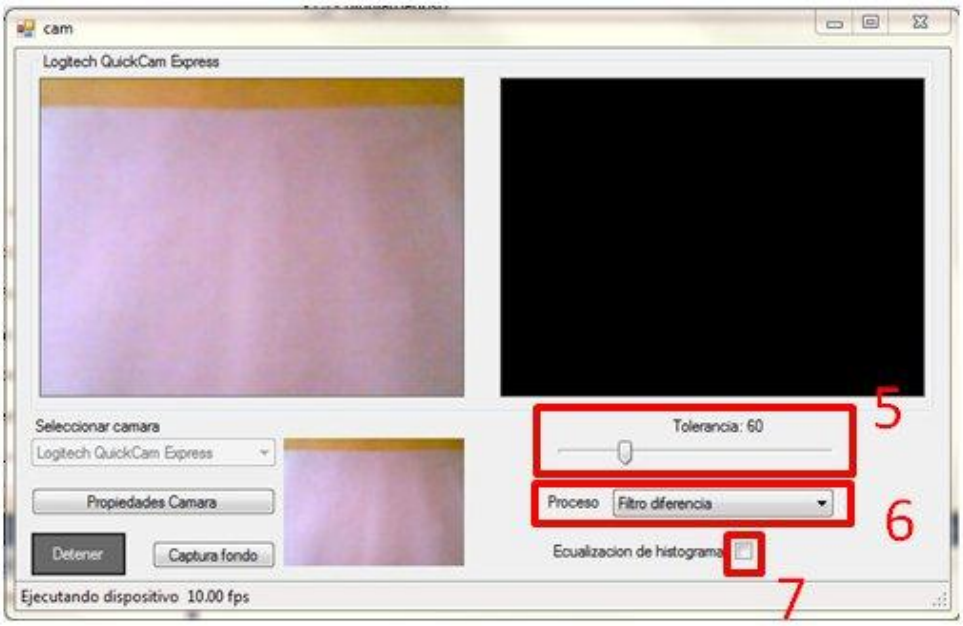
3. Este mensaje indica el estado del dispositivo de video (detenido o en ejecución) o en caso de no encontrarlo mostrara el mensaje "No se encontró alguna cámara".



4. Captura de fondo: este botón sirve para capturar el fondo que se utilizara para la sustracción de imágenes, solo disponible si existe una captura de video.

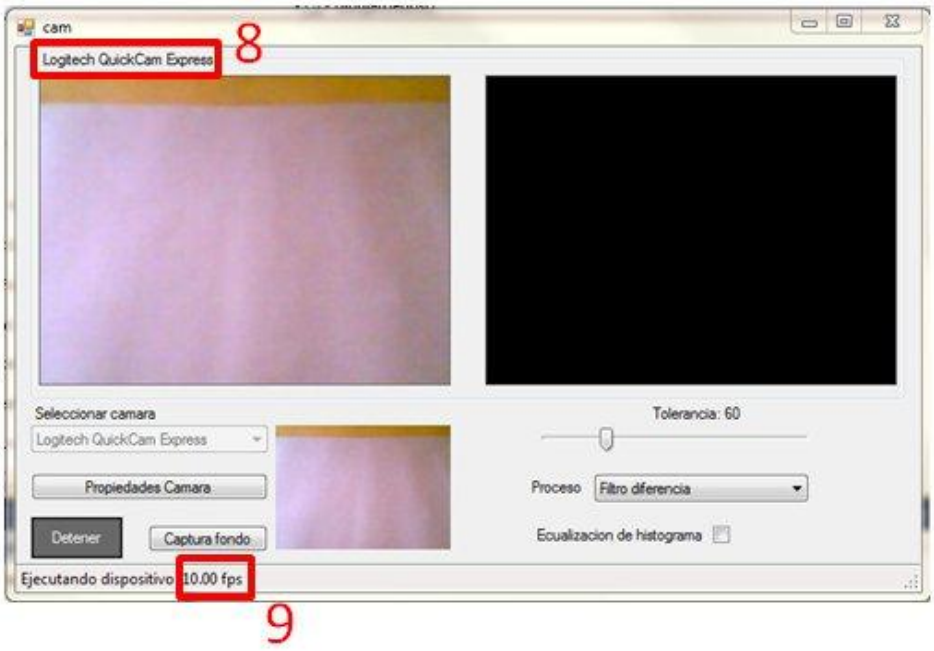


5. tolerancia: este trackbar sirve para ajustar el valor de la tolerancia.
6. Proceso: sirve para elegir el proceso que se desea ver.
7. Ecualización de histograma: sirve para utilizar la ecualización de histograma a cada imagen del video capturado.



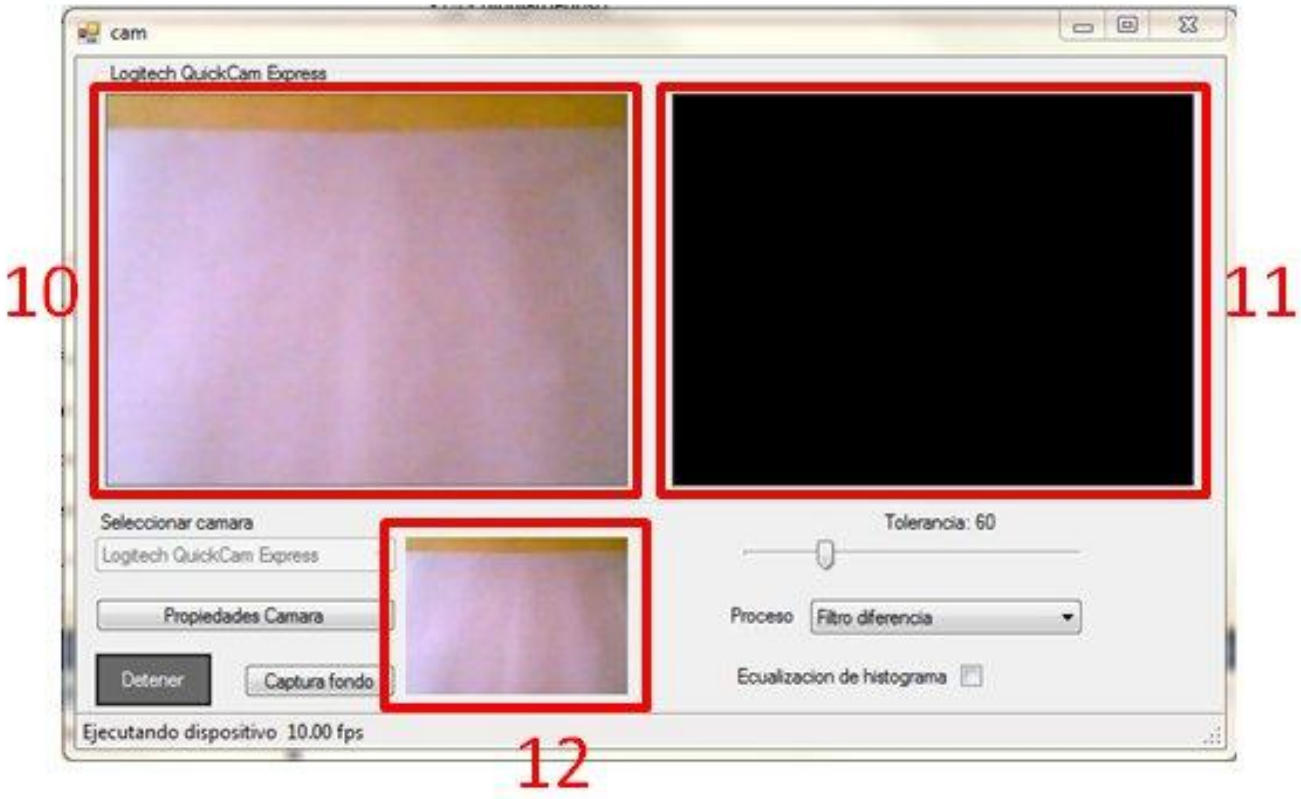
Esos son todos los controles que se tienen en la aplicación, ahora se muestran los mensajes que se muestran en la aplicación.

8. Etiqueta con el nombre del dispositivo de captura.
9. Tasa de cuadros por segundo que se están capturando.



Las imágenes son mostradas en tres distintos recuadros de la aplicación:

10. Recuadro de presentación del stream de video sin procesar.
11. Recuadro de presentación del video procesado.
12. Recuadro de presentación de la captura del fondo.



Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte de Proyecto Terminal

Estimación de pose de una mano a través de captura monocular

Coria Flores José Luis

Matricula: 202314318

_____

Trimestre 12 Primavera

Agosto de 2012

Risto Rangel-Kuoppa

Dr. en Ciencias de la Computación

Departamento de Sistemas

_____

Tabla de contenido

Acerca de este reporte:

El presente toca los puntos mencionados en el plan de trabajo de la Propuesta de Proyecto Terminal para las UEA's Proyecto Terminal 1 y Proyecto Terminal 2:

Proyecto Terminal 1:

1. Diseñar e implementar un módulo para la adquisición de video a través de un puerto USB.
2. Diseñar e implementar un módulo para la presentación del stream de video en pantalla.
3. Implementar algoritmo de ecualización de histograma para el mejoramiento del video.
4. Implementar algoritmo de obtención del borde de la mano.

Proyecto Terminal 2:

5. Implementar el algoritmo de esqueletización de la mano.
6. Diseñar e implementar un módulo para la estimación de una mano en tres dimensiones.
7. Probar el sistema en diferentes condiciones de uso.
8. Elaboración de manuales.

Diseñar e implementar un módulo para la adquisición de video a través de un puerto USB.

Para implementar la adquisición de video se utilizaron bibliotecas de Aforge.NET en específico las funciones de AForge.Video.DirectShow, con la cual se obtiene acceso a los datos de un dispositivo de captura de video local, como lo es una cámara web usb, además de configurar algunas opciones tales como el tamaño de video adquirido y la tasa de capturas deseado.

En la se muestra las distintas cámaras web que se utilizaron para probar el sistema y como son mostradas al usuario, además de las opciones de configuración de cada una.

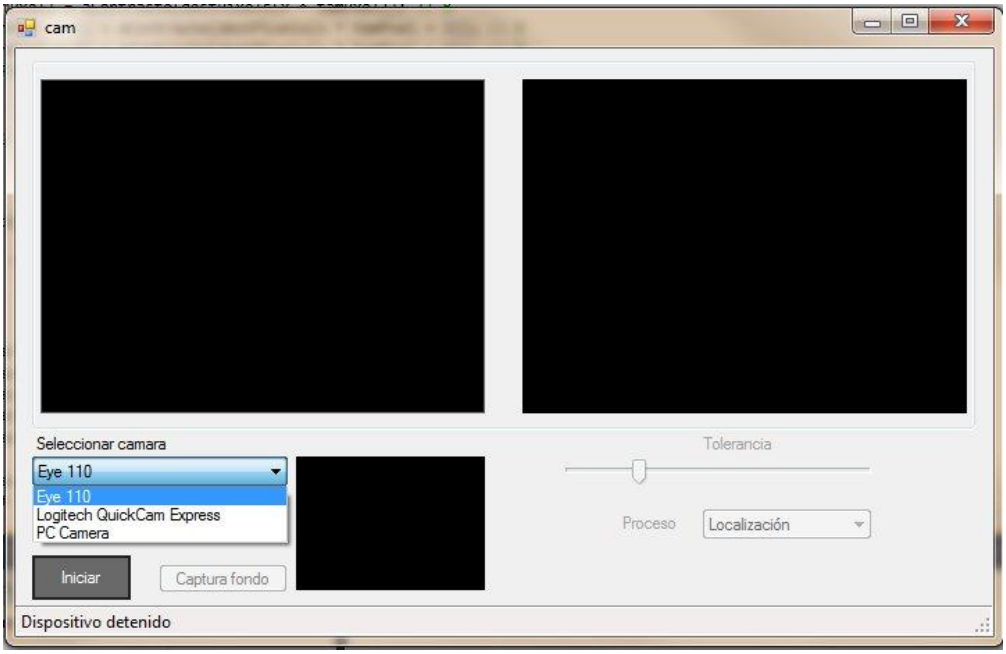


Figura 1. Cámaras web usb detectadas. Imagen tomada de mi aplicación.

Diseñar e implementar un módulo para la presentación del stream de video en pantalla

Para la presentación del stream de video también se utilizó las funciones de AForge.Video.DirectShow, el video sin procesar se muestra en el recuadro izquierdo, el video procesado del lado derecho y en el recuadro inferior la imagen cuando se obtiene el fondo que se requiere utilizar para la sustracción de imágenes.

Durante el desarrollo del sistema se observó que la tasa de cuadros por segundo varía dependiendo de diversos factores, como son la potencia de la computadora que está ejecutando el sistema, la cámara utilizada para obtener el video y por ultimo si es que se está procesando el video o solo se está viendo el stream capturado, ver Figura 2.

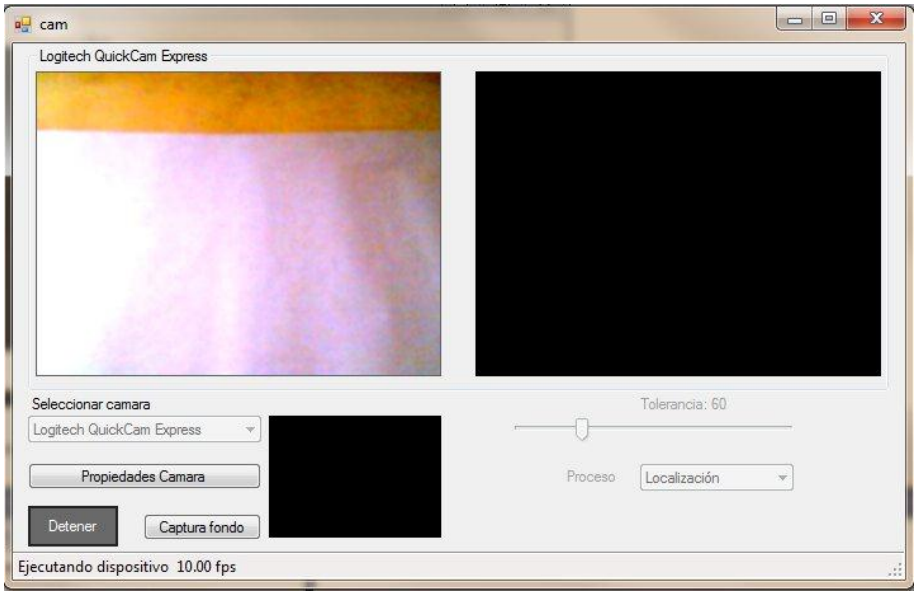


Figura 2. Presentación de stream de video. Imagen tomada de mi aplicación.

Implementar algoritmo de ecualización de histograma para el mejoramiento del video.

Para la mejora del video se utilizaron dos procesos distintos que el usuario puede elegir según le parezca más conveniente, el primero se utiliza la función mejoraContraste() (opción por defecto, ver Figura 3) para mejorar el contraste, la segunda opción se utiliza la ecualización de histograma mediante la utilización de la biblioteca de Emgu CV, en específico la función de _EqualizeHist() la cual normaliza el brillo e incrementa el contaste, esta última solo es aconsejada que se use si es que se tiene un contraste muy pobre en la imagen, ver Figura 4.

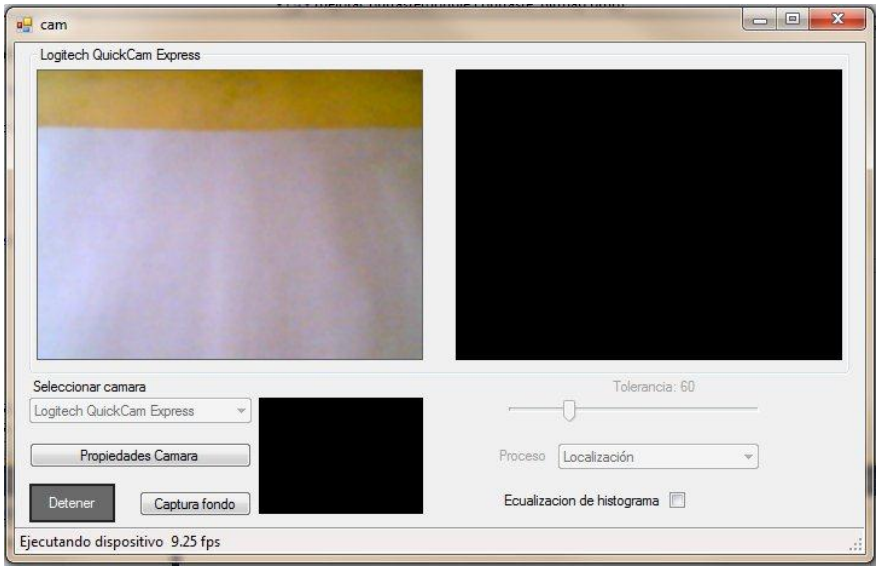


Figura 3. Video sin ecualización de histograma. Imagen tomada de mi aplicación.

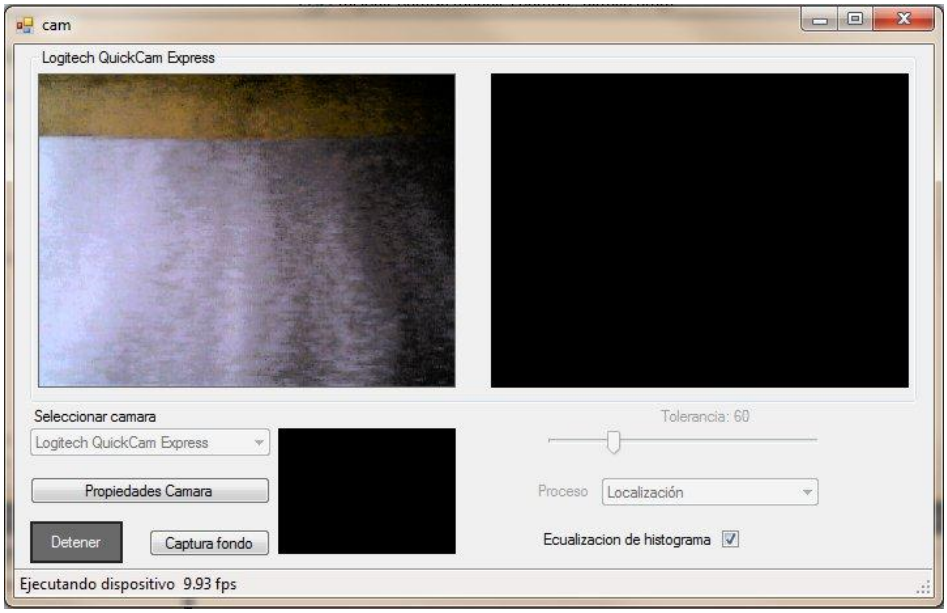


Figura 4. Video con ecualización de histograma. Imagen tomada de mi aplicación.

Como se puede observar en las imágenes anteriores, no siempre es recomendable usar la ecualización de histograma.

Implementar algoritmo de obtención del borde de la mano.

Para la obtención del borde de la mano se utilizó la sustracción de fondo, este método utiliza una imagen de fondo (ver) como base comparándola con las imágenes donde se encuentra el objeto deseado, en este caso la mano, obteniendo así solo el objeto de nuestro interés y desechando lo que no corresponde a este, en este proceso se devuelve una imagen de solo dos colores blanco (para el objeto) y negro (para todo lo demás), donde claramente se puede visualizar únicamente el objeto, ver Figura 6.

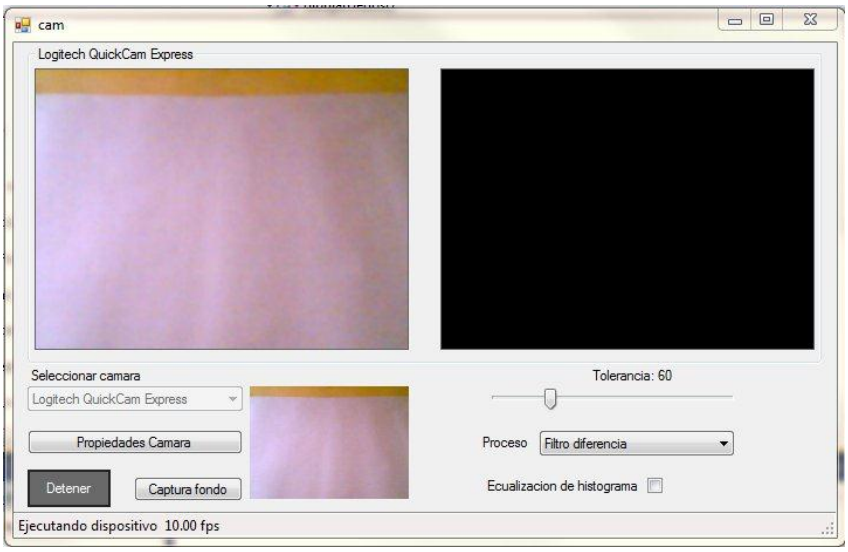


Figura 5. Obtención del fondo. Imagen tomada de mi aplicación.

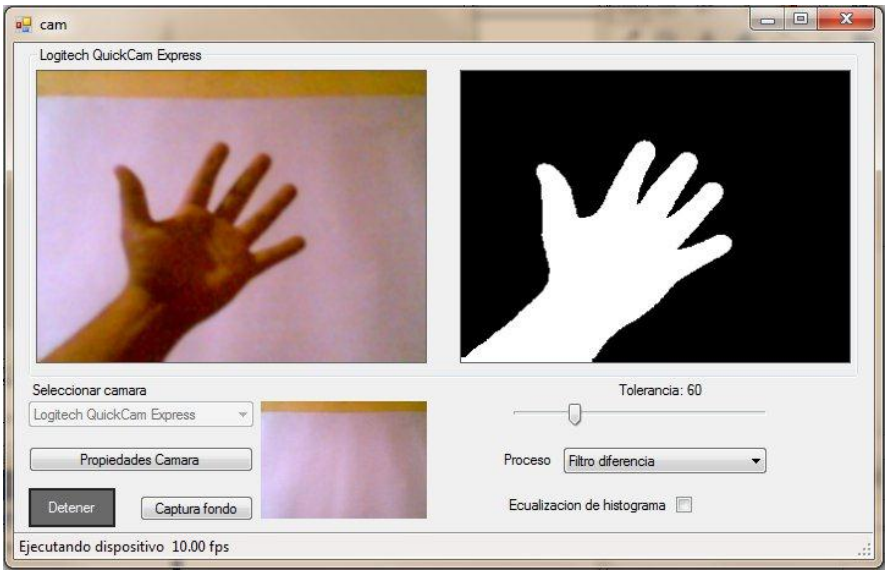


Figura 6. Sustracción del fondo. Imagen tomada de mi aplicación.

Además de la sustracción de fondo se utiliza otro filtro para evitar que la imagen tenga en lo posible ruido o partículas no pertenecientes al objeto, este es el filtro de erosión, el filtro de erosión quita las partículas más pequeñas dejando solo los objetos más grandes, ver Figura 7 ejemplo del proceso de erosión.

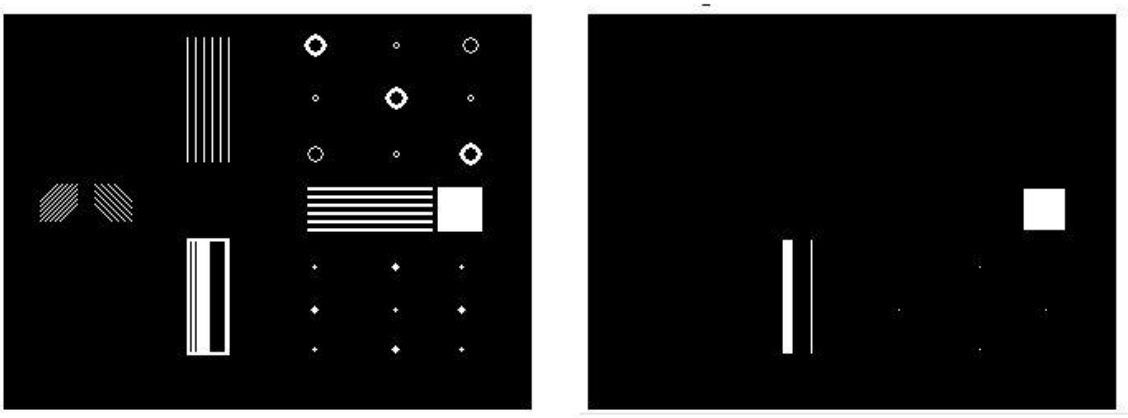


Figura 7. Izquierda imagen inicial, derecha imagen procesada. Imagen tomada de la documentación Aforge.Net

Implementar el algoritmo de esqueletización de la mano

El algoritmo de esqueletización pretende obtener de la imagen, un patrón continuo que contenga la menor cantidad posible de datos, pero que contenga la estructura original del objeto del que se desea obtener su esqueleto.

El proceso de esqueletización consiste en quitar de un patrón la mayor cantidad de pixeles posibles sin afectar la estructura general del patrón, es decir después de que los pixeles han sido quitados, el patrón debe ser reconocible, en general el esqueleto debe ser: tan delgado como sea posible, conectado y centrado, en la Figura 8 se puede observar la obtención de un esqueleto de una imagen.

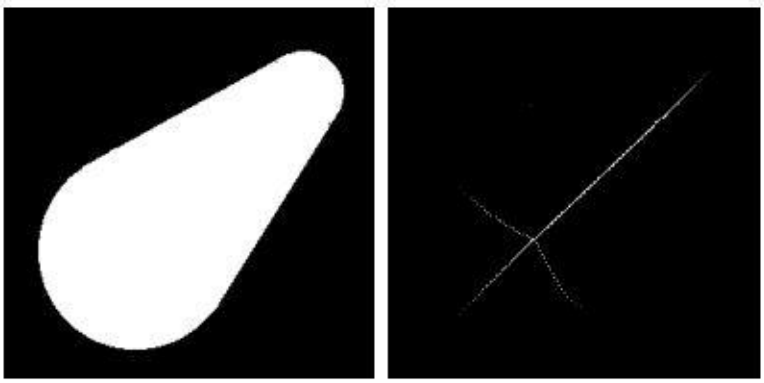


Figura 8.Esqueleto de un objeto. Imagen tomada del libro Digital Image Processing.

Para la implementación de la esqueletización de la mano se utilizó el filtro SimpleSkeletonization del framework Aforge.net; el cual obtiene el esqueleto de un objeto mediante el adelgazamiento de estos, hasta que tengan el ancho de un pixel, este filtro utiliza imágenes en dos colores, solo blanco y negro, es decir sin escalas de grises, utilizando el negro para identificar el fondo y el blanco para reconocer el objeto del que se quiere obtener el esqueleto. En la Figura 9 se aprecia como el filtro obtiene el esqueleto de la imagen procesada.

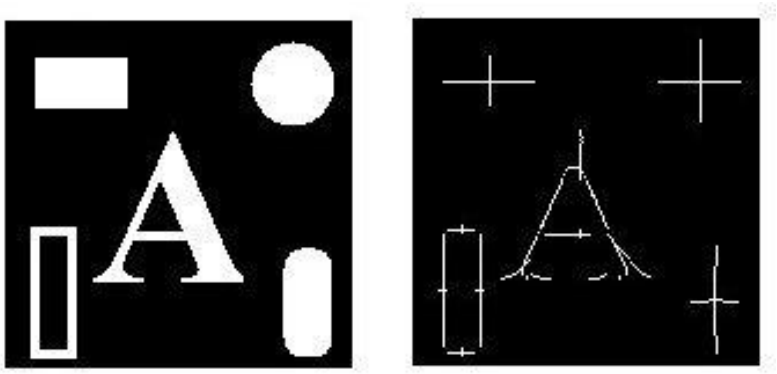


Figura 9. Izquierda, la imagen original, derecha la imagen procesada. Imagen tomada de la documentación Aforge.Net

Este proceso se realiza a cada fotograma del video, cabe mencionar que este proceso de obtención del esqueleto no es el que mejor resultados otorga en cuanto a conectividad, pero si nos da una muy buena aproximación de este y en poco tiempo de procesamiento, lo cual es importante para este proyecto, que debe procesar las imágenes en tiempo real.

Antes de realizar esta operación se tiene que realizar una sustracción de fondo dejando solamente el objeto de nuestro interés, esto se logra con los procesos desarrollados en el proyecto terminal1, que son

Diseñar e implementar un módulo para la estimación de una mano en tres dimensiones

Al ser una captura monocular la ubicación en 3 dimensiones es solo una estimación, debido a la falta de profundidad inherente a una sola fuente de video, en este caso una webcam.

Es por ello que se diseñó una estimación de la posición de la mano en el espacio a través de la obtención de distintos puntos, en los que se identifica la punta de cada uno de los dedos su unión con la palma de la mano y también el centro del contorno de toda la mano esto último para poder estimar la profundidad en la que se encuentra esta.

Para dibujar en la imagen estos puntos de referencia se utiliza el siguiente procedimiento:

- Se obtiene el contorno del objeto de mayor tamaño detectado en la imagen, ver Figura 10.

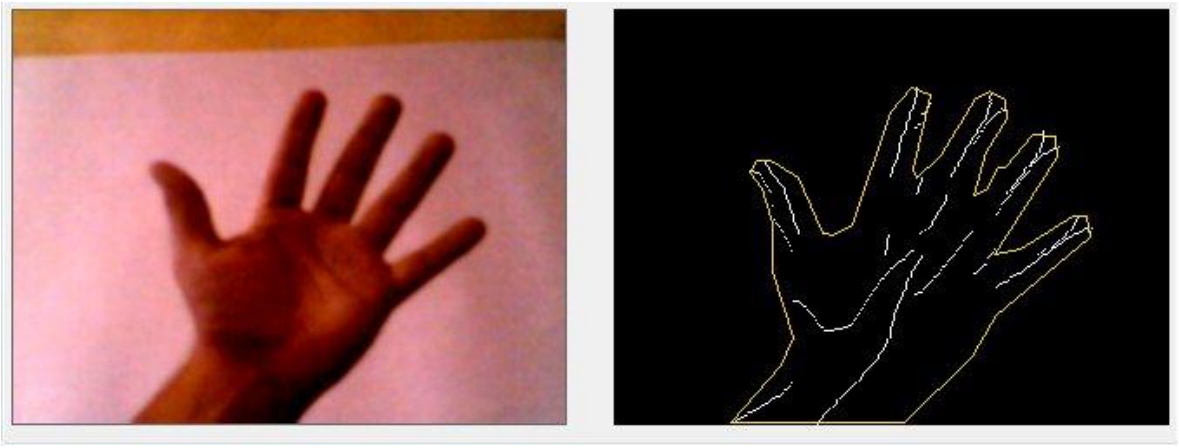


Figura 10. Contorno objeto más grande. Imagen tomada de mi aplicación.

- Se calcula el centro de este contorno, indicado con un punto azul, ver Figura 11.

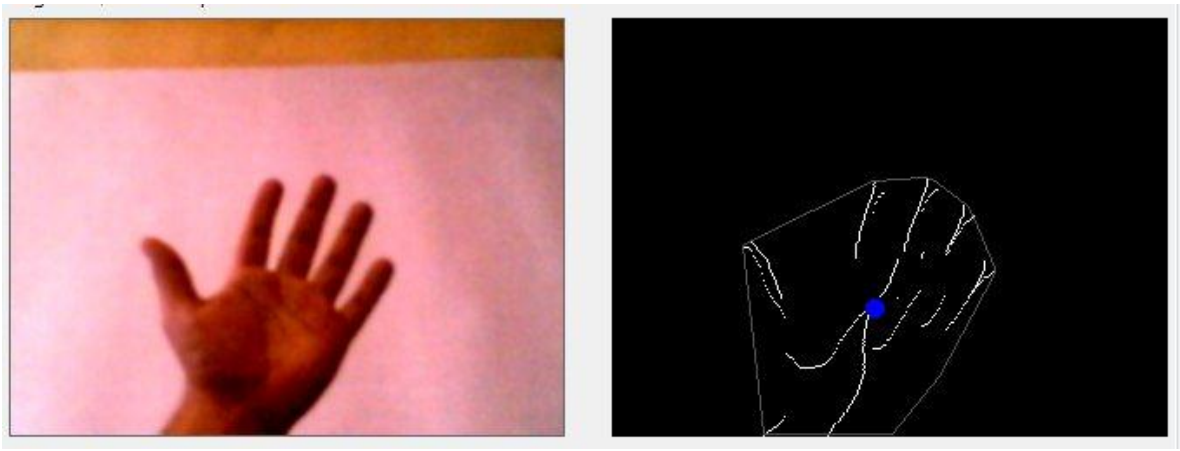


Figura 11. Centro de todo el objeto. Imagen tomada de mi aplicación.

- En el contorno se buscan las distintas concavidades que puedan encontrarse en este.
- Se muestran los puntos en rojo la detección de la punta de cada dedo y de color azul claro la unión con la palma, ver Figura 12.

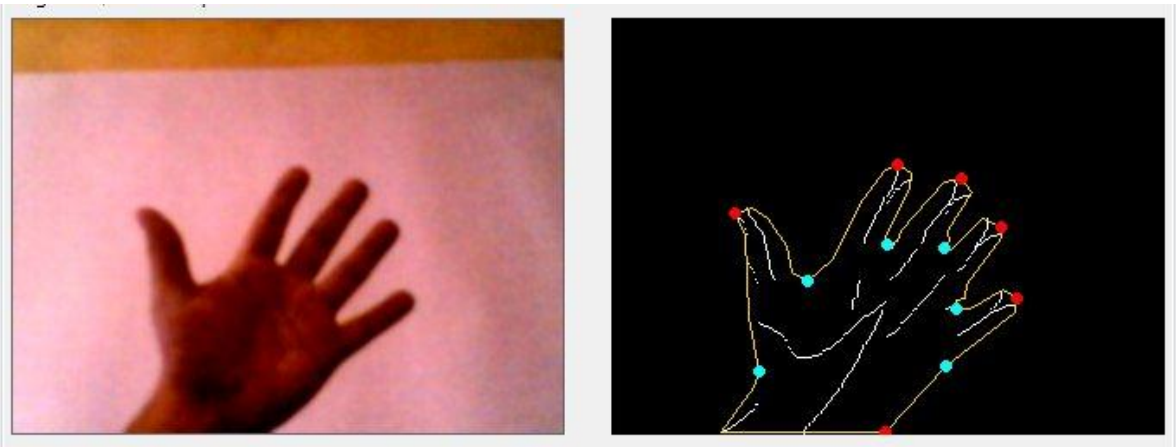


Figura 12. Mostrando los puntos encontrados. Imagen tomada de mi aplicación.

- Al tener todos esto pasos reunidos, se puede estimar la posición de cada dedo mostrado en la Figura 13 con color verde.

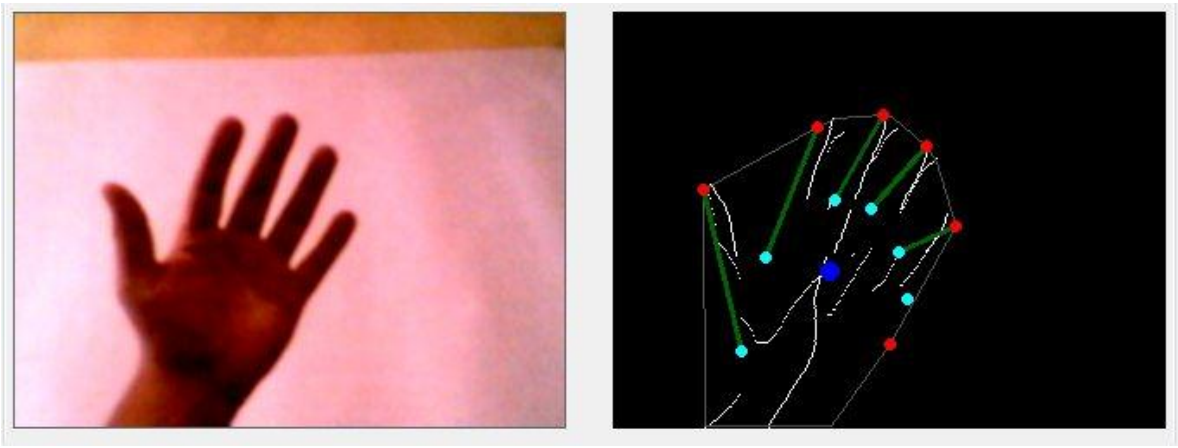


Figura 13. En color verde la estimación de la posición de los dedos. Imagen tomada de mi aplicación.

- Al final de todos estos procesos podemos estimar la pose de la mano como se ve en la Figura 14 y Figura 15 .

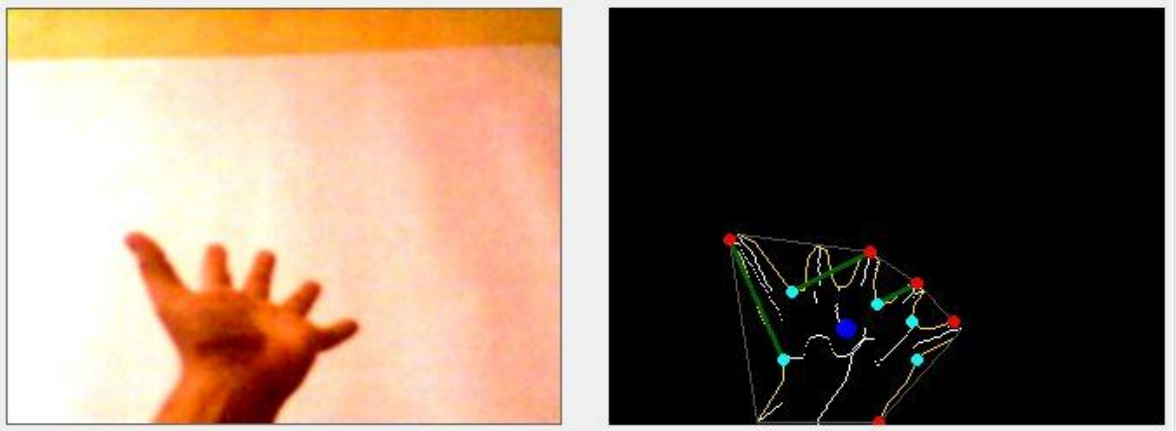


Figura 14. Estimación de pose. Imagen tomada de mi aplicación.

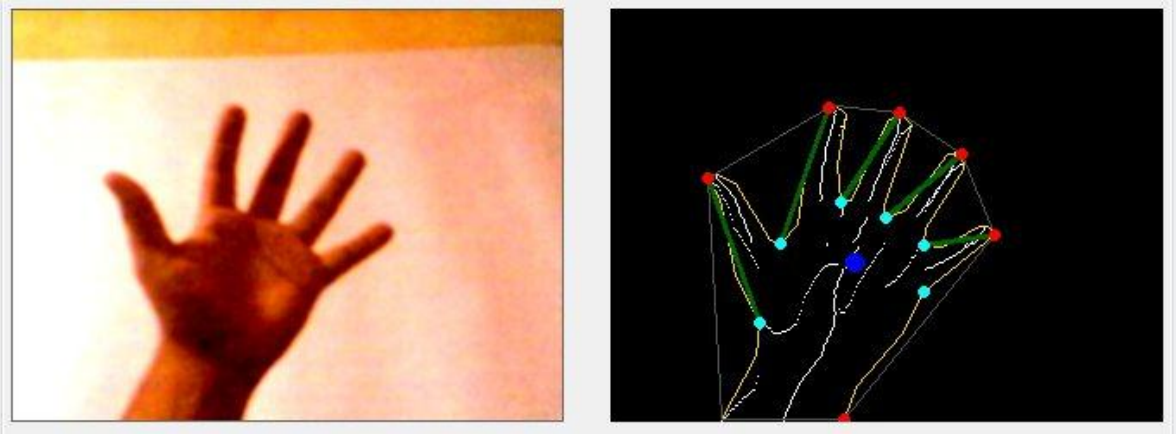


Figura 15. Estimación de pose. Imagen tomada de mi aplicación.

Probar el sistema en diferentes condiciones de uso

El sistema se probó en los sistemas operativos Windows xp y Windows 7, con tres cámaras web distintas.

Hay que recalcar que para que la estimación sea correcta deberá tener un fondo contrastante con el color de la mano que se desea detectar, además de configurar la cámara para que no ajuste automáticamente el balance de blanco y la exposición, los ajustes automáticos deberán ser desactivados, si se requiere ajustar la configuración de la cámara deberá realizarse manualmente.

Las pruebas fueron las siguientes:

Antes de ejecutar la aplicación se debe tener instalado .NET Framework 4, de lo contrario saldrá un mensaje de error y la aplicación no se ejecutara; además la cámara que se utilizara debe estar conectada antes de ejecutar para que sea reconocida en la aplicación.

- En Windows xp:

Con cámara web Logitech QuickCam Express e iluminación fluorescente.

Aun quitando la configuración automática la cámara realiza ajustes de iluminación, por lo cual la estimación no se realiza correctamente, ver Figura 16.

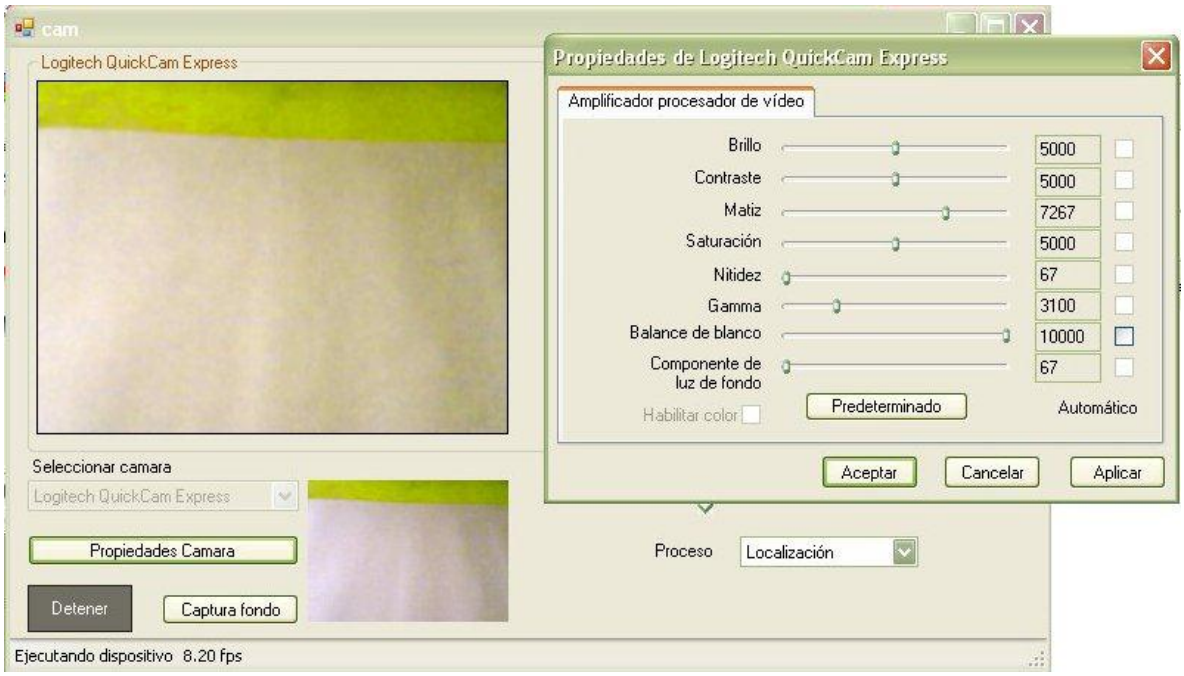


Figura 16.Ajustes cámara web Logitech QuickCam Express. Imagen tomada de mi aplicación.

Al principio de la ejecución de la aplicación, todo parece correcto pero después de unos segundos la cámara realiza un ajuste automático de balance de blanco, lo que provoca una incorrecta sustracción con respecto a la imagen de fondo, ver,

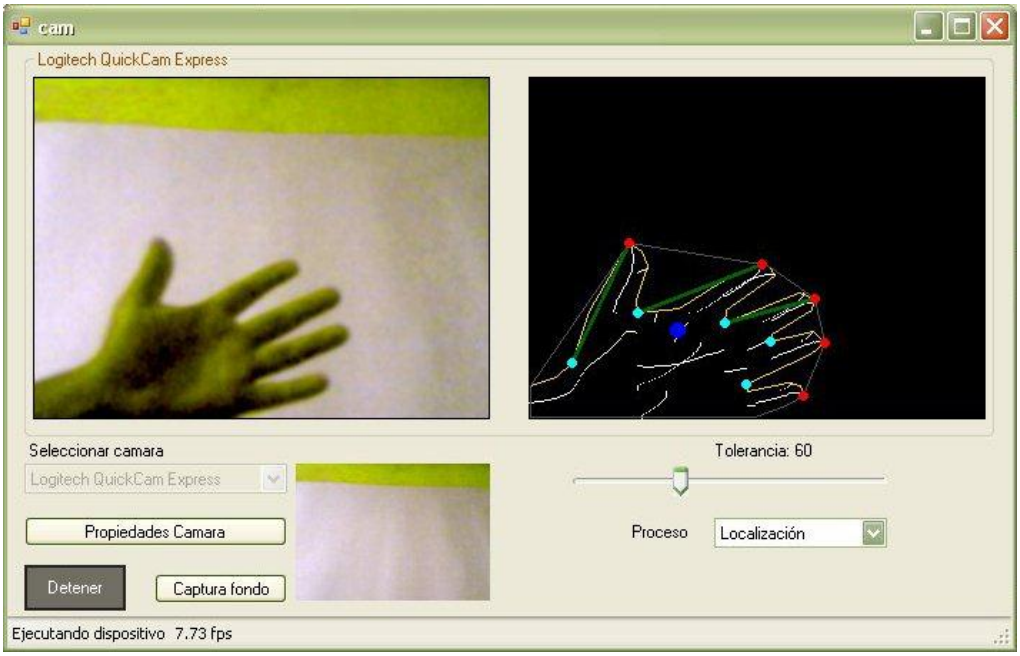


Figura 17. Prueba cámara web Logitech QuickCam Express. Imagen tomada de mi aplicación.

Figura 18. Incorrecta estimación de pose. Imagen tomada de mi aplicación.

Aunque el cambio es mínimo, es lo suficiente para afectar la correcta estimación de la mano.

Utilizando la cámara web Genius eye 110 e iluminación fluorescente.

Al igual que la cámara Logitech, se desactivaron los cambios automáticos, en la configuración accesible al usuario, evitando que estos cambios afecten a la correcta sustracción de fondo (ver Figura 19) y por consecuencia la estimación de la posición de la mano, ver Figura 20.

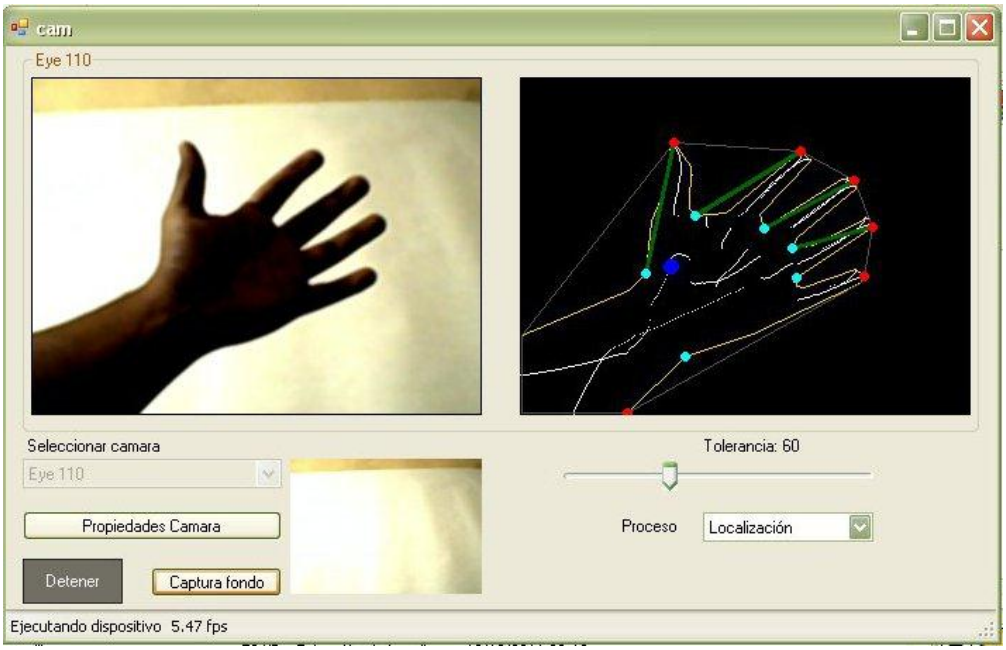


Figura 19. Prueba cámara web Genius eye 110. Imagen tomada de mi aplicación.

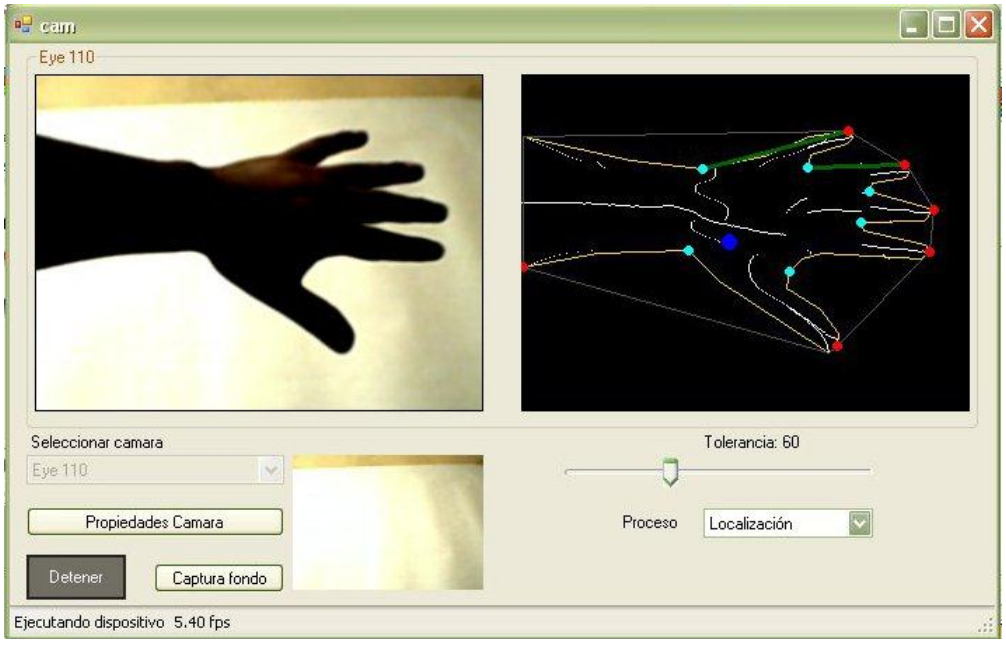


Figura 20. Después de unos segundos. Imagen tomada de mi aplicación.

Después de algunos segundos no existió cambio alguno de ajuste por parte del software de la cámara, con lo que no hubo problemas para estimar la pose de la mano.

Utilizando la cámara Hp wb918la#abm e iluminación fluorescente.

Se configuro la cámara para desactivar las correcciones automáticas de balance de blanco y exposición, que realiza el software que incluye esta. Ver Figura 21 y Figura 22.

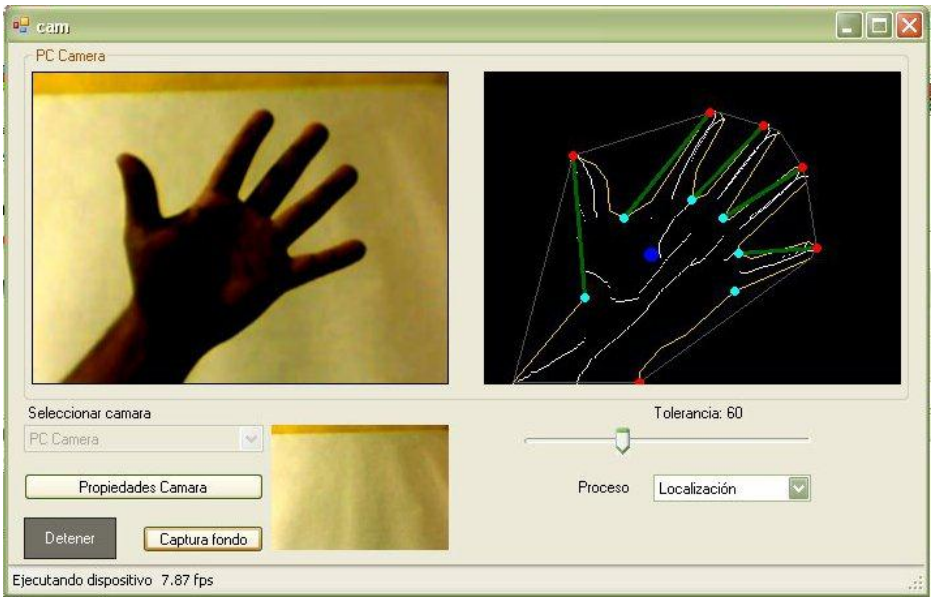


Figura 21. Probando cámara HP wb918la#abm. Imagen tomada de mi aplicación.

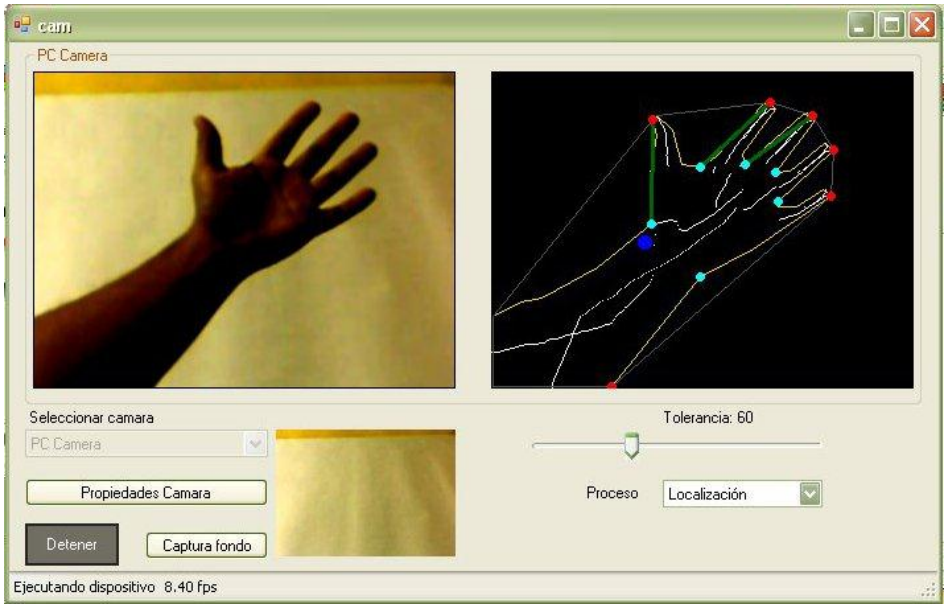


Figura 22.Después de unos segundos. Imagen tomada de mi aplicación.

En las diferentes pruebas en el sistema operativo Windows xp se observa que la estimación de la pose de la mano es correcta, solo destaca el caso de la cámara Logitech que realiza cambios automáticos, en las otras dos cámaras no se presentó este inconveniente.

- En Windows 7:

Con cámara web Logitech QuickCam Express e iluminación fluorescente.

En Windows 7 aparece otra pestaña de configuración que no aparece en Windows xp (ver Figura 23, control automático de ganancia), al desactivar esta opción ya no se producen cambios automáticos por parte del software de la cámara, por lo que la estimación de la pose ahora si es correcta, ver Figura 23 y Figura 24.

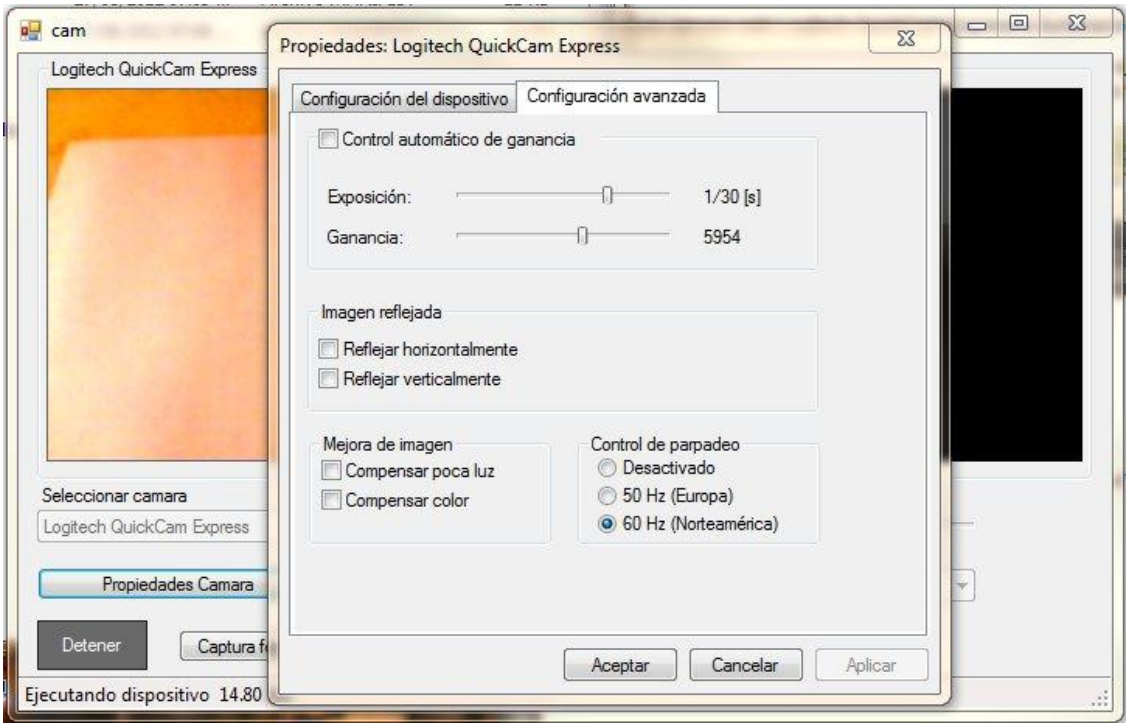


Figura 23.Opciones de configuración. Imagen tomada de mi aplicación.

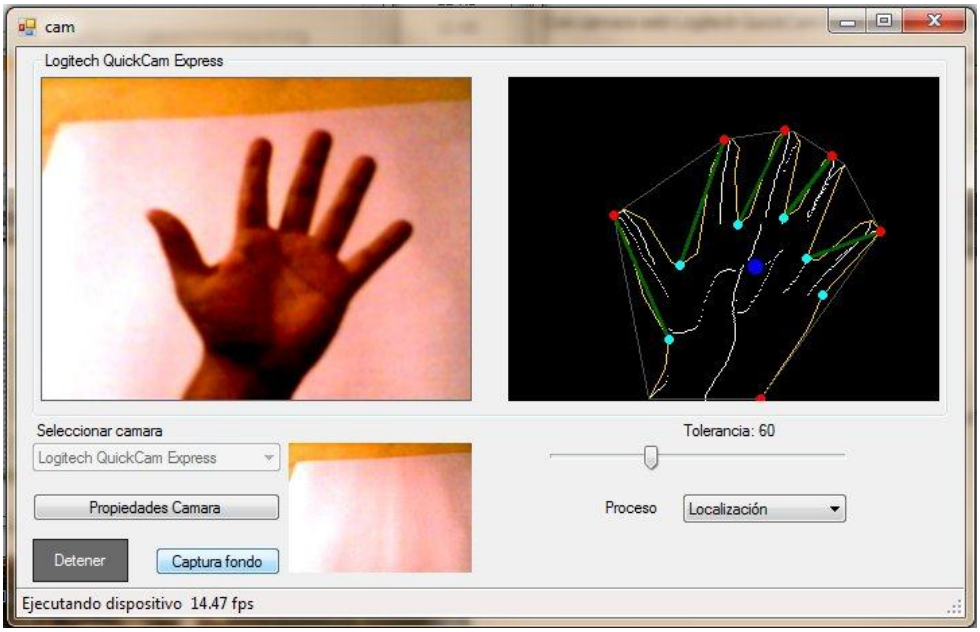


Figura 24. Probando la aplicación. Imagen tomada de mi aplicación.

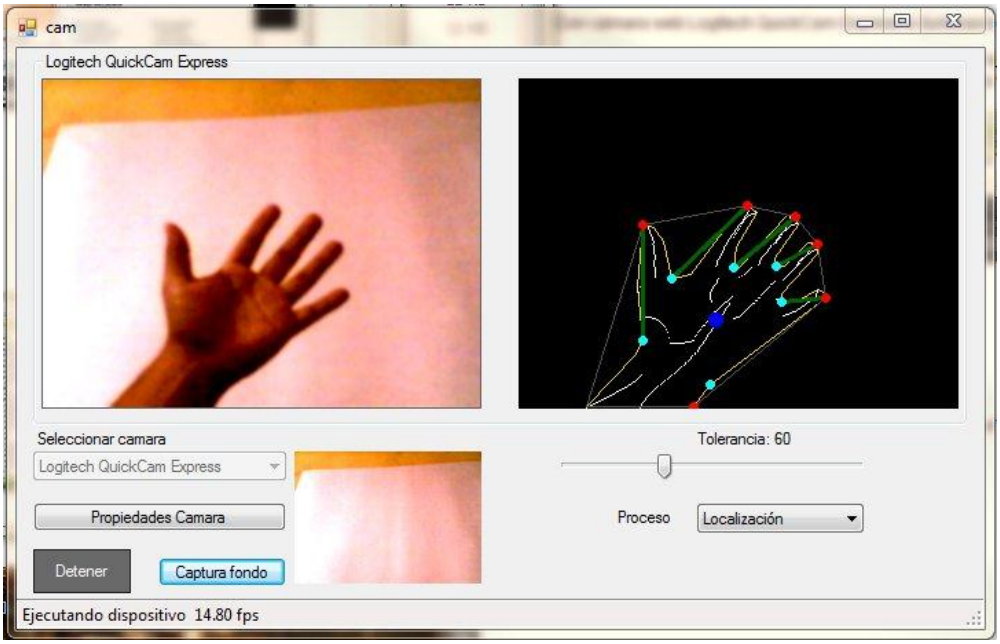


Figura 25. Después de unos segundos. Imagen tomada de mi aplicación.

En este caso al probar la aplicación con esta cámara no se tuvieron problemas, se logró correctamente la estimación de la pose de la mano, ver Figura 27.

Con cámara web Genius eye 110 e iluminación fluorescente.

Esta vez también se probó la tolerancia que permite al proceso de sustracción de fondo obtener el objeto deseado, al incrementar este valor se puede observar que a pesar del cambio de iluminación se puede conseguir una correcta estimación de la pose de la mano, ver Figura 26 y Figura 27.

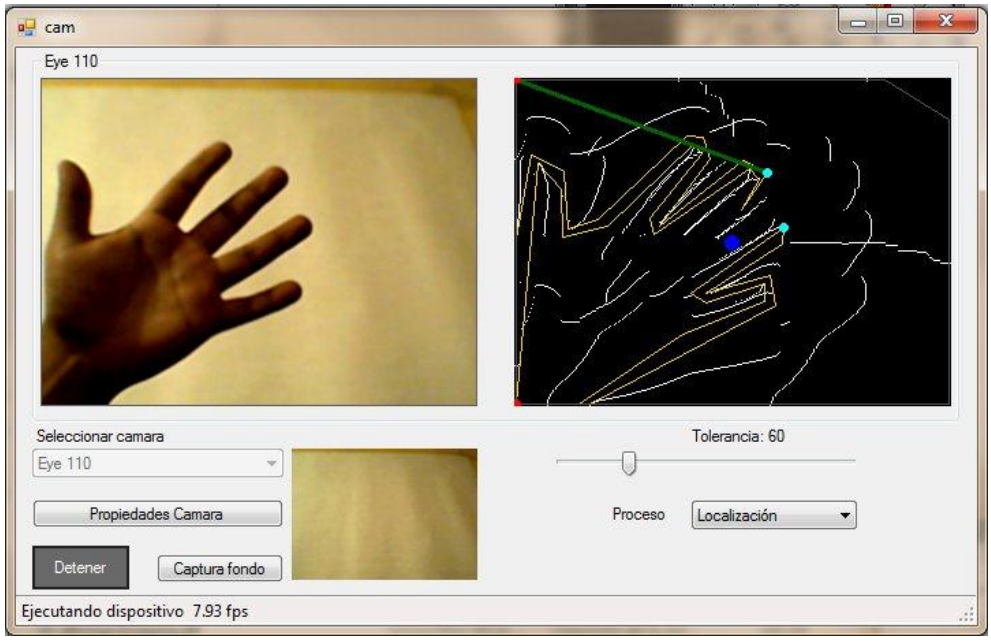


Figura 26. Incorrecta estimación por cambio de iluminación. Imagen tomada de mi aplicación.

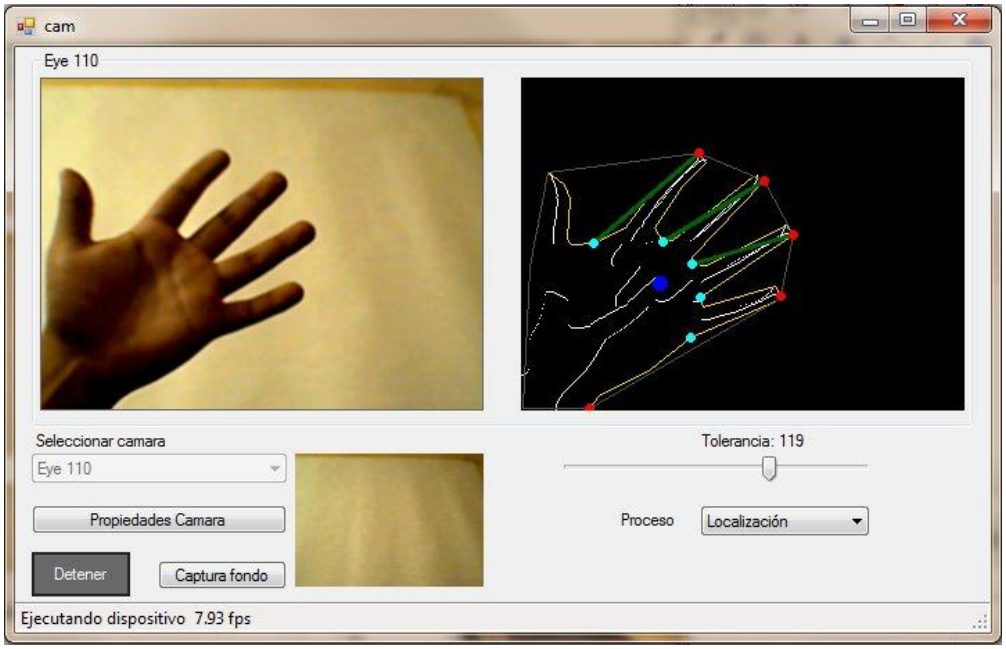


Figura 27. Ajustando el valor de tolerancia. Imagen tomada de mi aplicación.

Utilizando la cámara Hp wb918la#abm e iluminación fluorescente.

Al igual que en las pruebas anteriores se configuro la cámara para evitar que tuviera ajustes automáticos, aun así el software de esta realiza ajuste automático, por lo que es necesario cambiar el nivel de tolerancia para conseguir la estimación de pose de la mano, ver Figura 28 y Figura 29.

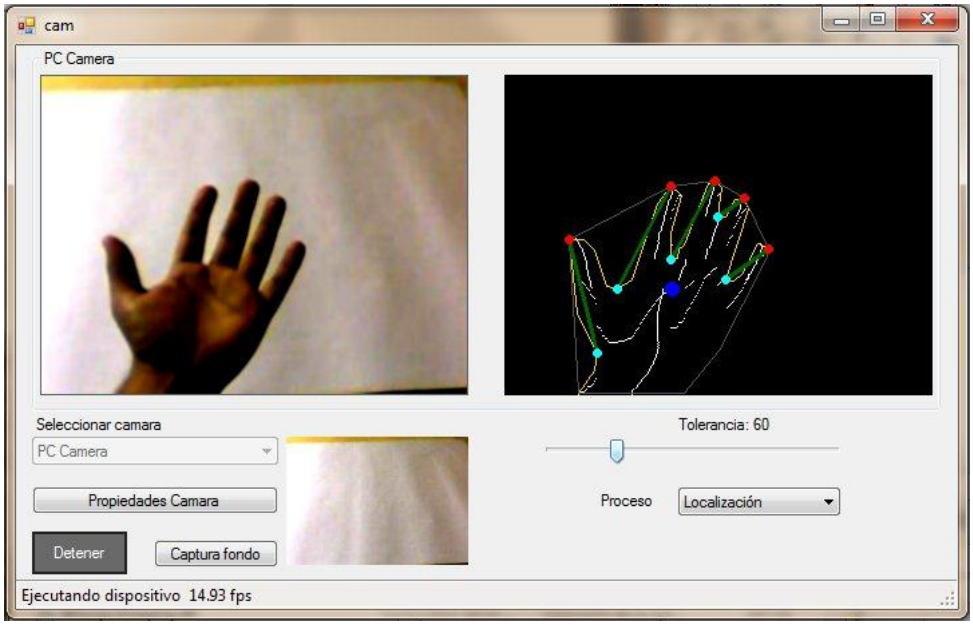


Figura 28. Probando cámara web Hp wb918la#abm. Imagen tomada de mi aplicación.

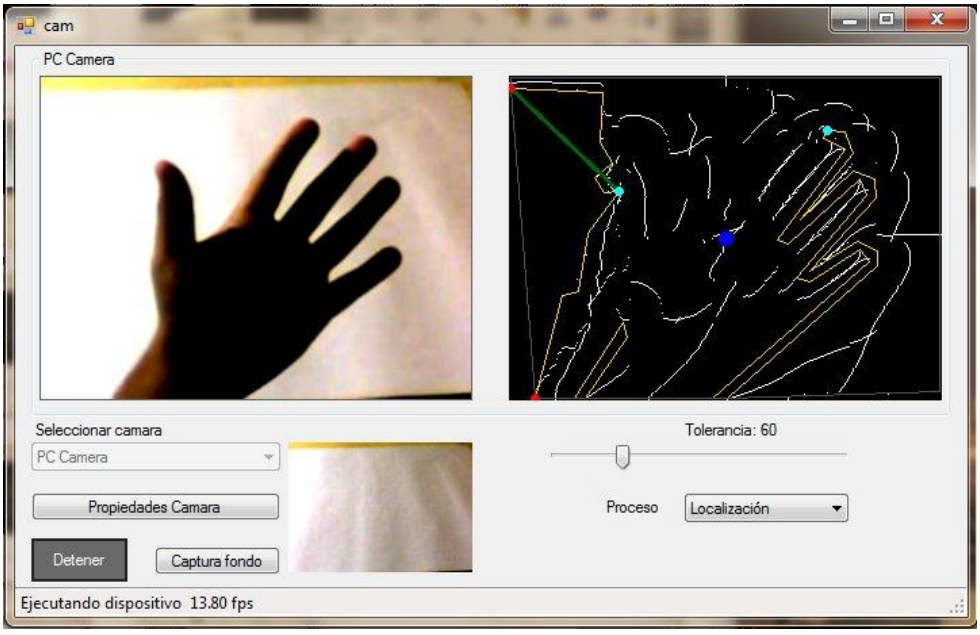


Figura 29. Después de unos segundos. Imagen tomada de mi aplicación.

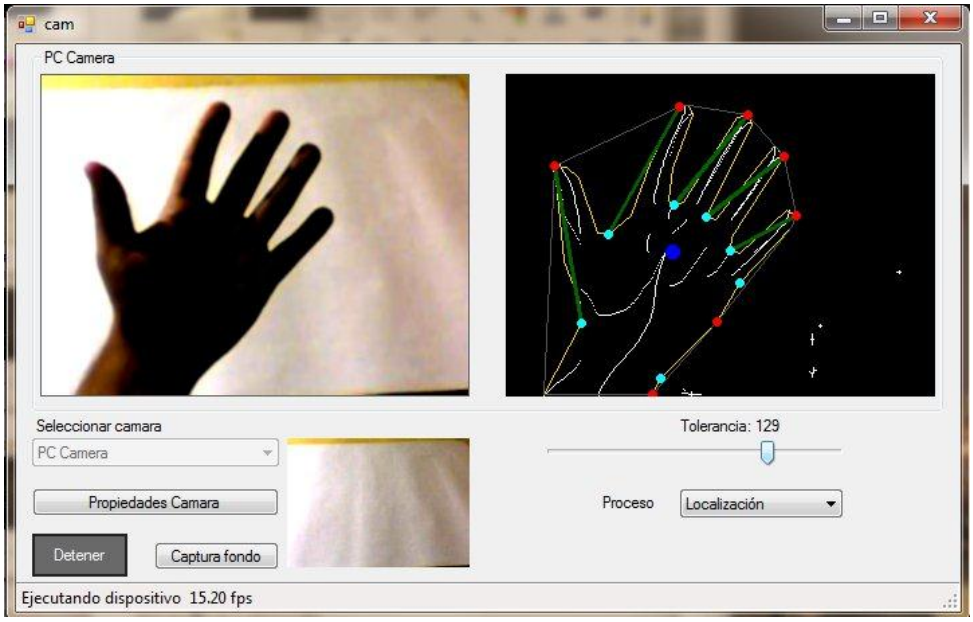


Figura 30. Ajustando el valor de tolerancia. Imagen tomada de mi aplicación.

Después de incrementar el valor de tolerancia se observa como se puede obtener la estimación correctamente aun después de un pequeño cambio en la iluminación de fondo, provocado por el software de la cámara web, ver Figura 30.

Conclusiones de las pruebas del sistema en diferentes condiciones de uso

En todas las pruebas se observó que el software de configuración de las cámaras, en modo de auto ajuste puede afectar considerablemente la obtención del objeto en el proceso de sustracción de imágenes, esto es debido a que cualquier ajuste automático que realice cambia notablemente la imagen de fondo que se utiliza para los procesos de detección del objeto. En las siguientes imágenes se puede ejemplificar el cambio de iluminación con los ajustes automáticos en la cámara Logitech QuickCam Express, en la imagen Figura 31 se está capturando el stream de video, en la Figura 32 se observa que no hay cambio de iluminación que afecte al stream de video, por último en la Figura 33 se observa un cambio notable en la iluminación del video debido al ajuste automático que realiza el software de la propia cámara web.

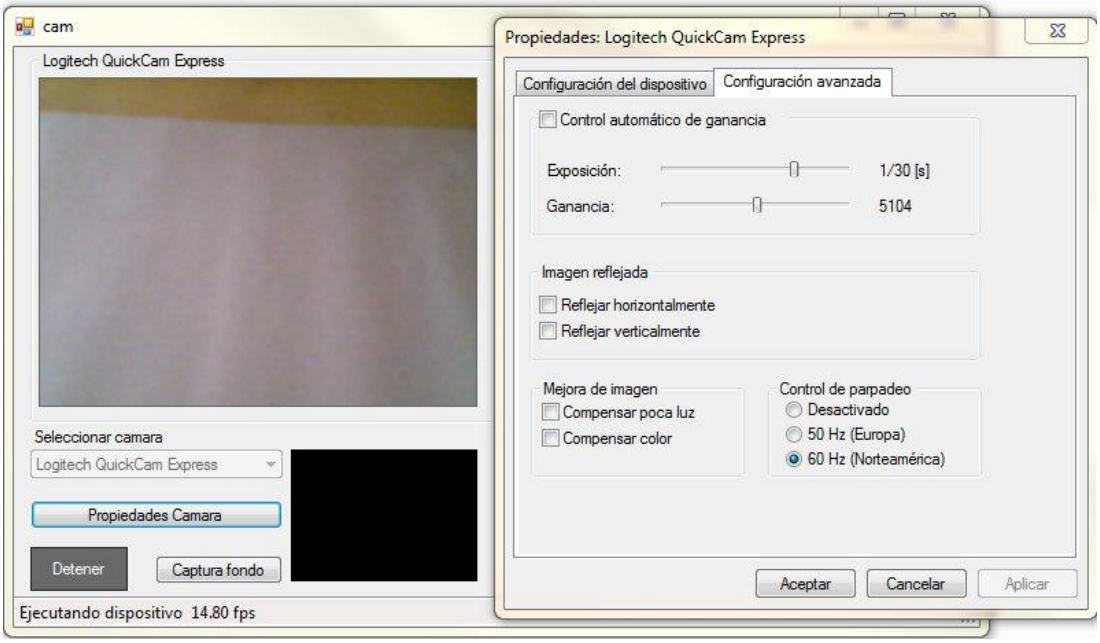


Figura 31. Captura de video sin ajustes automáticos. Imagen tomada de mi aplicación.

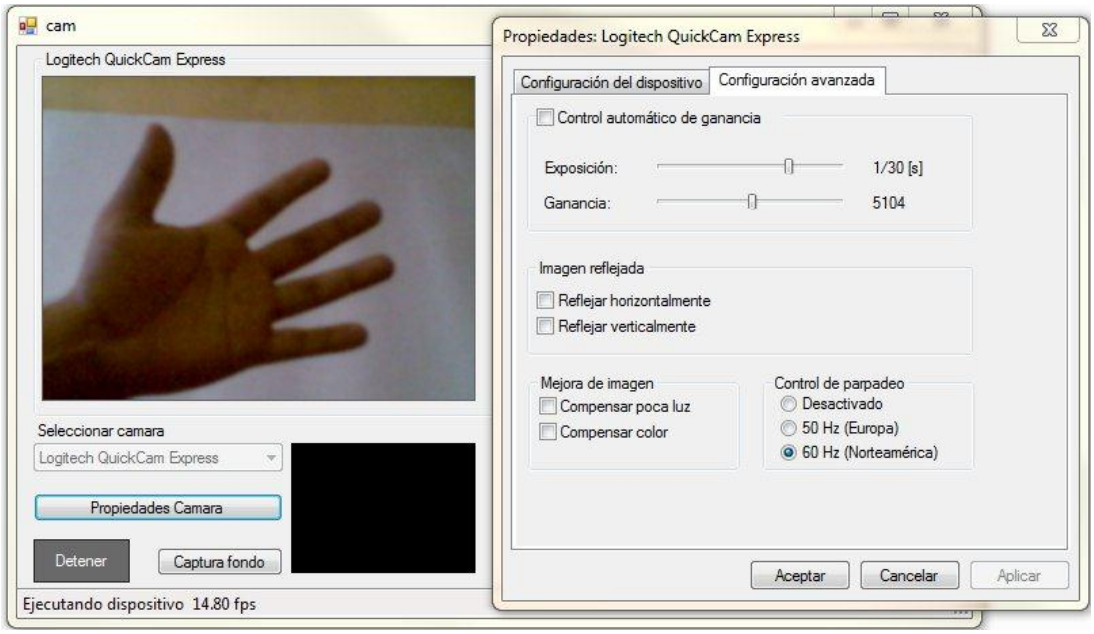


Figura 32. Sin ajuste automático no hay cambio de iluminación. Imagen tomada de mi aplicación.

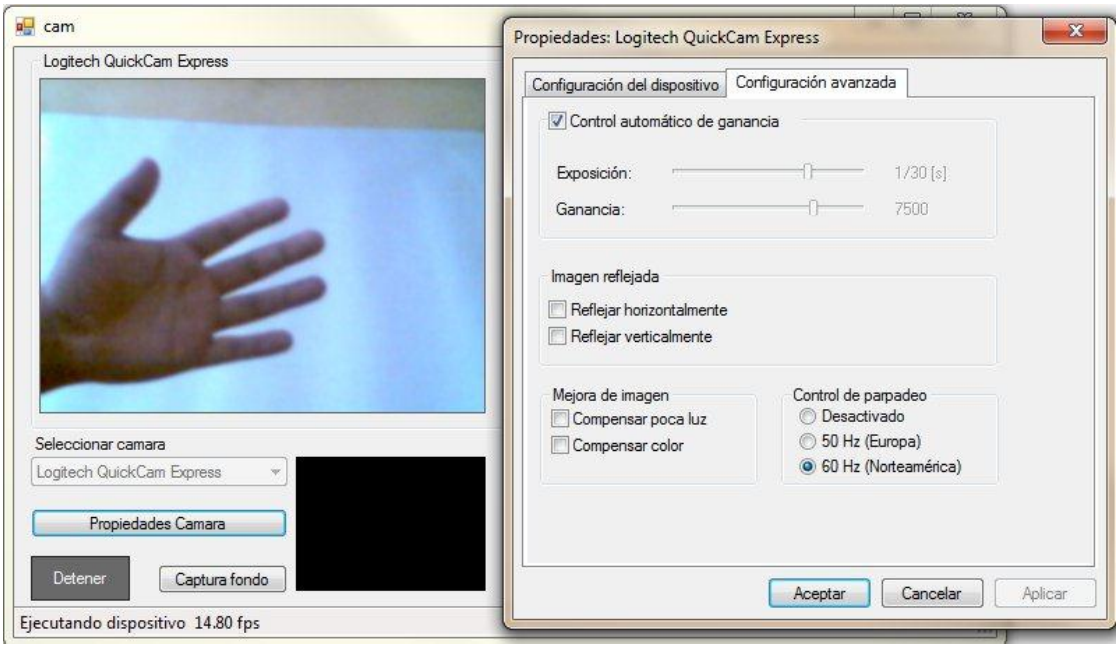


Figura 33. Con control automático de ganancia. Imagen tomada de mi aplicación.

Por lo tanto esta característica del software incluido para cada cámara web debe ser deshabilitado para su uso con este proyecto.

Elaboración de manuales.

La documentación de este proyecto consiste en:

- Reporte de Proyecto Terminal.
- Manual de usuario.
- Código documentado.

Referencias consultadas:

R. C. Gonzalez y R. E. Woods, "Morphological image processing," en *Digital Image Processing*, 3ª ed. New Jersey: Pearson Education, 2008, pp 627-688.

R. C. Gonzalez y R. E. Woods, "Image segmentation," en *Digital Image Processing*, 3ª ed. New Jersey: Pearson Education, 2008, pp 689-794.

R. C. Gonzalez y R. E. Woods, "Representation and Description," en *Digital Image Processing*, 2ª ed. New Jersey: Pearson Education, 2002, pp 643-692.

C. Manresa, *et al*. (2011, marzo 2). *Hand Tracking and Gesture Recognition for Human-Computer Interaction* [En linea]. Disponible: http://elcvia.cvc.uab.es/index.php/elcvia/article/view/109/106

A. Kirillov, *et al. (2011 agosto 25). Aforge.net* [En linea]. Disponible: http://www.aforgenet.com/

Emgu CV, a cross plataform .Net wrapper to the OpenCV, "*image processing library"*, mayo 2012 [En linea]. Disponible: http://file.emgu.com/wiki/index.php/Main_Page