Sistema de cálculo de la constante del medidor electromecánico de energía eléctrica para la prueba de exactitud de la Compañía Energética de Occidente

Kevin Jhordy Palomino López

Director: Mag. Elena Muñoz España

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería en Automática Industrial
Popayán, 2017

Sistema de cálculo de la constante del medidor electromecánico de energía eléctrica para la prueba de exactitud de la Compañía Energética de Occidente

Monografía presentada como requisito parcial para optar por el título de Ingeniero en Automática Industrial

Kevin Jhordy Palomino López

Director: Mg. Elena Muñoz España

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería en Automática Industrial
Popayán, 2017


POSTERIS LVMEN MORITVRIS EDAT
1827
UNIVERSIDAD DEL CAUCA

Nota de aceptación: _____

_____

_____

_____

_____

_____

Director: _____

Elena Muñoz España

_____

Firma del Jurado

_____

Firma del Jurado

Popayán, 2017

Agradecimiento muy especial a las personas que estuvieron durante este recorrido lleno de momentos especiales, pero sobre todo de mucho aprendizaje, un agradecimiento a la Mg. Elena Muñoz España por la confianza brindada para la realización de este proyecto y sobre todo un muy especial agradecimiento a mis padres y mi hermana por el amor y la paciencia brindada durante esta bonita etapa.

Resumen

En este proyecto se propone un algoritmo capaz de realizar la prueba de exactitud, en los contadores electromecánicos, realizada por la compañía energética de occidente (ceo), esto se hace con el fin de automatizar esta misma ya que el modo en que se realiza actualmente es manual y propenso a fallas por parte, tanto del error humano como del error de las herramientas análogas que se usan. Lo novedoso de este algoritmo es que aparte de realizar dicha prueba, también se complementa con dos etapas más de procesamiento, la primera etapa cumple con la estabilización de imagen o el video que se esté captando, esta hace frente al problema de manos temblorosas que pueda presentar el operario. La otra etapa contrarresta los cambios de iluminación que se puedan presentar ya que la prueba se realiza en el lugar en el que está instalado el dispositivo, por ende, se trata de un ambiente no controlado y los cambios en la iluminación se tornan perjudiciales porque pueden derivarse en falsos positivos.

Otro aspecto importante en esta propuesta es que el algoritmo va a ser implementado para una aplicación móvil, dando un escalamiento tecnológico en la empresa y una facilidad en el uso de la misma dado que en la época en la que nos encontramos un gran porcentaje de la población tiene acceso a estos dispositivos y entiende bien su uso.

La implementación del algoritmo se lleva a cabo en un IDE (entorno de desarrollo integrado) conocido como “Android Studio” en su versión de escritorio 2.3.3 y por supuesto, teniendo en cuenta el enfoque del algoritmo se requirió una librería enfocada al área de visión computacional, la librería elegida fue “OpenCV” en su versión 3.1.0 para sistema operativo móvil Android.

El desarrollo de esta aplicación permitió reducir errores en la ejecución de esta prueba, también se redujeron considerablemente los tiempos de ejecución por lo que la empresa podría abarcar más dispositivos en un mismo periodo de tiempo aumentando así su eficiencia.

Palabras clave: Visión computacional, estabilización de imagen, smartphone,

TABLA DE CONTENIDO

Lista de Figuras

Lista de Tablas

1. Introducción

1.1. Planteamiento del problema

La Compañía Energética de Occidente (CEO), es la empresa encargada de prestar el servicio de energía eléctrica a las zonas urbanas y rurales de la ciudad de Popayán y en todo el Departamento del Cauca. Al igual que cualquier empresa energética, una de sus principales preocupaciones es realizar correctamente la medición de la energía consumida por parte del usuario. Para efectuar una medición confiable, es necesario garantizar el funcionamiento adecuado del medidor, por lo cual se requiere que este tenga una exactitud en dicha medición dentro de un rango permitido. Para comprobar que se encuentra en dicho rango, la CEO realiza ensayos de rutina, los cuales consisten en una verificación en sitio de los medidores a los cuales se haya solicitado estás pruebas por una posible no conformidad del usuario o por alguna anomalía que detecte la compañía en sus registros.

La prueba de exactitud, consiste en conectar el medidor a un equipo de prueba y aplicar energía al contador, en fase alta, fase media o en fase baja, después se contabiliza el tiempo que se demora en realizar ciertas revoluciones y se procede al cálculo del error relativo, si este no sobrepasa el ± 5% se puede decir que está correctamente calibrado, si este error supera el error máximo permitido la compañía procede a trasladar el artefacto al laboratorio.

Los instrumentos utilizados para realizar las pruebas a los medidores, son robustos y de tecnología antigua, actualmente usan un TGO tipo *panel hand*, dispositivo parecido a un *Smartphone*, pero mucho más robusto a los golpes. Este equipo es el corazón de las pruebas, ya que paso a paso va indicando los procedimientos que deben realizarse, además de encargarse de los cálculos y resultados de las pruebas con los valores que son ingresados por los operadores. El *panel hand* TGOes un equipo que está enfocado en la robustez a los golpes, pero en el procesamiento se queda muy corto, ya que deja de funcionar por intervalos de tiempo en las pruebas, haciendo que estas tomen mucho más tiempo. Además, requiere mucha inversión económica para su mantenimiento y los operadores deben ingresar a la TGO el conteo de las revoluciones, el cual es realizado de forma manual, y por lo tanto el cálculo del error relativo depende de la agilidad visual de la persona encargada.

Así surge la necesidad de realizar la transferencia de tecnología a dispositivos móviles con mayor procesamiento y menos costo de mantenimiento o reemplazo

para mejorar el desarrollo de las pruebas y lograr que un sistema de forma automática realice el cálculo de la constante del error relativo.

Lograr una buena medición de las revoluciones del medidor que permitan calcular la constante del mismo y así el error relativo, utilizando aplicaciones de visión artificial es un desafío interesante, ya que es necesario adecuarse a condiciones ambientales diversas como cambios de iluminación, ángulo de medición, movimiento involuntario de la cámara en el proceso de medición, entre otros.

A partir de la problemática planteada, se propone el siguiente interrogante ¿Cómo contrarrestar los efectos de variables ambientales no controladas, tales como, iluminación, ángulo de medición y movimiento de la cámara, para llegar a una buena medición de las revoluciones de los medidores electromecánicos usando algoritmos de visión por computadora?

1.2. Estado del arte

En la actualidad la visión artificial tiene un alto impacto dentro de la mayoría de los procesos industriales, una de sus principales aplicaciones es la medición de variables físicas, ya que con este avance se lograminimizar los errores quepuede cometer un operario que se encargue de esta labor con sus capacidades visuales.

En la medición de variables físicas mediante visión artificial, se requiere contrarrestar las condiciones que no se pueden controlar dentro del entorno real, entre ellas están, la inestabilidad del video capturado que deriva en problemas de vibraciones y de distorsión de ángulo, y el de las variaciones lumínicas que se puedan presentar en el transcurso de la prueba, ya que de no ser tratadas correctamente producen falsos positivos al marcar una medición errónea y en el caso de la verificación realizada en la prueba de exactitud, conducir a una mala conclusión acerca del medidor en cuestión.

En la literatura revisada se encuentran varios trabajos académicos que proponen usar visión artificial para realizar mediciones en el área industrial y calibración de instrumentos, [1], [2], [3], [4], [5], [6], [7], [8], [9] estas aplicaciones no abordan uno de los principales problemas que se tiene cuando las condiciones de iluminación cambian, ya que están bajo condiciones lumínicas controladas.

En cuanto al problema de la estabilización del vídeo, se encuentran trabajos que abordan soluciones tipo hardware como en [10] y [11], éstas son soluciones destacables pero involucran un desarrollo hardware y aditamentos para realizar la medición, por lo cual en este proyecto se propone abordar una solución software.

La estimación del movimiento de la cámara es un paso esencial para el desarrollo de técnicas de estabilización de vídeo, mediante el cual se proporcionan efectos estables para el filtrado y compensación del movimiento [12]. Las técnicas de estabilización de vídeo se pueden dividir en dos categorías principales de acuerdo con su capacidad de estimar el movimiento de la cámara, enfoques basados en la intensidad y en características.

Los enfoques basados en intensidad, emplean las texturas de la imagen como vectores de movimiento para estimar una transformación afín global y luego reconstruir los marcos estables [13], [14]. Por otro lado, los enfoques basados en características [15], [16], localizan un conjunto pequeño de características confiables en marcos adyacentes para la estimación del movimiento de la cámara. Estos rasgos pueden obtenerse a partir de la transformación de características invariantes en escala (SIFT) [17],*Speeded Up Robust Features* (SURF) [18], seguidor de características de Lucas Kanade y Shi Tomasi (KLT) [19], entre otros.

Las técnicas de estabilización de vídeo han venido mejorando con el pasar de los años, ya que los usuarios de los dispositivos móviles son cada vez más exigentes en cuanto a las características de sus *Smartphones* y de otros dispositivos de mano como las cámaras de video, esta mejora se debe al constante desarrollo de distintos algoritmos que corrigen esta anomalía [12].

Existen soluciones para corregir el problema de inestabilidad en los videos, como se plantea en [20], este trabajo es aplicado a un problema en el reconocimiento óptico de caracteres (OCR), en él se propone estabilizar la imagen mediante la creación de un marcador, este se fija a un objeto, en este caso a la palabra que se encuentre más cercana al centro de la pantalla, este marcador se queda fijo a la palabra, después se usa una fuerza para devolver este marcador al centro de la pantalla cada que esta se mueva. Con esto se minimiza significativamente el tiempo necesario para responder al movimiento y también se corrige el problema de manos temblorosas.

Una vez se obtiene un vídeo estabilizado, la substracción de fondo es la técnica comúnmente empleada para la detección de objetos en movimiento en una escena [21], [22], [23], [24]. Estas técnicas principalmente constan de dos pasos, el primero consiste en construir un modelo que describa los objetos estáticos de la escena (*background*), el segundo paso es sustraer los objetos en movimiento a partir de la diferencia que se presenta en el marco de vídeo comparado con el modelo construido. La imagen de fondo debe ser actualizada regularmente para que se adapte a las diferentes condiciones de luminancia de la escena.

El enfoque propuesto por [21], consta de tres etapas de procesamiento de la imagen, en la primera etapa el objetivo es encontrar un conjunto de puntos que se

puedan clasificar como fondo de una manera fiable, es decir, teniendo la certeza de que los puntos seleccionados si hacen parte del fondo incluso en presencia de fuertes cambios de iluminación. Una vez obtenido este sub-conjunto de puntos, se elimina la distorsión fotométrica para ello en la segunda etapa el algoritmo alinea en tonalidad la trama actual y la imagen de fondo y en la tercera etapa se realiza una sustracción en pixeles entre la imagen resultante obtenida de la alineación de la tonalidad y la imagen de fondo original.

En [23] para realizar la sustracción de fondo se utiliza un método llamado modelo de libro de códigos, este utiliza un modelo de color cilíndrico que se desarrolla basado en la observación de que, bajo variaciones de iluminación, los pixeles están casi totalmente distribuidos en una forma cilíndrica y se aprovecha esto para modelar la distribución del valor del píxel. Los parámetros geométricos de este modelo de color son el centro, la distorsión del color y los límites de intensidad. Se utiliza un filtro de paso bajo para actualizar el centro y la distorsión de color. El límite se actualiza utilizando los criterios máximos. Los fondos aprendidos incorrectamente se eliminan utilizando una técnica de longitud máxima de ejecución. Esta técnica se desarrolla bajo la suposición de que una apariencia que desaparece durante mucho tiempo no se debe utilizar para el modelado de fondo.

En el área de la video vigilancia el trabajo [24] propone un algoritmo que se desarrolla en distintas secciones: la primera se encarga de realizar la diferenciación de figuras que pertenecen al primer plano y figuras pertenecientes al fondo, esto se hace por medio de una comparación entre el valor del pixel y el valor que toma la muestra más cercana del modelo de fondo que se está construyendo, la siguiente sección es la que corrige el ruido para eso se emplea una etapa de post-filtración, este proceso implícitamente implica una retroalimentación entre el modelado y los componentes de detección fondo y la última sección es abordada introduciendo un razonamiento temporal en el primer plano para la detección de eventos anómalos, como: movimiento de la cámara o la generación de falsos positivos, esto se aborda realizando una re inicialización del modelo.

En [25]se utilizan algoritmos de sustracción de fondo para el conteo y la identificación de vehículos en grandes autopistas. Este método de sustracción de fondo utiliza el modelo gaussiano mixto (*Gaussian mixture model*). Los primeros 100 marcos se utilizan para estimar o modelar el fondo inicial el cual se actualiza secuencialmente. El aporte interesante que realiza este trabajo es que propone algoritmos para contrarrestar los efectos de sombras y brillos causados comúnmente para condiciones no controladas.

También se encuentran trabajos que serían de mayor utilidad, ya que dentro de estos se utilizan o desarrollan algoritmos capaces de contrarrestar los efectos de

las variaciones lumínicas pero no aplicados a un área en específico sino en pro de aportar conocimiento acerca de este tema [26].

Trabajos académicos como [22] pueden ser de gran utilidad, ya que se realiza una sustracción de fondo y seguimiento a una variable, similar a lo que se planea en este trabajo, para la sustracción de fondo se usa inicialmente, el promedio de algunos fotogramas iniciales y se toman de referencia, para evitar la correlación espacial entre pixeles, después se usa un algoritmo de sustracción de fondo y un mecanismo de diferencia temporal para poder estimar el movimiento de la variable que se desea seguir.

De la literatura revisada, se concluye que se encuentran múltiples algoritmos para el trabajo de la sustracción de un área de interés en imágenes estáticas o en video, también se encuentran algoritmos capaces de realizar una buena estabilización del mismo pero no se aprecia que dentro de las áreas de desarrollo aplicadas se encuentre la medición de variables en instrumentos eléctricos por lo que se propone desarrollar un algoritmo para contrarrestar los cambios de iluminación y reducir los problemas de inestabilidad de video, que son parte del medio ambiente en condiciones no controladas, para poder realizar una buena medición de la variable que se desea seguir.

1.3. Objetivos

1.3.1. Objetivo General

Proponer un sistema para determinar la constante del error con la cual se puede calcular el error porcentual del medidor, en condiciones ambientales no controladas, para la prueba de exactitud realizada por las brigadas de la CEO.

1.3.2. Objetivos Específicos

- Diseñar un algoritmo de visión por computadora para obtener el número de revoluciones del rotor de medidores electromecánicos que permita determinar la constante del error.
- Implementar el algoritmo en una aplicación móvil que sea robusto ante condiciones ambientales no controladas (iluminación, ángulo de lectura o vibraciones).
- Determinar la robustez de la aplicación ante condiciones ambientales no controladas.

1.4. Estructura del documento

Este trabajo se encuentra dividido en cinco capítulos, los cuales establecen una secuencia lógica que abarca desde la explicación del problema que se aborda, generalidades, implementación y validación del algoritmo, resultados de experimentos hasta el desarrollo de las conclusiones y trabajos futuros.

Capítulo I: En este capítulo se hace una introducción al problema que se va a desarrollar, se dejan claros los objetivos a cumplir al término del trabajo, y se exponen las ideas de otros autores sobre cómo se han abordado problemas similares.

Capítulo II: Se da una familiarización al lector sobre los conceptos y términos que se trabajan a lo largo del documento, esto para que pueda entender a cabalidad de qué se trata y como se desarrolla el trabajo.

Capítulo III: En esa etapa del documento se explica el algoritmo diseñado, se aborda detalladamente cada etapa, como se lleva a cabo su implementación y se facilita el pseudocódigo del algoritmo desarrollado.

Capítulo IV: En este capítulo se realizan las pruebas correspondientes al algoritmo desarrollado, se presentan y se analizan los resultados obtenidos.

Capítulo V: Finalmente en este capítulo se concluirá acerca de la calidad del algoritmo y su factibilidad en el uso por parte de la CEO, también se hablarán sobre posibilidades en trabajos en esta línea, tanto de desarrollo como de investigación.

2. Conceptualización

En este capítulo se investigan, se analizan y se definen conceptos y términos que se encuentran a lo largo del desarrollo del trabajo, también se explican conceptos y procedimientos que están en la normativa de la CEO. Esto se hace con el fin de familiarizar al lector con el problema que se está tratando y la solución que se le dará a dicho problema.

2.1. Medidor de energía eléctrica

Un medidor de Energía Eléctrica es el conjunto de elementos electromecánicos o electrónicos que se utilizan para el registro del consumo de energía eléctrica, tanto activa como reactiva, y en algunos casos su demanda máxima.

Existen varios tipos de medidores dependiendo de la construcción, tipo de energía que miden, clase de precisión y conexión a la red eléctrica, aunque existen instalados en terreno medidores electromecánicos. [27]

2.1.1. De acuerdo a su construcción:

- **Medidor de Inducción o Electromagnético.** Es un medidor en el cual las corrientes en las bobinas fijas reaccionan con las inducidas en el elemento móvil o disco, haciéndolo mover. El principio de funcionamiento es muy similar al de los motores de inducción y se basa en la teoría de la relación de corriente eléctrica con los campos magnéticos.

- **Medidores Estáticos o Electrónicos.** Medidores en los cuales la corriente y la tensión actúan sobre elementos de estado sólido (electrónicos) para producir pulsos de salida y cuya frecuencia es proporcional a los Vatios-hora o Var-hora. Están construidos con dispositivos electrónicos y son de mayor precisión que los electromagnéticos.

2.1.2. De acuerdo a la energía que miden:

- **Medidores de Energía Activa.** Miden el consumo de energía activa en kilovatios – hora.
- **Medidores de Energía Reactiva. M**iden el consumo de energía reactiva en kilovares – hora.

2.1.3. De acuerdo a la exactitud

De acuerdo a la Norma NTC 2288 y 2148, los medidores se dividen en las siguientes clases: 2, 1, 0.5 y 0.2 Estos valores significan los límites de error porcentual admisible para todos los valores de corriente entre el 10% nominal y la corriente máxima con un factor de potencia igual a 1.

- **Medidores clase 2** Se incluye medidores monofásicos, bifásicos para medir energía activa en casas oficinas, locales comerciales y pequeñas industrias con cargas inferiores a 45 kVA.
- **Medidores clase 1** Incluye los medidores trifásicos para medir energía activa y reactiva de grandes consumidores. Para cargas mayores a 45 kVA, se exige que sean medidores electrónicos.
- **Medidores clase 0.5** Se utilizan para medir a grandes consumidores. Cuando el usuario es no regulado o la tarifa es horaria, el medidor debe tener un puerto de comunicación o modem para enviar la información a través de la línea telefónica.
- **Medidores clase 0.2** Se utilizan para medir la energía activa suministrada en bloque en punto de frontera con otras empresas electrificadoras o grandes consumidores alimentados a 115 kV.

2.1.4. De acuerdo con la conexión a la Red

- **Monofásico bifilar.** Se utiliza para el registro de consumo en una acometida que tenga un solo conductor activo o fase y un conductor no activo o neutro. Es el medidor de uso más frecuente en las instalaciones residenciales. Está compuesto por una bobina de tensión y una de corriente. Su capacidad normalmente es de entre 15 y 60 A.
- **Monofásico trifilar.** Se utiliza para el registro del consumo de una acometida monofásica de fase partida (120/240 V) donde se tienen dos conductores activos y uno no activo o neutro.
- **Medidor bifásico trifilar.** Se utiliza para el registro del consumo de energía de una acometida en B.T de dos fases y tres hilos, alimentadas de la red de B.T de distribución trifásica tetrafilar. Se usa para medir la energía consumida por aparatos que requieran para su funcionamiento dos fases a 220 voltios, como por ejemplo motores de menos de 10 HP o aires acondicionados hasta 12000 BTU/H.
- **Medidor trifásico trifilar**.Se utiliza para registrar el consumo de energía de una cometida trifásica de tres fases sinneutro.

- **Medidor trifásico tetrafilar**. Se utiliza para registrar el consumo de una acometida trifásica en B.T. de tres fases y cuatro hilos. Se utiliza para medir la energía consumida por aparatos que requieran funcionar con tres fases a 220 voltios, como por ejemplo motores de más de 10 HP. Están compuestos por tres (3) bobinas de tensión y tres (3) bobinas de corriente.

El medidor que se considera para realizar este proyecto es un ISKRA modelo E82C2-02, este pertenece al grupo de los medidores monofásicos bifilares. Sus datos completos se encuentran en la Tabla 1.

MARCA	MODELO	CLASE	$F_n(V)$	$V_r(V)$	I		CERTIFICAD CIDET
					$I_b\,(A)$	$I_b(A)$	
ISKRA	E8C2C2	2.0	60	120	15	60	000399

Tabla 1: Datos del medidor ISKRA E8C2-02 (datos obtenidos de la información suministrada por la CEO)

CIDET (Centro de investigación y desarrollo tecnológico del sector eléctrico) es la corporación que se encarga de aprobar cuales son los medidores, entre otros dispositivos electrónicos, que pueden usar las compañías prestadoras del servicio de electricidad en Colombia.

En la Figura 1 se muestra el medidor descrito anteriormente.

Figura 1: Medidor de energía eléctrica ISKRA E8C2C2 (imagen realizada por el autor)

2.2. Verificación de medidores de energía eléctrica

La verificación en sitio es una tarea que deben realizar todas aquellas empresas que presten el servicio de suministro eléctrico y tengan elementos de medida con la necesidad de cuantificar el consumo de sus clientes o usuarios [28]. Esta aporta evidencia objetiva de que un elemento satisface los requisitos especificados tanto por el fabricante, por la misma empresa y por el cliente de la empresa prestadora del servicio. Para realizar las inspecciones a los equipos de medida existen varias pruebas, tales como: La prueba de consumo instantáneo en el medidor, la prueba de corrientes en la acometida, la prueba de relación de corriente, la prueba de relación de potencial, la prueba tiempo potencia, la prueba de exactitud y la prueba de integración o dosificación.

Las verificaciones que realiza la CEO se dividen en:

- **Verificación en laboratorio**: es aquella que se realiza en las instalaciones de CEO con equipos especializados y ambientes controlados, esta a su vez se divide en dos verificaciones:
    - **Verificación inicial:** se realiza a un medidor antes de ser instalado en una red eléctrica (medidores nuevos).
    - **Verificación posterior:** se realiza a un medidor retirado de su sitio de instalación para comprobar el estado en el cual venía funcionando (medidores usados).
- **Verificación en sitio:** es aquella que se realiza a un medidor en su sitio de instalación, con el fin de validar sus condiciones de funcionamiento físico y metrológico[28], al igual que en la verificación en laboratorio anteriormente descrita se divide en dos.
    - **Verificación inicial**: se realiza luego de 6 meses de usos para los medidores electromecánicos y 12 meses para medidores estáticos (electrónicos).
    - **Verificación posterior**: 2 años para medidores de clase 0.2s, 4 años para medidores de clase 0.5s y 10 años para medidores de clase 1, 2 y 3.

La prueba a la que está dirigido este proyecto se encuentra dentro de la verificación en sitio, es por esa razón que se van a concentrar esfuerzos en dicha sección.

2.2.1. Verificación en sitio:

Las empresas que brindan el servicio de suministro eléctrico se deben regir por normas y estándares que garanticen la veracidad de este proceso. Las normas

más relevantes en este caso son la norma técnica colombiana NTC 5900 “Verificación en sitio de equipos para medición de energía eléctrica” y la norma técnica colombiana NTC 4856 “Verificación inicial y posterior de medidores de energía eléctrica” [28], [29]. Estas verificaciones se dan entre otras cosas cuando los clientes o la empresa sospechan del incorrecto funcionamiento de uno de sus medidores es entonces cuando se realizan las pruebas técnicas de verificación en sitio. Cuando los clientes realizan una PQR (Peticiones, quejas y reclamos) a la CEO sobre el mal funcionamiento del medidor, las brigadas técnicas que verifican el medidor se llaman Brigadas PQR. Cuando la CEO a través de variables estadísticas determina que un medidor no funciona correctamente (perdidas de energía), las brigadas técnicas que verifican estos medidores se llaman Brigadas MACRO, las cuales son programadas en los sitios donde se sospecha que hay pérdidas de energía. Cuando se va a atender un PQR o están en campañas MACRO, la compañía envía una brigada técnica conformada por dos operarios y un supervisor. Las pruebas técnicas que se realizan en la verificación en sitio se pueden ver en la Figura 2.

Figura 2: Actividades realizadas en la verificación en sitio (imagen tomada de la documentación de CEO)

A continuación, se listan los instrumentos usados en la verificación en sitio para las diferentes pruebas.

	
Analizador verificador de medidores (AVM): el equipo principal, el cual contiene todos los módulos de las pruebas a realizar y al cual se conectan los otros elementos.	**Carga fantasma: este equipo** simula un consumo controlado de alta, media y baja tensión para las pruebas de exactitud y dosificación.
	
TGO: este elemento se conecta vía Bluetooth con el AVM y le provee una interfaz gráfica de las funcionalidades que este trae.	**Multímetro:** Instrumento electrónico portátil para medir directamente magnitudes eléctricas activas dentro de la prueba.
	
Emisor: se usa para emitir pulsos digitales que son interpretados por el dispositivo AVM	**Cámara digital:** se usa para tomar las fotos en la prueba de dosificacion y para llevar el registro fotográfico del proceso.

	
Sensor de pulsos: se usa para contar los pusos que emite un led del contador en la prueba de exactitud. La velocidad de intermitencia es según el tipo de contador.	**Impresora portatil:** se usa para imprimir los comprobantes del proceso tanto para la CEO, la Unión de Trabajadores de la Industria Energética Nacional (UTEN) y el usuario.

Tabla 2: Instrumentación usada en las pruebas de verificación en sitio (información e imágenes tomadas desde la documentación de CEO)

2.3. Prueba de exactitud:

Según la norma NTC 5900, esta prueba consiste en verificar el error de medida (exactitud) del medidor en el punto de prueba (sitio de instalación), con un valor de corriente y voltaje elegidos por el operario dependiendo de qué fase desea aplicar, puede ser baja, media o alta. Usualmente se elige la fase baja, esta prueba tiene como resultado un error relativo que es expresado como porcentaje, y puede ser calculado por la comparación entre el medidor objeto de verificación y un objeto patrón, la prueba debe realizarse bajo las siguientes condiciones de referencia:

- Los rangos de tensión permitidos para la evaluación del equipo de medida deben encontrarse entre el -20% y el +15% de la tensión nominal o rango de tensión indicada en las características del medidor.
- Los rangos de corriente para la realización de las pruebas deben ubicarse entre el 10% de la corriente básica y la corriente máxima según lo especifique la placa de características.

El resultado de la prueba es satisfactorio cuando el error registrado en el punto de prueba se encuentra entre el -5% y el +5%, como lo ilustra la Figura 3.

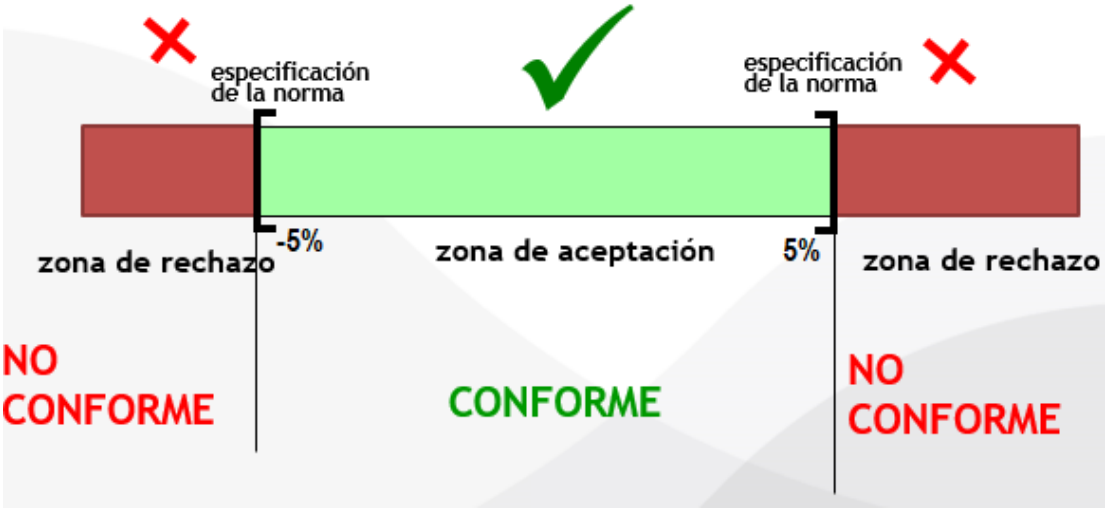


Figura 3: Zona de aceptación y zona de rechazo establecidas para el medidor (imagen realizada por el autor)

Para la realización de esta prueba se debe establecer la conexión que se muestra en la Figura 4.

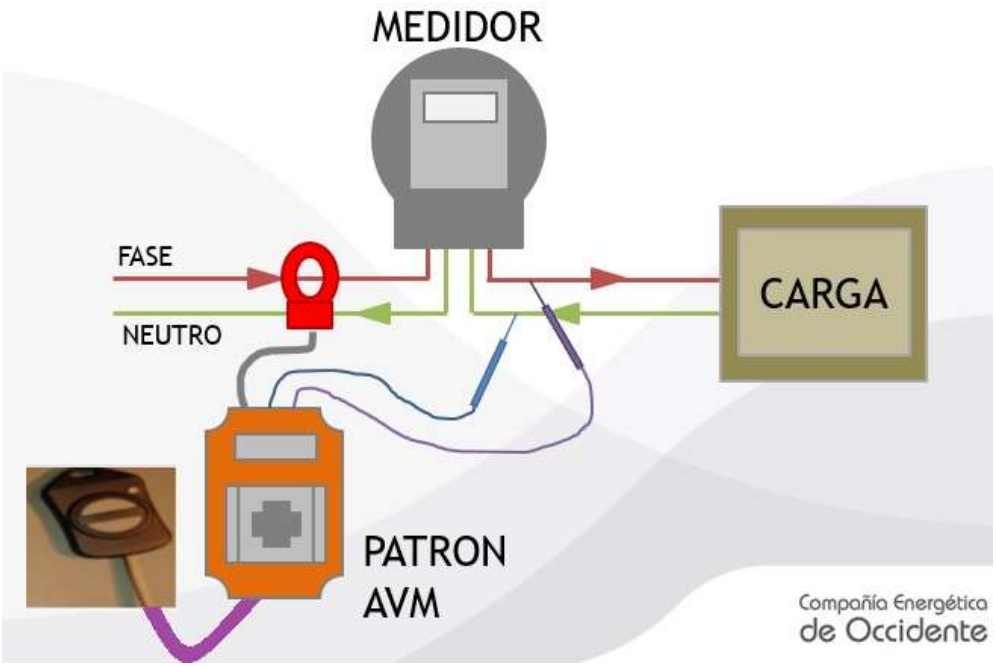


Figura 4: Conexión requerida para la realización de las pruebas (imagen tomada desde la documentación de CEO)

Una vez realizada la conexión se induce una carga fantasma con el equipo que se presenta en la Tabla 3, inmediatamente el medidor entra en funcionamiento y el

operario deberá presionar el botón del artefacto llamado emisor cada vez que la marca del disco giratorio pase por la marca del frente del medidor, esto se realiza para siete revoluciones, enseguida se calcula internamente el error de medición que es el que indica si el medidor es apto para seguir funcionando o no.

La aplicación que usa actualmente la CEO para la prueba de exactitud está contenida en la TGO (dispositivo presentado en la Tabla 3), dicha aplicación es llamada SYSMOVIL, este dispositivo cuenta con un sistema operativo Windows y conexión Bluetooth para su comunicación con el AVM, parte de su interfaz puede ser observaba en la Figura 5, en esta figura también se observan los datos necesarios para realizar la prueba de exactitud, como únicas incógnitas se tienen el número de pulsos (o revoluciones) y el tiempo que se toma en realizar la prueba.

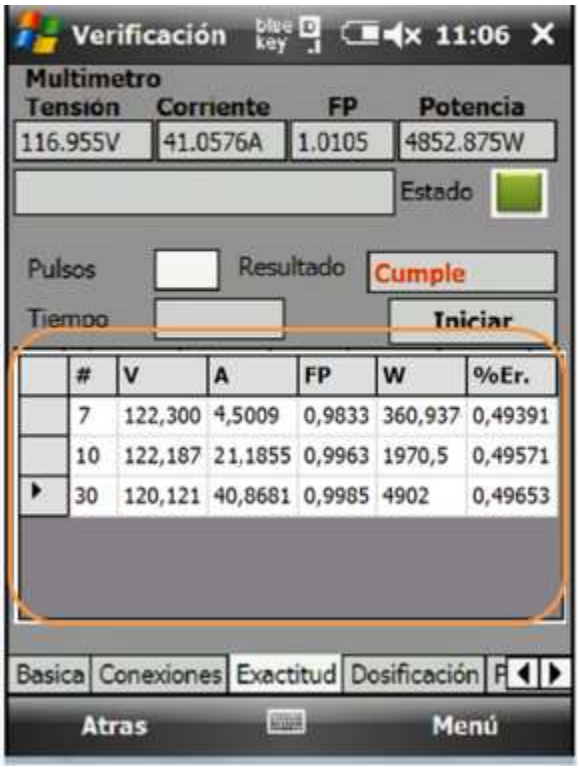


Figura 5: Interfaz de la aplicación SYSMOVIL (imagen tomada desde la documentación de CEO)

El error porcentual del medidor, calculado está definido por la ecuación (1).

$$e_{pe} = \frac{N_e - N_r}{N_r} x100 \quad (1)$$

En donde:

N_r : es el número de impulsos registrados por el dispositivo contador durante la calibración

N_e : es el número de impulsos del patrón esperados durante el periodo de calibración, este valor puede ser calculado por la ecuación (2).

$$N_e = \frac{K_d\ I\ U\ T}{2700000} \quad (2)$$

En donde:

K_d : Constante del medidor en $\frac{imp}{kWh}$.

I : Corriente en amperios.

U : Tensión en voltios.

T : Tiempo de la duración de la prueba.

2.4. Visión de máquina

La visión de máquina, visión artificial o visión por computador es una disciplina dentro de la rama de la inteligencia artificial, esta se encarga de extraer información del mundo que nos rodea a partir de imágenes y su interpretación utilizando un computador.

En otras palabras, la visión de máquina hace referencia a la extracción automática, análisis y compresión de la información valiosa a partir de una imagen o una secuencia de imágenes, para esto se han desarrollado bases teóricas y algoritmos que ayudan en la comprensión visual automática.

2.4.1. Visión humana vs visión artificial:

El ojo tiene una forma, aproximadamente, esférica de unos 2.5 cm de diámetro y está formado por una óptica y una zona sensorial. La óptica está constituida por la córnea, el iris o pupila y el cristalino[30]. La córnea es un material transparente y funciona como lente fijo. La pupila regula la cantidad de luz que entra en el interior y el cristalino hace las veces de lente variable, permitiendo el enfoque dependiendo de la distancia de los objetos.

El ser humano, ha querido llevar la naturaleza de sus sentidos a artefactos tecnológicos, es el caso de las cámaras de video que se puede encontrar en múltiples dispositivos incluyendo los *Smartphones,* estos dispositivos con sus

ópticas, hacen las veces del globo ocular, mientras el computador realizará las tareas de procesamiento, emulando el comportamiento del cerebro.

Incluso con los avances a los que hoy en día ha llegado la humanidad no se conoce el mecanismo que el cerebro usa para obtener la información de la percepción, tal como el trabajo [31]donde el cerebro es capaz de determinar la distancia de los objetos, de reconocerlos en diferentes posiciones no importa si estos se encuentran rotados y con información parcialmente oculta. Es decir que en nuestro cerebro se encuentran inmersos incontables algoritmos que el ser humano no ha podido desarrollar para que estos sean implementados en tecnología artificial.

Lo que si se ha logrado con visión artificial es construir nuevos algoritmos que sean capaces de obtener información de bajo nivel visual. Y aunque todavía se esté años luz de la percepción visual de los seres vivos, la visión artificial es robusta en tareas visuales repetitivas y alienantes para el hombre. Por ejemplo, en el campo de la inspección de productos en la industria o en contar células en una imagen de microscopía o en determinar la trayectoria de un vehículo en una autopista, entro otras.

2.4.2. Procesamiento digital de imágenes

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información [32].

Un paradigma comúnmente usado en el procesamiento de imágenes es el proceso de filtrado, dicho procedimiento consiste en la aplicación de una “matriz de filtrado” de tamaño $N \times N$ compuesta por números enteros (generalmente de 3x3 como se enseña en la Tabla 3, aunque puede ser mayor) a cada uno de los píxeles de la imagen, lo que genera que el valor numérico para cada pixel se vea modificado a uno nuevo en función del valor original. El resultado final se divide entre un escalar, por lo general este escalar es el resultado de la suma de los coeficientes de ponderación. Los filtros se pueden expresar mediante la ecuación (3).

$$ND'_{i,j} = \frac{ND_{i-1,j-1} + ND_{i,j-1} + ND_{i+1,j-1} + ND_{i-1,j} + ND_{i,j} + ND_{i+1,j} + ND_{i-1,j+1} + ND_{i-1,j+1} + ND_{i-1,j+1}}{9} \quad (3)$$

Donde i y j representan la fila y la columna de cada pixel, $ND_{i,j}$ su “Nivel Digital” y $ND'_{i,j}$ el “Nivel Digital” obtenido tras hacer el filtrado.

1	1	1	
1	1	1	Div = 9
1	1	1	

Tabla 3: Filtro de la media

En ambos casos, tanto en la ecuación como en la tabla se está representando un filtro que se denomina filtro de la media o filtro promediado, y mediante combinaciones de parámetros asignados a los diferentes pixeles circundantes se pueden conseguir diversos efectos.

En [33] este proceso (filtrado de imágenes digitales) se define como el conjunto de técnicas englobadas dentro del pre-procesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella. Los principales objetivos que se persiguen con la aplicación de filtros son:

- Suavizar la imagen. Reducir la cantidad de variaciones de intensidad entre pixeles vecinos.

- Eliminar ruido. Eliminar aquellos pixeles cuyo nivel de intensidad es diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
- Realzar bordes. Destacar los bordes que se localizan en una imagen.

- Detectar bordes. Detectar los pixeles donde se produce un cambio brusco en la función intensidad. Por tanto, se consideran los filtros como operaciones que se aplican a los pixeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella.

  En la Figura 6 se tiene un filtro de la mediana, con el que se consigue un efecto de suavizado, este filtro es comunmente usado cuando la imagen presenta ruido.

Figura 6: Filtro de la media (imagen realizada por el autor)

Mientras que a la Figura 7 se le aplica el filtro de Sobel, técnicamente es un operador diferencial discreto [34] que calcula una aproximación al gradiente de la función de intensidad de una imagen. Para cada punto de la imagen a procesar, el resultado del operador Sobel es tanto el vector gradiente correspondiente como la norma de este vector. El operador Sobel detecta los bordes horizontales y verticales de manera separada y después realiza una composición de la imagen.

Figura 7: Filtro de Sobel (imagen realizada por el autor)

2.4.3. Estabilización de imagen

La estabilización de video o estabilización de imagen es aplicada a una secuencia de frames para corregir el llamado “shaking hands[1]” o problemas de manos temblorosas, desde los años 60 se ha intentado dar solución a esta falencia, esto se ha hecho por dos ramas [35]: sistemas de estabilización mecánica y sistemas de estabilización digital, dentro de la primera rama destacan un “estabilizador de imagen mecánico ubicado en la óptica”, este es un dispositivo que se sitúa aparte del cuerpo de la cámara, lo que hace básicamente es detectar el movimiento de la cámara con giroscopios o acelerómetros y este cuerpo genera un movimiento en el sistema de lentes de estabilización, por lo que dicho movimiento se ve compensado antes de que la imagen sea procesada, el otro dispositivo destacado perteneciente a esa rama es el “Estabilizador de imagen mecánico por desplazamiento del sensor de imagen” que tiene los mismos principios del método anterior pero los movimientos de compensación generados se aplican al sensor de la imagen. En la otra rama, y de mayor importancia para este trabajo, ya que las compensaciones de movimiento se hacen mediante software, destacan dos, estabilización de video mediante “Speeded Up Robust Features”(SURF), y estabilización de video mediante “Scale-invariant feature transform” (SIFT).

Los anteriores algoritmos tienen en común que son usados principalmente para la extracción de características relevantes de las imágenes, estas pueden usarse posteriormente en reconocimiento de objetos, detección de movimiento, estereopsis[2] [36] o como es el caso del presente trabajo en estabilización de imagen, los trabajos de estabilización imagen usualmente están conformados por tres etapas, estimación de movimiento, suavizado de movimiento y complementación del video final, esto busca eliminar los movimientos visualmente desagradables, el primer paso para lograrlo, con la ayuda de cualquiera de estos dos algoritmos, es extraer las características de tramas consecutivas, y realizar una estimación de movimiento que se pueden llevar a cabo con distintos algoritmos como lo son: Algoritmo de concordancia de bloques, Correlación de fase y métodos de dominio de frecuencia, algoritmos recursivos de píxeles, flujo óptico, etc. Lo que le sigue a este paso es la suavización del movimiento, para ello existen algoritmos como: suavizado aditivo, filtro de Butterworth, suavizado exponencial, filtro de Kalman, Suavizado laplaciano, etc. Y por último paso se

[1]Shakinghands: situación en el sujeto en cuestión, involuntariamente, presenta temblores en las manos, esto es causado por problemas en áreas del cerebro que controlan el movimiento.
[2]Estereopsis: es el fenómeno dentro de la percepción visual por el cual, a partir de dos imágenes ligeramente diferentes del mundo físico proyectadas en la retina de cada ojo, el cerebro es capaz de recomponer una tridimensional.

tiene la complementación de video, en esta parte final se deben reconstruir los frames ya procesados, esto se puede lograr, al igual que en los anteriores pasos, con algoritmos ya propuestos, como es el caso de [36] que propone un algoritmo basado en las síntesis de texturas, también existen otros como: algoritmo basado en vector de movimiento, algoritmo basado en ecuaciones diferenciales parciales.

La extracción de características con ayuda del algoritmo **SIFT** fue un trabajo propuesto por David Lowe en 1991 pero fue hasta 2004 cuando lo patentó y lo publicó en su trabajo[37] llamado "Object recognition from local scale-invariant features," este se lleva a cabo de la siguiente manera: como primer paso se detectan los extremos o esquinas en la escala espacio-espacio, para lograr esto se aplica la diferencia gaussiana en tramas o frames consecutivos a lo largo del espacio determinado por (x, y) (coordenadas) esto da una fuerte respuesta de puntos positivos para ser clasificados como esquinas, por tanto en esta etapa se compara cada uno de esos puntos con los de las tramas vecinas y se crean coincidencias, en la siguiente etapa de localizan estos puntos a precisión para esto se “refina” cada sub-píxel usando la expansión de la Serie de Taylor, si el valor de la esquina es menor que cierto valor de umbral el punto se descarta, como la diferencia gaussiana también detecta bordes estos son eliminados a través de la “matriz hessiana”, el siguiente paso es asignar una orientación a cada punto de interés, para garantizar la invarianza respecto a la rotación de las imágenes para realizar este trabajo se toman los puntos vecinos en torno a cada punto de interés y se calcula la magnitud y la dirección del gradiente, entonces se hace un histograma de dichas direcciones ponderado por la magnitud del gradiente, y el mayor pico en el histograma indica la orientación del punto de interés. Como paso siguiente se crean los descriptores de cada punto de interés, para cada punto se toma un vecindario 16x16 puntos. Este, a su vez, se divide en sub-bloques de 4x4, para cada uno se crea un histograma de orientaciones. La concatenación en un vector de los valores de las cajas de cada histograma para los 16 sub-bloques del punto de interés constituye su descriptor y finalmente se realiza la correspondencia de los puntos de interés, La correspondencia entre los puntos de interés de dos imágenes se obtiene a través de una búsqueda del punto más próximo en el espacio de los descriptores de puntos de interés. Pero se encuentran casos en el que el segundo punto más próximo puede estar muy cerca del primero por culpa del ruido. Esto se corrige calculando la razón entre la distancia al más cercano y al segundo más cercano y si ésta está por encima de cierto umbral, los puntos son descartados.

Por otro lado, se tiene el algoritmo **SURF** este algoritmo es capaz de entrenar un sistema para que interprete imágenes y determine su contenido. El Algoritmo SURF se presentó por primera vez por Herbert Bayen el año 2006 y publicado en el trabajo [38], SURF es un detector y un descriptor de alto rendimiento de los

puntos de interés de una imagen, donde se transforma la imagen en coordenadas, utilizando una técnica llamada multi-resolución. Consiste en hacer una réplica de la imagen original de forma Piramidal Gaussiana o Piramidal Laplaciana, y obtener imágenes del mismo tamaño, pero con el ancho de banda reducido, cabe aclarar que el algoritmo SURF está basado en su predecesor SIFT por tanto la explicación va a ser menos detallada que en el anterior caso, SURF está basado en los mismos principios y pasos que el SIFT, pero utiliza un esquema diferente y esto provee mejores resultados en cuanto rapidez. Con el fin de detectar puntos característicos o punto de interés en una escala de manera invariable SIFT utiliza filtros de aproximación en cascada (filtros Haar). Donde la Diferencia de Gaussianas, DoG, se calcula sobre imágenes re-escaladas progresivamente. Para la detección de las esquinas se usa un método Semejante al DoG, en vez de calcular gaussianas para promediar la imagen, se utiliza la imagen integral. SURF utiliza un detector de BLOB (Binary Large Object) basado en el Hessiano para encontrar puntos de interés. El determinante de la matriz Hessiana expresa un cambio local alrededor del punto de interés. El detector se basa en la matriz Hessiana, debido a su buen desempeño en la precisión. Es decir, se detectan estructuras BLOB en lugares donde el factor determinante es el máximo. El siguiente paso es localizar el punto de interés y también representarlo a escala ya que los puntos de interés deben ser encontrados en diferentes escalas, entre otras cosas porque la búsqueda de correspondencias a menudo requiere su comparación en las imágenes donde se les ve a diferentes escalas. Los espacios escala se aplican en general como una pirámide de imagen. Las imágenes se suavizan repetidamente con un filtro gaussiano y luego, se sub-muestrea a fin de conseguir un nivel superior de la pirámide. Lo siguiente es agregar el descriptor a cada punto, ese descriptor es generado basándose en el área circundante de cada punto (se obtiene un vector descriptor por cada uno). Este vector contiene valores como orientación, iluminación, rotación y como paso final se detectan las coincidencias, esto se hace buscando coincidencias de los puntos de interés en cuadros consecutivos, esto se hace comparando puntos característicos o de interés, junto a su respectivo descriptor en tramas consecutivas y determinar correspondencia.

Estos dos algoritmos explicados recientemente son muy efectivos para el fin de estabilizar una trama de imágenes, pero solo SURF tiene soporte en la librería que se va a usar, sin embargo, es un algoritmo “pesado” para ser implementado en una aplicación de tiempo real.

2.4.4. Detector de esquinas de Harris

Una esquina está definida como una región perteneciente a una imagen, con gran variación de intensidad en todas las direcciones. Se han propuesto múltiples algoritmos desde que se estudia el área de visión computacional, un intento antiguo, pero acertado para encontrar estos puntos característicos fue dado por Chris Harris y Mike Stephens en su artículo [39] "*A combined Corner and Edge Detector* " publicado en 1988, ahora este método es nombrado "Detector de esquinas de Harris". El llevó la idea de plantear en una forma matemática la diferencia en intensidad para un desplazamiento de (u, v)en todas las direcciones. Como se expresa en la ecuación (4):

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 \tag{4}$$

Donde $w(x,y)$ es la ventana en la posición(x,y) y w es la variable que asigna un peso a cada pixel, $I(x,y)$ es la intensidad en (x,y) y $I(x+u, y+v)$ es la intensidad con un desplazamiento determinado por $(x+u, y+v)$.

Lo que se busca son ventanas en las cuales exista una esquina, es decir que se buscan ventanas con gran variación de intensidad, para dar con estas grandes variaciones, los autores proponen maximizar solo la parte de desplazamiento ($[I(x+u, y+v) - I(x,y)]^2$) descrito en la ecuación (4)usando la expansión de Taylor, realizando las operaciones correctamente se obtiene la ecuación (5):

$$E(u,v) \approx \sum_{x,y} u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2 \tag{5}$$

La ecuación (5) también puede ser expresada en forma matricial como:

$$E(u,v) \approx \left(\sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \tag{6}$$

Ahora se denota:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \tag{7}$$

Reemplazando las ecuaciones (6) y (7) en la ecuación (5) se obtiene la ecuación(8):

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{8}$$

Los autores definieron la ecuación (9), la cual determinará si una ventana puede contener una esquina o no.

$$R = \det(M) - k(trace(M))^2 \quad (9)$$

Donde:

$$\det(M) = \lambda_1 \lambda_2 \quad (10)$$

$$trace(M) = \lambda_1 + \lambda_2 \quad (11)$$

$\boldsymbol{\lambda_1}$ $\boldsymbol{y}$ $\boldsymbol{\lambda_2}$ Son valores propios de M

Y los valores de estos auto-valores o valores propios son los que determinan si la región en cuestión es esquina, borde o plano de la siguiente manera:

- Cuando $|\boldsymbol{R}|$ es pequeña, (sucede cuando $\boldsymbol{\lambda_1}$ $\boldsymbol{y}$ $\boldsymbol{\lambda_2}$ son valores pequeños) la región es plana.
- Cuando $\boldsymbol{R} < 0$(sucede cuando $\boldsymbol{\lambda_1} \gg \boldsymbol{\lambda_2}$ o viceversa) la región es borde.
- Cuando $\boldsymbol{R}$ tiende a un valor muy grande (sucede cuando $\boldsymbol{\lambda_1}$ $\boldsymbol{y}$ $\boldsymbol{\lambda_2}$ son valores grandes) la región es una esquina.

2.4.5. Detector de esquinas Shi-Tomasi

El “detector de esquinas Shi-Tomasi” fue propuesto en 1994, por J. Shi y C. Tomasi y publicado en su artículo [40]“Good Features to Track”, este es una mejora del “detector de esquinas de Harris”, y el algoritmo muestra mejores resultados en comparación con su predecesor. La modificación que se plantea en este artículo es la de reemplazar la ecuación (9), se propone la ecuación (12):

$$R = \min(\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}) \quad (12)$$

En su artículo “Good Features to Track”, Shi y Tomasi demostraron experimentalmente que este criterio de puntuación era mucho mejor. Si R es mayor que un determinado valor predefinido, puede marcarse como una esquina. Para concluir cada punto o ventana se tendría que:

- Cuando $\boldsymbol{R}$ es mayor al valor predeterminado (sucede cuando el valor mínimo entre $\boldsymbol{\lambda_1}\ \boldsymbol{y}\ \boldsymbol{\lambda_2}$ superan dicho valor) la región es una esquina.
- Cuando $\boldsymbol{R}$ es pequeña, (sucede cuando $\boldsymbol{\lambda_1}\ \boldsymbol{y/o}\ \boldsymbol{\lambda_2}$ son menores que el valor determinado) la región puede calificarse como borde o como plana, se dice que la región es un borde cuando $\boldsymbol{\lambda_1}\ \boldsymbol{y}\ \boldsymbol{\lambda_2}$ son menores que el valor predeterminado y se dice que la región es plana cuando solo un valor propio ya sea el de $\boldsymbol{\lambda_1}\ \boldsymbol{o}\ \boldsymbol{\lambda_2}$ supera el valor predeterminado.

2.4.6. Transformada Afín o transformada rígida:

La transformada afín está incluida dentro de las transformaciones geométricas, estas (transformadas geométricas) modifican la relación espacial entre pixeles. En términos de procesamiento de imágenes digitales una transformación geométrica consiste en dos operaciones básicas [34]:

1. Una transformación espacial que define la reubicación de los píxeles en el plano de imagen.

2. Interpolación de los niveles de grises, los cuales tienen que ver con la asignación de los valores de intensidad de los píxeles de la imagen transformada.

La técnica de transformación afín se utiliza típicamente para corregir las distorsiones o deformaciones geométricas que se producen con ángulos de cámara no ideales. La siguiente tabla ilustra las diferentes transformaciones afines: translación, escalamiento, cizallamiento y rotación.

En la Tabla 4 se ilustran las diferentes transformaciones afines: translación, escalamiento, cizallamiento y rotación.

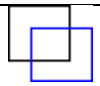
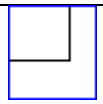
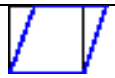
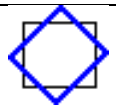
Transformación Afín	Ejemplo	Matriz de transformación	
Translación		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$\boldsymbol{t_x}$ especifica el desplazamiento en el eje x. $\boldsymbol{t_y}$ especifica el desplazamiento en el eje y.
Escalamiento		$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & t_y & 1 \end{bmatrix}$	$\boldsymbol{s_x}$ especifica el factor de escala a lo largo del eje x $\boldsymbol{s_y}$ especifica el factor de escala a lo largo del eje y
Cizallamiento		$\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\boldsymbol{sh_x}$ especifica el factor de cizallamiento a lo largo del eje x $\boldsymbol{sh_y}$ especifica el factor de cizallamiento a lo largo del eje y
Rotación		$\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\boldsymbol{q}$ especifica el ángulo de rotación

Tabla 4: transformaciones afines (tomada de [41])

2.4.7. Flujo óptico

El flujo óptico está definido como el patrón de movimiento aparente de los objetos, superficies y bordes en una escena, esto se debe al movimiento relativo entre un observador (un ojo o una cámara) y la escena. El concepto de flujo óptico se estudió por primera vez en la década de 1940 y, finalmente, fue publicado por el psicólogo estadounidense James J. Gibson como parte de su trabajo [42] titulado “A theory of affordance”.

El concepto de flujo óptico es ampliamente utilizado en temas relacionados con la visión artificial, abarca técnicas relacionadas desde el procesamiento de imágenes y el control de navegación incluyendo detección de movimiento, segmentación de objetos, información de tiempo de contacto, enfoque de cálculos de expansión, luminancia, codificación de movimiento compensado y medición de disparidad estéreo.

Otro autor citado en [43] como Helmholtz se preocupa principalmente por la percepción de profundidad y describe el flujo óptico como las "variaciones de la

imagen retiniana" que son debidas al movimiento del cuerpo y dependen de la estructura, a saber, la distancia, así como la rigidez, del medio ambiente. Gibson describe el desplazamiento de la estructura en la matriz óptica como una transformación que se convierte en "vivo con movimiento cuando el observador se mueve".

En un contexto bio-inspirado, el flujo retinal es el cambio de patrones estructurados de luz en la retina, lo que se puede interpretar como una impresión de movimiento de las imágenes visuales proyectadas sobre la retina. En la Figura 8.a se muestra la producción de flujo óptico de la retina para los desplazamientos de dos características visuales ejemplares. En términos técnicos y en el contexto de la visión por ordenador, los cambios del entorno en la imagen están representados por una serie de marcos de imagen. En la Figura 8.b muestra tres fotogramas de una secuencia de imágenes, que puede ser obtenido mediante un muestreo espacial. El flujo óptico capta el cambio en estas imágenes a través de un campo vectorial. La investigación hace hincapié en la estimación precisa y en píxeles del flujo óptico, que es una tarea exigente desde el punto de vista computacional.

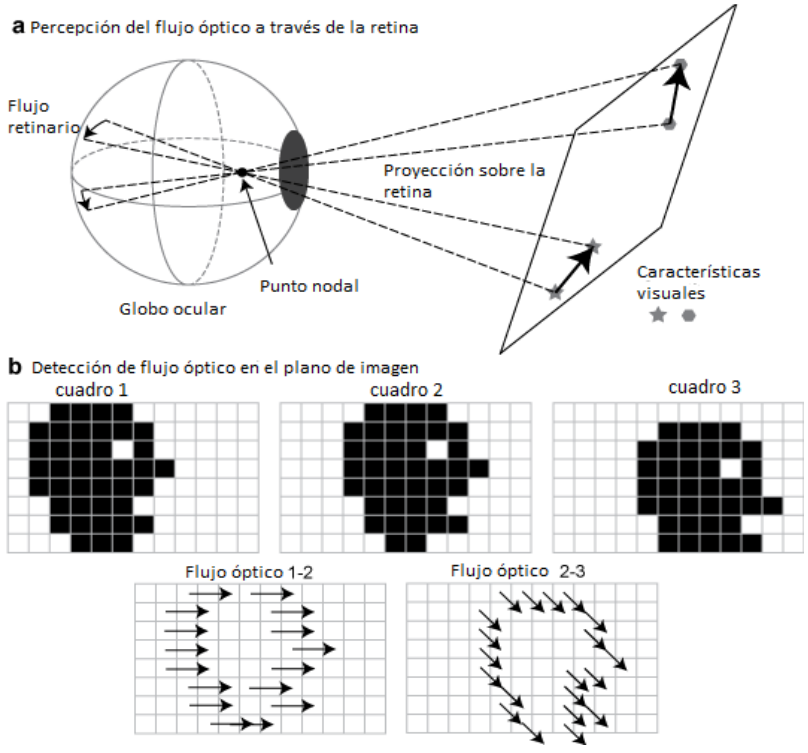


Figura 8: Flujo óptico (modificada de [43])

En la Figura 8a, el flujo óptico se genera en la retina por cambios en los patrones de luz. En la imagen se muestra el cambio de dos características visuales (estrella y hexágono) en un plano y sus desplazamientos angulares en la superficie del globo ocular, mientras que en la Figura 8b, Si la luz estructurada es muestreada espacialmente y temporalmente, se obtiene una secuencia de imágenes, en este ejemplo se muestran tres marcos, en los que se visualiza el movimiento de la silueta de una cabeza. El flujo óptico se representa como la correspondencia de los píxeles de contorno entre los fotogramas 1 y 2, así como los fotogramas 2 y 3. Para los métodos de estimación de flujo, el desafío es encontrar la correspondencia de puntos para cada píxel de la imagen, no sólo los píxeles de contorno.

2.4.8. Binarización de imágenes:

La binarización es una técnica de procesamiento de imágenes donde los únicos valores numéricos que puede tener cada píxel son 1 y 0 (Blanco y negro). Para determinar qué valor tendrá cada píxel, se necesita un valor numérico que se llama umbral. Los píxeles cuyos valores de intensidades (escala de grises) que se encuentren por debajo de este umbral tomaran un nuevo valor numérico de 0 (negro) y los valores de intensidades que estén por encima de este umbral tomaran valores de 1 (blanco)[44]. La ventaja de esta técnica es la de poder especificar el valor del umbral cuando estamos buscando objetos con colores específicos, de esta manera se pueden separar objetos o regiones de interés del fondo.

2.4.9. Operaciones morfológicas en imágenes:

Las operaciones o transformaciones morfológicas son un tipo de procesamiento digital de imágenes. Esta técnica se emplea comúnmente cuando las tareas de segmentación no son capaces de dar un resultado exacto de la delimitación de los objetos o regiones de interés [45]. Aparecen píxeles mal clasificados, bordes imprecisos de los objetos o regiones que están ocultas. Por tanto, antes de extraer más características de nivel medio se requiere de una etapa de post-procesamiento. En esta fase se suele emplear el tratamiento morfológico que en definición es una técnica de procesamiento no lineal de la señal, caracterizada en realzar la geometría y forma de los objetos [30]. Su fundamento matemático se basa en la teoría de conjuntos. En un principio estas operaciones solo eran aplicables sobre imágenes binarizadas, luego se extendieron a las imágenes en niveles de grises. Este uso a niveles de grises permitió vislumbrar que el

procesamiento morfológico también se puede utilizar como técnica de procesado de la señal. Concluyendo, estas transformaciones se pueden emplear tanto en el procesado, como en las etapas de segmentación post-procesado o en fases de mayor nivel de información visual. Actualmente se puede encontrar aplicaciones en la restauración de imágenes, en la detección de bordes, en el análisis de texturas, en el aumento del contraste y hasta en la compresión de imágenes.

Existen operaciones morfológicas bases, pero antes de hablar de ello se deben tener claros los siguientes conceptos:

- Inclusión: Y es subconjunto de X si todos los elementos de Y pertenecen a X:

$$Y \subseteq X \Leftrightarrow (p \in Y \implies p \in X) \tag{13}$$

  La inclusión es reflexiva $(X \subseteq X)$, antisimétrica $(Y \subseteq X \; y \; X \subseteq Y \implies X = Y)$ y transitiva$(Y \subseteq X \; y \; X \subseteq Z \implies Y = Z)$. Un conjunto que cumpla estas tres condiciones se dice que es un conjunto totalmente ordenado.

- Intersección: La intersección de dos conjuntos X e Y es el conjunto de los elementos que pertenecen a ambos conjuntos:

$$X \cap Y = (p | p \in X \;\; y \;\; p \in Y) \tag{14}$$

  La intersección es conmutativa, asociativa e idempotente[3]. Esta última propiedad es importante en Morfología e indica que $X \cap X$.

- Unión: La unión de dos conjuntos se constituye por los elementos que pertenecen a uno o al otro:

$$X \cup Y = (p | p \in X \;\; y \;\; p \in Y) \tag{15}$$

  Al igual que la intersección, la unión de conjuntos es conmutativa, asociativa e idempotente.

[3]La **idempotencia** es la propiedad para realizar una acción determinada varias veces y aun así conseguir el mismo resultado que se obtendría si se realizase una sola vez.

- Extensiva y anti extensiva: Una transformación Ψ , sobre un conjunto X, es extensiva si el conjunto entrada está incluido en el conjunto salida y anti extensiva cuando el resultado de la transformación está incluido en el conjunto de la entrada:

$$Extensiva: \quad X \subseteq \Psi(X) \tag{16}$$

$$Anti\ extensiva: \quad X \supseteq \Psi(X) \tag{17}$$

El objetivo de las transformaciones morfológicos es la extracción de estructuras geométricas en los conjuntos sobre los que se opera, mediante la utilización de otro conjunto de forma conocida, al que se le denomina elemento estructurante. El tamaño y forma del elemento estructurante se elige, a priori, de acuerdo con la morfología sobre la que va a interseccionar y en función de la obtención de formas que se desea extraer.

Las operaciones que se usan en este trabajo son las siguientes:

- **Erosión binaria:**

  La transformación de la erosión es el resultado de comprobar si el elemento estructurante B está completamente incluido dentro del conjunto X. Cuando no ocurre, el resultado de la erosión es el conjunto vacío:

$$\varepsilon_B(X) = X \Theta B = \{x | B_x \subseteq X\} \tag{18}$$

  Cuando los objetos de la escena sean menores que el elemento estructurante, éstos desaparecerán. Otra interpretación de la erosión supone tomar el valor mínimo de la imagen en el entorno de vecindad definido por el elemento estructurante.
  Su utilidad consiste en definir una geometría determinada al elemento estructurante y pasarlo sobre la imagen. Los objetos menores al elemento estructurante no aparecerán en la imagen resultante. Los objetos que queden de la transformación habrán sido degradados. Por tanto, la erosión supone una degradación de la imagen. La aplicación iterativa de esta transformación hará que se eliminen todos los objetos existentes en la imagen. Un ejemplo de ello se tiene en la Figura 9.

Figura 9: Erosión binaria de una imagen digital. (tomada de [45])

- **Dilatación binaria:**

  La dilatación es la transformación dual a la erosión. El resultado de la dilatación es el conjunto de elementos tal que al menos algún elemento del conjunto estructurante B está contenido en el conjunto X, cuando B se desplaza sobre el conjunto X:

$$\delta_B(X) = X \oplus B = \{x | X \cap B_x \neq \emptyset\} \tag{19}$$

  La dilatación binaria representa un crecimiento progresivo del conjunto X. Al pasar el elemento estructurante dentro del conjunto, éste no se modificará. Sin embargo, en la frontera del conjunto X, al desplazar a B, el conjunto resultado se expansionará. La aplicación iterada de este operador haría degradar la imagen, haciendo coincidir el conjunto dilatado con la imagen. La dilatación es una transformación extensiva:

$$X \subseteq \delta_B(X) \tag{20}$$

  La dilatación también se interpreta como el valor máximo del entorno de vecindad definido por el elemento estructurante. En la Figura 10 se observa un ejemplo de esta operación.

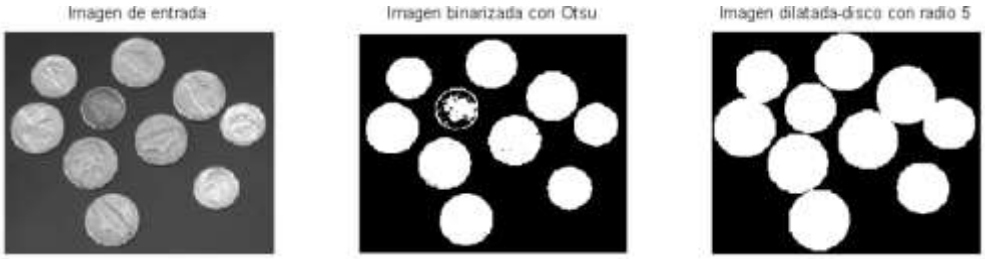


Figura 10: Dilatación binaria en una imagen digital. (tomada de [45])

Generalmente las operaciones de erosión y dilatación tienen el inconveniente de disminuir o aumentar el tamaño de objetos a los que no se les quiere alterar, pero este efecto puede ser subsanado con una aplicación en cascada de erosión y dilatación binaria con igual elemento estructurante, cabe la aclaración de que las aplicaciones de las operaciones de erosión seguida con una dilatación no son conmutativas. Los resultados son diferentes dando paso a las aperturas y cierres morfológicos de la siguiente manera:

$$Apertura\ (Opening): Erosión + Dilatación.$$

$$Cierre\ (Closing): Dilatación + Erosión.$$

- **Apertura morfológica:**
  La apertura binaria elimina todos los objetos que no están completamente contenidos en el elemento estructurante, pero además no disminuye el tamaño a los objetos que superen la erosión. Sin embargo, la imagen resultante no recupera la misma forma de los objetos filtrados de la imagen de entrada. Esta operación puede ser ideal para la eliminación de ruido, aunque no preserva la geometría de los objetos. Los bordes serán suavizados. Un ejemplo de esto es observable en la Figura 11.

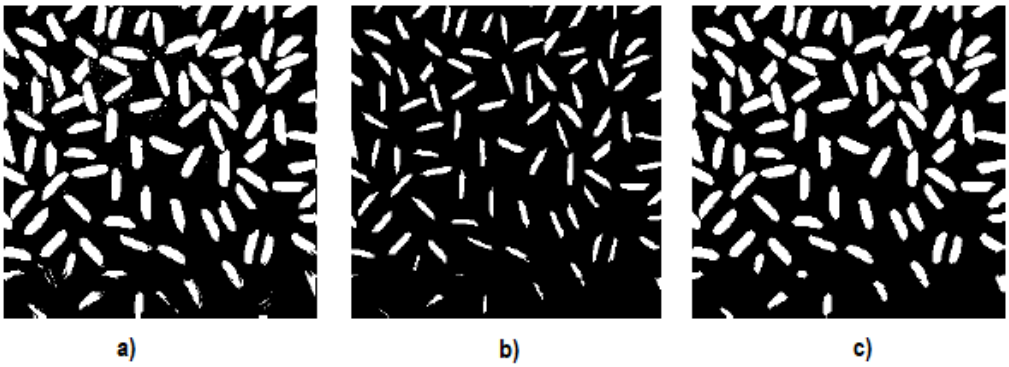


Figura 11: Apertura morfológica a imagen binarizada (modificada de [45])

En el ejemplo de la Figura 11, se tiene la parte a) que representa la imagen binarizada, lista para su procesamiento, el literal b) la misma imagen se encuentra la misma imagen, pero esta vez se aplica la operación de erosión con un elemento estructurante de radio 2, y finalmente en la parte c) se tiene la imagen resultante después de ser aplicada la apertura morfológica, es decir en la parte final se aplicó dilatación con el mismo elemento estructurante de radio 2.

- **Cierre morfológico:**
  El cierre binario morfológico produce que la dilatación rellene las estructuras que la erosión no puede separar. Los contornos de los objetos también serán suavizados, pero habiendo rellanado las fisuras. Un ejemplo es apreciado en la Figura 12.

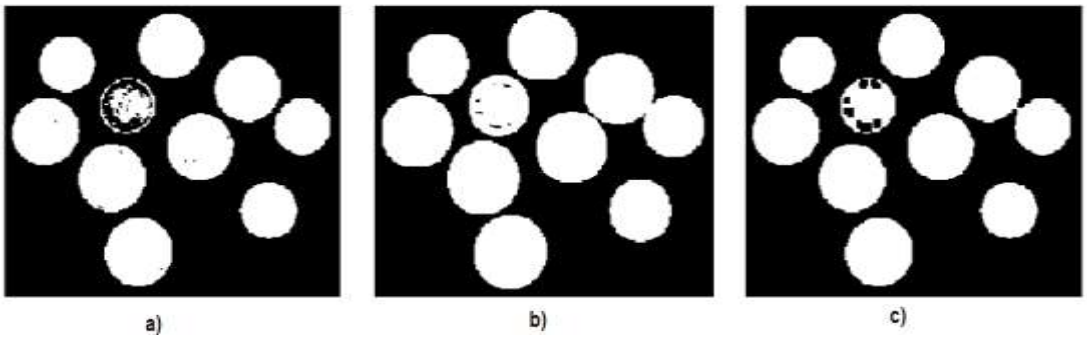


Figura 12: Cierre morfológico de una imagen binarizada (modificada de [45])

En la parte a) de la Figura 12 se tiene la imagen binarizada, en la parte b) se encuentra la misma figura, pero esta vez se le ha aplicado una dilatación

con un elemento estructurante de radio 5, y finalmente en la Figura 12, literal c, se le aplica una erosión con el mismo elemento estructurante de radio 5, obteniendo finalmente un cierre morfológico.

Se pueden usar estas dos operaciones (Cierre morfológico y Apertura morfológica) combinadas para obtener mejores resultados, tal como se percibe en la Figura 13, donde se aprecia una imagen a la cual se le practica inicialmente la transformación de apertura y finalmente una operación de cierre, como se evidencian los resultados son mejores que si únicamente se aplicara una sola operación.

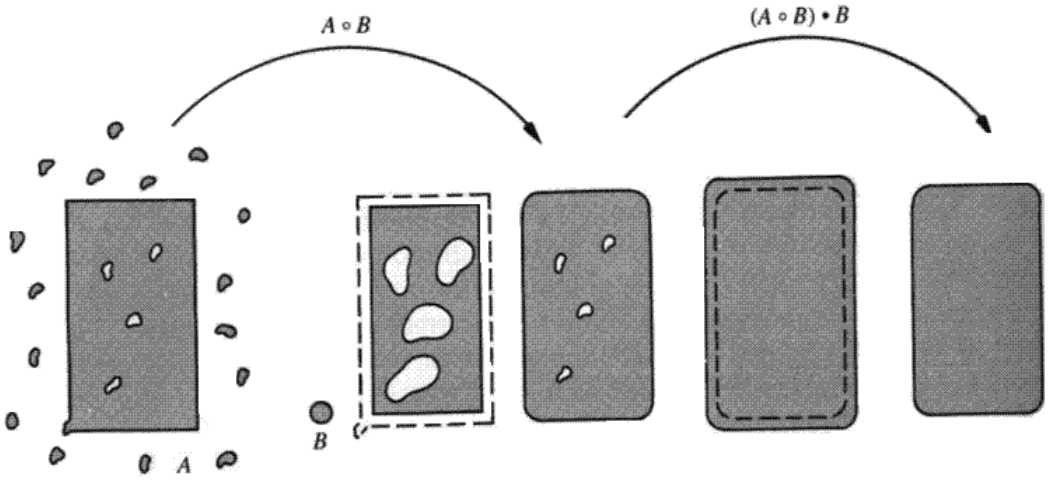


Figura 13: Apertura + Cierre morfológico (imagen tomada de apuntes de visión de máquina)

2.4.10. Sustracción de fondo

También conocido como detección de primer plano, es una técnica de procesamiento de imágenes y de visión por ordenador, donde el primer plano de una imagen se extrae para procesamiento adicional. En general, las regiones de una imagen de interés son objetos (los seres humanos, los carros, texto, etc.) en su primer plano. Después de la etapa de pre-procesamiento de imagen (que puede incluir la eliminación de ruido de imagen, post-procesamiento como morfología, etc.) se requiere la localización de objetos que pueden hacer uso de esta técnica.

La sustracción de fondo es un método ampliamente utilizado para la detección de objetos en movimiento en los videos de las cámaras estáticas. La meta de esteenfoque es el de detectar los objetos que se mueven a partir de la diferencia entre el cuadro actual y un marco de referencia, a menudo llamada "imagen de fondo", o "modelo de fondo". La sustracción de fondo se hace sobre todo si la imagen en cuestión es una parte de una secuencia de vídeo. Esto se puede aplicar para numerosas aplicaciones en visión por ordenador, por ejemplo, seguimiento o vigilancia humana, etc.

La sustracción de fondo se basa generalmente en un hipotético fondo estático, que a menudo no es aplicable en entornos reales ya que los métodos fondos estáticos tienen dificultades con las escenas al aire libre [46].

Se conocen múltiples métodos para realizar sustracción de fondo, estos son listados a continuación, junto con su definición general.

- ***Diferenciación de fotogramas***

Un algoritmo de detección de movimiento comienza con la parte de segmentación donde los objetos en primer plano (Foreground) o en movimiento están segmentados desde los objetos de fondo (Background). La manera más sencilla de implementar esto es tomar una imagen como fondo y tomar los fotogramas obtenidos en el momento *t*, denotados*I(t)* para comparar con la imagen de fondo denotada por *B*. Aquí usando cálculos aritméticos simples, se pueden segmentar los objetos simplemente usando la técnica de resta de imágenes de visión por computadora [47]para cada píxel en *I(t),* Tomando el valor de píxel denotado por *P [I (t)]* y restarlo con los píxeles correspondientes en la misma posición en la imagen de fondo denotada como *P [B].*

Su representación matemática está dada por:

$$P[F(t)] = P[I(t)] - P[B] \tag{21}$$

Donde el fondo es el fotograma en el instante t. Esta imagen de diferencia sólo mostraría cierta intensidad para las ubicaciones de píxeles que han cambiado en las dos tramas. Aunque aparentemente se ha eliminado el fondo, este enfoque sólo funcionará para los casos en que todos los píxeles de primer plano se mueven y todos los píxeles de fondo son estáticos.

- ***Filtro promedio***

En este método se usa para calcular la imagen que contiene solamente el fondo, se realiza mediante la resolución de una serie de imágenes, estas son promediadas [47].

$$B(x, y, t) = \frac{1}{N}\sum_{i=1}^{N} V(x, y, t - i) \tag{22}$$

En la ecuación (22), N es el número de imágenes precedentes tomadas para promediar. Esto quiere decir que lo que se va a promediar es el valor de los píxeles correspondientes en las imágenes dadas. N dependerá de la velocidad del video. Después de calcular el fondo B (x, y, t) se puede restarlo de la imagen V (x, y, t) en el tiempo t = t y su umbral. Así, el primer plano queda determinado por:

$$|V(x, y, t) - B(x, y, t)| > Th \tag{23}$$

Donde Th es el umbral. De manera similar también se puede usar la mediana en lugar de la media en el cálculo anterior de B (x, y, t).

- ***Running gaussian average***

Para este método, se propone la adaptación de una función de densidad probabilística gaussiana sobre los últimos n marcos[47]. Con el fin de evitar el ajuste de una función de densidad probabilística Gaussiana desde el principio a cada nuevo tiempo de trama t, se calcula un promedio en ejecución (o en línea acumulativa).

La función de densidad probabilística gaussiana de cada píxel se caracteriza por la media m y la varianza sigma.

Lo siguiente es una posible condición inicial (suponiendo que inicialmente cada píxel es de fondo):

$$\mu_o = I_o \tag{24}$$

$$\sigma_0^2 = algún\ valor\ por\ defecto \tag{25}$$

Donde I (t) es el valor de la intensidad del píxel variando en el tiempo t. Con el fin de inicializar la varianza, se puede, por ejemplo, utilizar la varianza en X e Y de una pequeña ventana alrededor de cada píxel.

Teniendo en cuenta que el fondo puede cambiar con el tiempo (por ejemplo, debido a cambios de iluminación u objetos de fondo no estáticos). Se debe re-acomodar ese cambio, en cada trama t, la media y la varianza de cada píxel deben ser actualizadas, de la siguiente manera:

$$\mu_t = \rho I_t + (1-\rho)\mu_{t-1} \quad (26)$$

$$\sigma_t^2 = d^2\rho + (1-\rho)\sigma_{t-1}^2 \quad (27)$$

$$d = |(I_t - \mu_t)| \quad (28)$$

Dondeσ(rho) determina el tamaño de la ventana temporal que se usa para ajustar la función de densidad probabilística gaussiana usualmente el valor de $\sigma = 0.01$ y d determina la distancia euclídea entre la media y el valor del píxel.

Ahora se puede clasificar cada pixel como pixel de fondo o pixel de primer plano, es pixel de fondo si su intensidad actual está dentro de cierto intervalo de confianza de la media de su distribución, se determina su clasificación cuando se reemplazan los valores en las ecuaciones (29) y (30):

$$\frac{|(I_t - \mu_t)|}{\sigma_t} > k \rightarrow Foreground \quad (29)$$

$$\frac{|(I_t - \mu_t)|}{\sigma_t} \leq k \rightarrow Background \quad (30)$$

Donde el parámetro k es un umbral libre (usualmente $k = 2.5$). Un valor mayor para k permite un fondo más dinámico, mientras que un k más pequeño aumenta la probabilidad de una transición de fondo a primer plano debido a cambios más sutiles.

- ***Modelo de mezcla gaussiana:***

El modelo de mezcla gaussiana es la base del algoritmo elegido para el software a desarrollar, la primera parte de la implementación de esta función se basa en determinar si cada pixel muestreado pertenece al marco del primer plano (ForeGround, FG) o al marco del fondo (BackGround, BG) [48], teniendo en cuenta que el valor de un cada píxel muestreado en el tiempo *t* en RGB o algún

otro espacio de colores se denota$\vec{x}^{(t)}$. La decisión de a que marco pertenece cada píxel está dada por la “decisión bayesiana” determinada por la ecuación(31).

$$R = \frac{p\left(BG\middle|\vec{x}^{(t)}\right)}{p\left(FG\middle|\vec{x}^{(t)}\right)} = \frac{p\left(\vec{x}^{(t)}\middle|BG\right)p(BG)}{p\left(\vec{x}^{(t)}\middle|FG\right)p(FG)} \quad (31)$$

Los resultados de la substracción de fondo se propagan generalmente a algunos módulos de nivel superior, por ejemplo, los objetos detectados son a menudo rastreados. Mientras se rastrea un objeto se podría obtener algún conocimiento sobre la apariencia del objeto rastreado y este conocimiento puede ser usado para mejorar la sustracción de fondo[49], como un caso general en el que no se sabe nada acerca de los objetos de primer plano, ni cuándo se pueden ver, ni con qué frecuencia se van a presentar en escena se establece lo siguiente $p(FG) = p(BG)$y se asume una distribución uniforme para la apariencia del objeto en primer plano$p\left(\vec{x}^{(t)}\middle|FG\right) = c_{FG}$, se tiene la certeza que el pixel pertenece al fondo si y solo si:

$$p\left(\vec{x}^{(t)}\middle|BG\right) > C_{thr}(= R_{CFG}) \quad (32)$$

DondeC_{thr}es un valor umbral.

$p\left(\vec{x}^{(t)}\middle|BG\right)$hace referencia al modelo de fondo. Este modelo se estima a partir de un conjunto de entrenamiento denotado como χ. El modelo estimado se denotará por $\hat{p}\left(\vec{x}^{(t)}\middle|\chi, BG\right)$ y depende del conjunto de entrenamiento como se indica explícitamente, se asume que las muestras son independientes y el principal problema es ¿cómo estimar eficientemente la función de densidad y adaptarla a los posibles cambios? Aquí en la implementación de este algoritmo el autor presenta una mejora del GMM.

En la práctica, la iluminación en la escena podría cambiar gradualmente (condiciones diurnas o climáticas en una escena al aire libre) o de repente (cambiar la luz en una escena interior). Un nuevo objeto podría ser llevado a la escena o un objeto presente eliminado de ella. Para adaptarnos a los cambios se puede actualizar el conjunto de entrenamiento añadiendo nuevas muestras y descartando las viejas. Para realizar esto se elige un periodo de tiempo razonable denominado T y en el tiempo$t$ se tiene $\chi_T = \{x^{(t)}, \dots, x^{(t-T)}\}$. Para cada nueva muestra se actualiza el conjunto de datos de entrenamiento χ_T y se calcula nuevamente$\hat{p}\left(\vec{x}^{(t)}\middle|\chi, BG\right)$. Sin embargo, entre las muestras de la historia reciente podría haber algunos valores que pertenecen a los objetos de primer plano y por

lo tanto hay que realizar dicha estimación, esta se denota de la siguiente manera$p\left(\vec{x}^{(t)}\middle|\chi_T, BG+FG\right)$. Y se utiliza en la ecuación(33) tomando M componentes:

$$\hat{p}\left(\vec{x}^{(t)}\middle|\chi_T, BG+FG\right) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I) \tag{33}$$

Donde $\hat{\vec{\mu}}_1, \ldots, \hat{\vec{\mu}}_M$ son las estimaciones de las medias y $\hat{\sigma}_1, \ldots, \hat{\sigma}_M$ son las estimaciones de las varianzas que describen los componentes gaussianos, suponiendo que las matrices de covarianza son diagonales y la matriz identidad *I* tiene dimensiones apropiadas. Los "*pesos de mezcla*" están indicados por $\hat{\pi}_m$ estos no son negativos y su suma tiene como resultado el número uno. Dada una nueva muestra de datos$\vec{x}^{(t)}$ en el tiempo t las ecuaciones de actualización recursiva son:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) \tag{34}$$

$$\hat{\vec{\mu}}_m \leftarrow \hat{\vec{\mu}}_m + o_m^{(t)} (\alpha/\hat{\pi}_m)\vec{\delta}_m \tag{35}$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T \vec{\delta}_m - \hat{\sigma}_m^2) \tag{36}$$

Donde $\vec{\delta}_m = \vec{x}^{(t)} - \hat{\vec{\mu}}_m$. En lugar del intervalo de tiempo T que se menciona anteriormente, la constante α describe una envolvente exponencialmente decadente que se utiliza para limitar la influencia de los datos antiguos y se mantiene la misma notación teniendo en cuenta que $\alpha = 1/T$ aproximadamente. Para una nueva muestra, la propiedad $o_m^{(t)}$ se iguala a 1 para el componente más 'cercano' y con el mayor valor que pueda tener $\hat{\pi}_m$, en cuanto a los demás componentes se igualan a cero. Se define que una muestra es "cercana" a un componente si la distancia de Mahalanobis del componente es por ejemplo menos de tres desviaciones estándar.

Este nuevo modelo de mezcla gaussiana permite que el número de componentes por pixel sea seleccionado automáticamente, y esto mejora bastante el tiempo de procesamiento y ligeramente la segmentación comparado con los otros métodos.

Al elegir el número de componentes automáticamente para cada píxel se convierte en un algoritmo que puede ser implementado para uso en tiempo real, el algoritmo puede adaptarse automáticamente a la escena.

3. Diseño del algoritmo

Este capítulo está dedicado al diseño, la implementación y la explicación del algoritmo que se creó, adicionando pseudocódigos y diagramas de flujo para su posterior análisis, también se consideran trabajos y algoritmos similares que fueron de gran ayuda para comprender y diseñar este software a partir de la librería OpenCV.

El algoritmo diseñado se encuentra dividido en tres etapas, estas se consideraron apropiadas para solucionar el problema, se visualizan en la Figura 14.

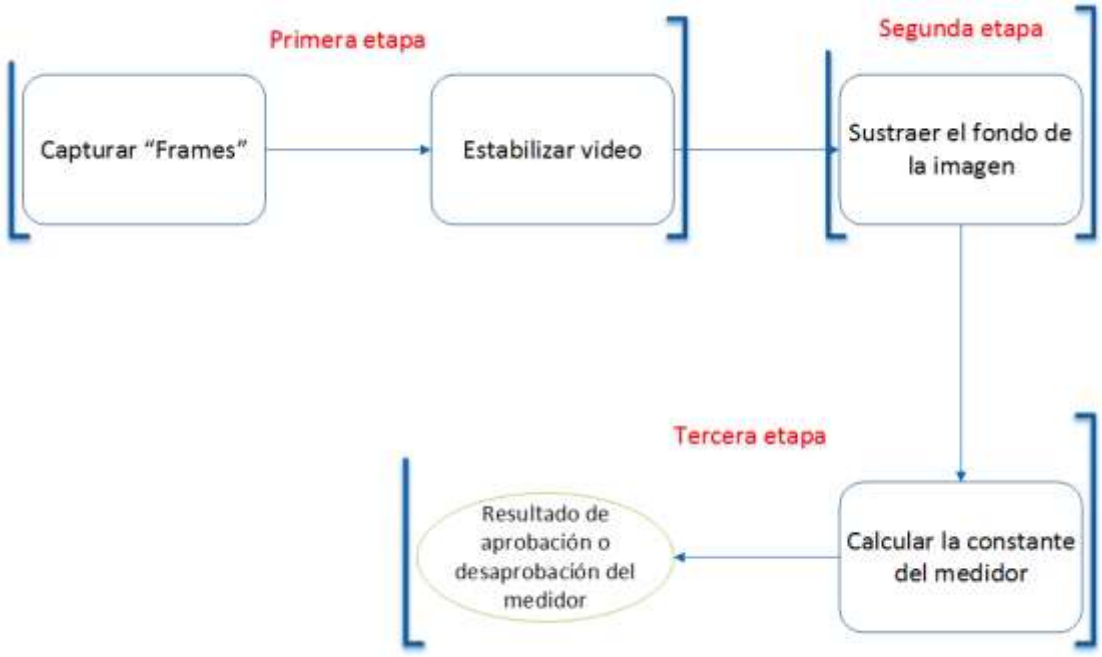


Figura 14:Etapas generales del algoritmo diseñado (realizada por el autor)

A continuación, se describe cada etapa por separado, los requerimientos de estas mismas y como se abordaron.

3.1. Primera etapa: Estabilización de video:

En la primera etapa del algoritmo se plantea solucionar el problema de estabilidad del video en tiempo real, se ha tomado como referencia el trabajo [50] de Yue Yang y Feng Lin, trabajo que se publicó en el año 2017, este trabajo describe el

desarrollo de un nuevo algoritmo para abordar el problema de la estabilización de vídeo en tiempo real para vehículos aéreos no tripulados. El algoritmo propuesto evita la necesidad de estimar el modelo de movimiento más usado que es la transformación proyectiva, y considera modelos de movimiento más simples, tales como la transformación rígida y la transformación de similaridad, también un gran aporte que hace este algoritmo es conseguir una alta velocidad de procesamiento, para esto se emplea el seguimiento basado en flujo óptico en lugar de usar métodos de seguimiento y emparejamiento convencionales. La idea es construir un modelo apropiado para la trayectoria de movimiento global, estimar los parámetros de movimiento entre dos tramas sucesivas utilizando un detector de punto clave muy eficiente y compensar el movimiento a través de un seguimiento óptimo basado en el flujo óptico de los puntos clave.

En primer lugar, la novedad clave del algoritmo propuesto radica en la estimación de alta velocidad de la trayectoria del parámetro de movimiento entre dos tramas consecutivas, en el que otros métodos son menos eficientes. Esto se debe a que la fase de estimación de movimiento elige una homografía[4] de orden inferior apropiada, ya sea rígida o de similitud entre tramas. Esto es para que el algoritmo sea capaz de usar con frecuencia el orden inferior de la homografía para estimar los parámetros de movimiento a menos que haya una necesidad de utilizar un orden superior. El resultado: se consigue una alta velocidad de procesamiento. En segundo lugar, la fase de compensación de movimiento, que se propone, suaviza los parámetros a través de una ventana de promedio. Por último, actualizar el marco según los parámetros suavizados. El programa es capaz de procesar hasta 100 fotogramas. Esto hace que el algoritmo sea fácilmente implementado en aplicaciones de tiempo real, resumiendo, el algoritmo tendría tres sub-etapas las cuales serían: sub-etapa de estimación de movimiento, sub-etapa de estimación de homografías (basado en la transformada rígida) y sub-etapa de compensación de movimiento y composición de imagen.

Estas sub-etapas están representadas en el siguiente diagrama de bloques:

[4] Homografía: toda transformación proyectiva que determina una correspondencia entre dos figuras geométricas planas, de forma que a cada uno de los puntos y las rectas de una de ellas le corresponden, respectivamente, un punto y una recta de la otra.

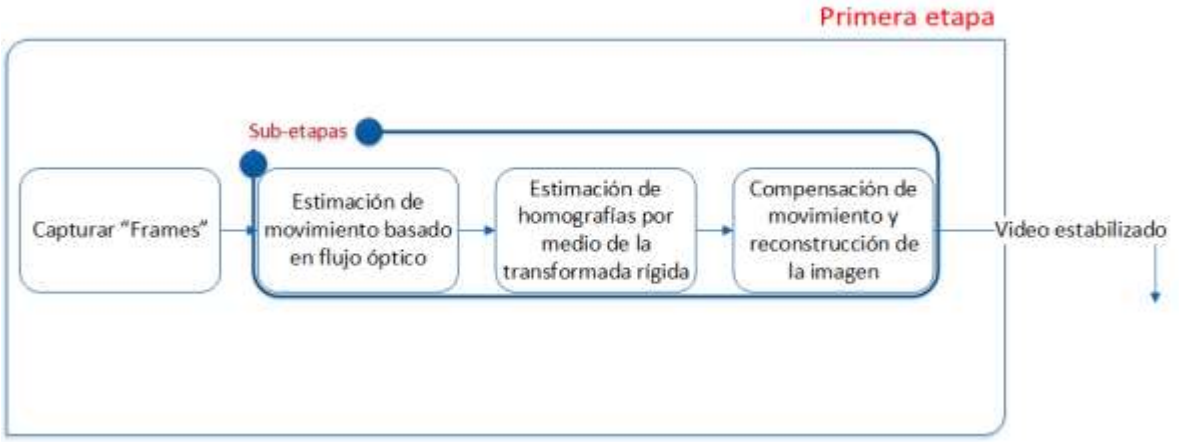


Figura 15: Diagrama de bloques de sub-etapas de la primera etapa del algoritmo (realizada por el autor)

3.1.1. Primera sub-etapa: estimación de movimiento basado en flujo óptico

El primer paso de este proceso es la estimación de movimiento, también es el paso que más tiempo consume, el principal objetivo del trabajo citado consistió en reducir significativamente el tiempo en el que se calcula la trayectoria de los parámetros de movimiento entre las tramas consecutivas en el video. La trayectoria de los parámetros se calcula partiendo de como los "Puntos característicos" se mueven entre los marcos consecutivos para inicializar la estimación de movimiento, en ese orden de ideas el primer paso es hallar los puntos característicos.

Los puntos característicos de la imagen son lugares donde se presentan grandes variaciones de intensidad en todas las direcciones, y son muy importantes en el paso de estimación de movimiento, ya que determinan la calidad de la estimación de movimiento

El método utilizado para detectar puntos característicos es el "detector de esquinas Shi-Tomasi" expuesto en el trabajo [40] presentado por Jianbo Shi junto a Carlo Tomasi, siendo este una mejora del detector de esquinas de Harris. Para la implementación en el algoritmo se usa la función **`goodFeaturesToTrack()`**, esta función se encuentra dentro de la librería OpenCV, una vez se tienen identificados los puntos característicos, se propone una coincidencia basada en flujo óptico entre tramas consecutivas, de los puntos anteriormente hallados, el flujo óptico está definido como el movimiento aparente de los objetos, en una secuencia de imágenes, y esto se ve representado en un vector de desplazamiento (para cada punto) que muestra el movimiento de los puntos entre dos fotogramas o frames consecutivos, para realizar este procedimiento se usa el

algoritmo de Lucas-Kanade, basado en flujo óptico, se lleva a la implementación mediante la función **`calcOpticalFlowPyrLK()`**, con la cual se obtienen los vectores de desplazamiento o vectores de flujo óptico. Un ejemplo de cómo funciona el seguidor de características se observa en la Figura 16.

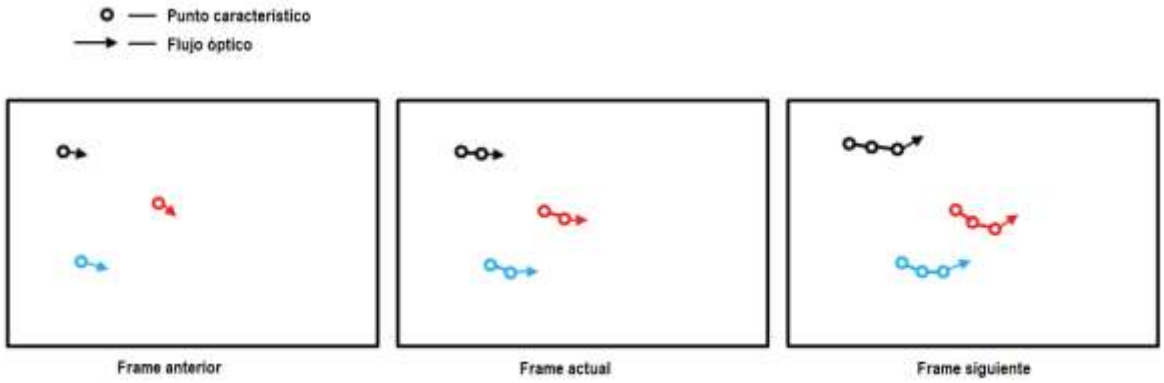


Figura 16: Estimación de movimiento por flujo óptimo (adaptada de [50])

La mejora en esta parte del proceso consiste en que para otros trabajos se usan alrededor de 1000 puntos de características para el proceso de estabilizar un video, lo que consume muchos recursos computacionales y no permite que este pueda ser implementado en aplicaciones de tiempo real, los autores del trabajo presentado en [40] realizaron experimentos y concluyeron que basta solamente 50 puntos característicos para obtener una buena estabilización de video implementable en tiempo real, una muestra de cómo funciona este algoritmo se ilustran en la Figura 17 y en la Figura 18.

Figura 17: Detección de puntos característicos por medio de la función shi-tomasi (realizada por el autor)

Figura 18: Seguimiento de los parámetros de movimiento por medio de flujo óptico (imagen creada por el autor)

En la Figura 17 se tiene la detección de los puntos característicos por medio de la función ya mencionada, estos puntos son a los cuales se les van a calcular el vector de desplazamiento, lo que se percibe en la Figura 18, donde se tienen los mismos puntos característicos pero esta vez, en verde que es donde se encontraban los puntos en el frame anterior y en rojo, que es donde se encuentran los puntos en el frame actual.

3.1.2. Segunda sub-etapa: estimación de homografía por medio la transformada rígida

El principal objetivo de esta etapa es calcular cuantitativamente el desplazamiento entre el frame actual y el frame anterior, estos valores se estiman con ayuda de los vectores de flujo óptico calculados previamente. Lo que se debe saber antes de iniciar es que una secuencia de imágenes puede tener distintos tipos de desplazamiento, generado por el movimiento del objeto o el movimiento de la cámara, entre frames consecutivos, estos tipos de desplazamiento están gráficamente descritos en la Figura 19.

Figura 19: Posibles desplazamientos en una imagen (imagen realizada por el autor)

Lo que se propone es hallar los factores de desplazamiento por medio del cálculo de transformaciones rígidas o transformaciones afín. Con esta operación se obtienen varias matrices con los datos o factores de desplazamiento, tal como se observa en la Tabla 4, es válido notar que estas transformaciones pueden calcular los datos de desplazamiento de hasta seis grados de libertad, que se resumen en cuatro movimientos: translación, rotación, cizallamiento y escalamiento.

La función base para cumplir con esta parte del algoritmo es **`estimateRigidTransform()`**, la cual es capaz de calcular las matrices anteriormente mencionadas, esta función es apta para estimar transformaciones completas, es decir que tengan 6 grados de libertad (rotación, translación, escalamiento y cizallamiento) y también transformaciones parciales, que únicamente constan de 4 grados de libertad (rotación, translación y escalamiento uniforme). Lo que se propone es computar transformaciones parciales y omitir el cizallamiento, ya que este fenómeno es el menos propenso a ser presentado y omitiéndolo se ahorra tiempo de ejecución en la aplicación.

3.1.3. *Tercera sub-etapa:* Compensación del movimiento y composición de la imagen

En esta última sub-etapa de la primera parte lo que se hace es la suavización de las trayectorias para así obtener finalmente un video más estable, aquí se empleó un método llamado “Suavizado de ventanas de promedio”, se utiliza un procedimiento básico para acumular parámetros de trayectoria y luego suavizarlos usando una ventana de promedio, este método consiste en tratar de llevar un punto característico en el frame actual a la coordenada en donde se encontraba ese mismo punto en el frame anterior, llevar este paso al software en creación se hizo posible mediante el uso de la función **`warpAffine()`**, la cual trabaja bajo los datos de desplazamiento que han sido calculadas previamente, para la reconstrucción del video, es válido notar que al momento de reconstruir la imagen

los cuadros suavizados se aplican sobre un cuadro base, que en este caso es una imagen sin desplazamiento, lo que podría generar que no toda la imagen se sitúe dentro de la ventana del video, a este método se le llama adición de imágenes mediante deformación, esto se puede ver reflejado en la Figura 20.

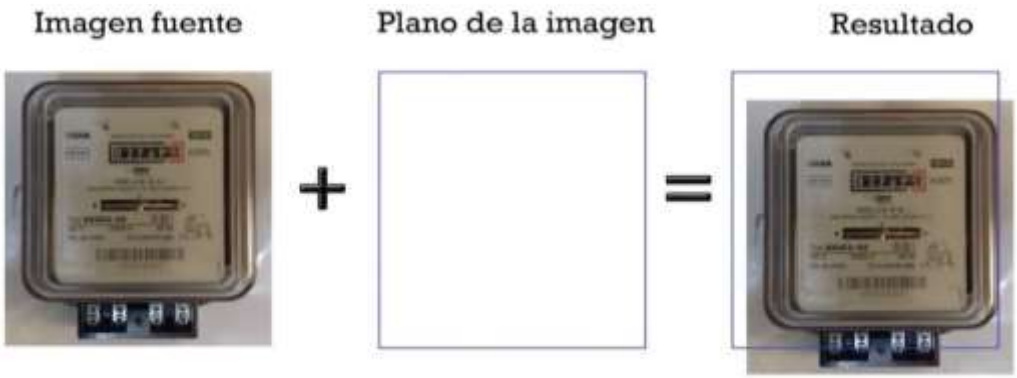


Figura 20: Adición de cuadros por deformación (imagen creada por el autor)

Es más fácil observar el resultado total de este método en video, por eso se llevó a cabo la implementación de esta parte del código en un IDE de escritorio y se grabó un video comparando el video inestable contra el video estable gracias al algoritmo, se adjuntan dos enlaces en los cuales se encuentran videos estabilizados.

https://goo.gl/yqSgLw

https://goo.gl/kaJuKu

3.1.4. Diagrama de flujo de la primera etapa del algoritmo

El diagrama de flujo de la primera etapa del algoritmo queda representado de la siguiente manera:

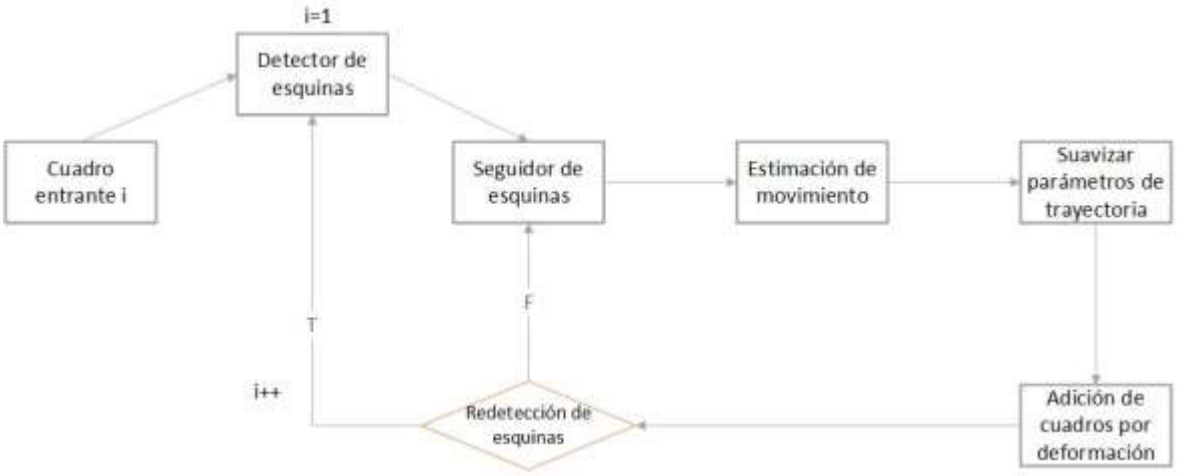


Figura 21: Diagrama de flujo de la primera etapa del algoritmo (adaptada de [50])

3.1.5. Pseudocódigo de la primera etapa del algoritmo:

Algoritmo 1: Video estabilización

Input: Secuencia de Frames
Output: Video estabilizado
1: Se capturan frame anterior y frame actual
1: Encontrar los puntos característicos con en el frame anterior y actual con la función goodFeaturesToTrack()
2: Eliminar los puntos que no coinciden en el frame actual con respecto al frame anterior
3: Calcular el flujo óptico con los puntos característicos restantes mediante la función calcOpticalFlowPyrLK()
4: Calcular la transformada rígida del frame anterior al frame actual utilizando la función estimateRigidTransform(), aquí se obtiene el desplazamiento que se de $dx, dy, d\theta$
5: Con los datos de desplazamiento, suaviza el movimiento entre el frame anterior y el frame actual, se lleva a cabo con la función wrapAffine()

3.2. Segunda etapa: sustracción de fondo

En la segunda parte del algoritmo, que consiste en contrarrestar los cambios de iluminación, se va a utilizar la ayuda del algoritmo "*BackgroundSubstractorMOG2*", este algoritmo fue propuesto por Zoran Zivkovic, en el año 2006 y descrito en su trabajo [48], llamado "*Improved adaptive Gaussian mixture model for background*

subtraction" siendo este una mejora del trabajo publicado por el mismo autor, descrito en [49]llamado "*Efficient adaptive density estimation per image pixel for the task of background subtraction*" y publicado en el 2004. Este método es una mejora del modelo de gaussianas mixtas o modelo de mezcla gaussiana (GMM) propuesto por N. Friedman and S. Russell, en su trabajo [51] desarrollado en 1997.

Para esta sección se va a aplicar el algoritmo solamente a la región de interés, que es la única que provee la información para poder calcular la constante del medidor, esta región se encuentra enmarcada de color verde en la Figura 22, otra razón para reconocer y trabajar únicamente en esta zona es para evitar falsos positivos en otras partes de la escena que no son importantes.

Figura 22: Región de interés en el contador (imagen realizada por el autor)

Para reconocer esta zona o región de interés lo que primero se hace es pre-procesar digitalmente la imagen para así obtener una escena limpia, es decir sin ruido y figuras que puedan afectar el reconocimiento de esta zona, para eso se umbraliza o binariza la imagen y después se le aplican transformaciones morfológicas denominadas transformaciones de apertura y cierre, que interiormente utilizan operaciones más pequeñas de erosión y dilatación. Con esto se asegura que los contornos que se pretenden hallar están completos y también se salvaguarda que se han removido las partículas más pequeñas de la imagen, una vez hecho esto se usa una función de reconocimiento de contornos, llamada

`findContours()` que internamente utiliza un algoritmo propuesto por Satoshi Suzuki en su trabajo [52] llamado "*Topological structural analysis of digitized binary images by border following*" y publicado en 1985 donde facilita el reconocimiento de bordes en una imagen digital, una vez hecho esto se debe condicionar una sola región acorde a la forma del rectángulo que se desea reconocer, para cumplir este objetivo se calcula la masa de cada contorno hallado con el apoyo de la función llamada `moments()` que está basada en el "*Teorema de Green*" y es usada para calcular volúmenes de superficies o áreas de figuras con la resolución de una serie de integrales, esta retorna un valor de masa para cada contorno, después se comparan entre si y el contorno elegido será en que más masa tenga, a este se le dibuja un rectángulo a su alrededor para diferenciarlo del resto de área. Y a esta pequeña región es a la que se le aplica el algoritmo `BackgroundSubstractorMOG2()`.

En la Figura 23 se describe el pseudocódigo en diagrama de bloques de cómo se llevó a cabo la implementación de esta etapa del algoritmo.

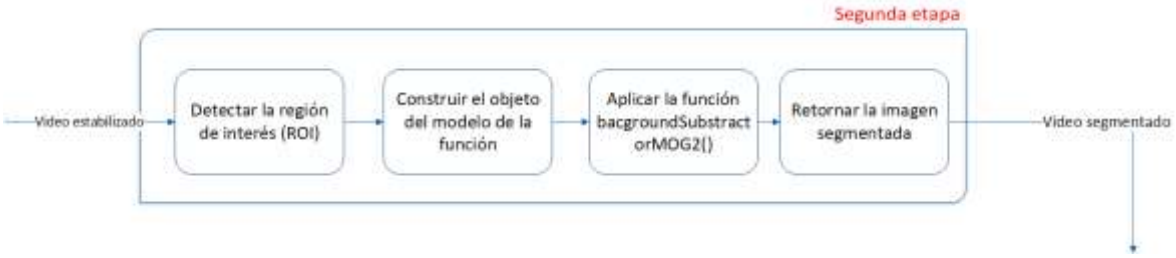


Figura 23: Diagrama de bloques de la segunda etapa (Realizada por el autor)

El pseudocódigo que describe en palabras la implementación de la función se aprecia a continuación:

Algoritmo 2: Sustracción de Fondo

Input: Video estabilizado
Output: Video segmentado

1: Reconocer la región de interés.
2: Construir el objeto del modelo backgroundSubstractorMOG2().
3: Aplicar la función backgroundSubstractorMOG2() únicamente sobre la región de interés.
4: Retornar el ForeGround.

3.3. Tercera etapa: cálculo de la constante:

En esta etapa, al igual que en las anteriores, se van a usar funciones, las cuales resuelvan esta parte del problema, lo que se busca es contar la marca del disco giratorio un número determinado de veces (siete) y contar el tiempo en el que el medidor se tomó en hacer este proceso, después de eso se procede a calcular el error porcentual que es el quien determina si el medidor es apto o por el contrario no lo es (ver Figura 3), inicialmente se usó un “alertador” para saber cuándo la aplicación está lista para realizar el conteo, esto se hizo con el fin de asegurar que la región detectada en el punto anterior sea la correcta, cuando dicha región sea la verificada la persona encargada de la prueba deberá pulsar un botón para así iniciar la prueba, para realizar la medición del tiempo se modificó la función `chronometer()` de la librería de Android Studio ya que esta no cuenta con conteo de milésimas de segundo, fundamental para que la prueba tenga buena exactitud, para el conteo de las revoluciones se usó un contador que va a ser activado cuando la marca sea detectada dentro de la región de interés, y para detectarla nuevamente se usa la función `moments()` para calcular la masa de los objetos que sean detectados dentro de la región de interés y solamente se va a contar cuando dicho objeto pase de determinado valor, esto elimina falsos conteos que pueden ser generados por el ruido del video, también para evitar que la marca sea leída varias veces dentro una sola revolución cada que haya un conteo se desactiva la detección del objeto por un número determinado de frames hasta que la marca salga de la región de interés, una vez el contador llegue a siete se detendrán estos dos procesos, se exporta el valor del tiempo en el que se realizó la prueba y se calcula el error con ayuda de las ecuaciones (1) y (2), un reto importante en esta etapa fue entender que Android Studio inicialmente solo puede llevar a cabo un proceso y este es el de mantener activa la interfaz, lo que hace que los procesos del contador y el cronómetro solo puedan ser llevados a la par con la interfaz creando “hilos” para que estos sean reconocidos como subprocesos y que se puedan ejecutar a la vez.

En la Figura 24 se muestra una captura de pantalla de móvil, en el sector izquierdo se tiene la cámara del celular donde se enfoca el rotor del medidor electromecánico el lado derecho está dividido en cuatro secciones, en la primera se encuentra el número de revoluciones contadas hasta el momento, en la segunda parte está el tiempo parcial o total dependiendo si la prueba está en proceso o ya haya terminado, en la tercera parte se encuentra el estado de la prueba y finalmente en la cuarta parte se encuentra el sector donde se va a mostrar el error calculado una vez finalizada la verificación.

Figura 24: Tercera etapa del algoritmo (imagen realizada por el autor)

Algoritmo 3: Cálculo del error de medición

Input: Video segmentado
Output: Porcentaje de error

1: Alertar la aplicación para que empiece el conteo.
2: Calcular la masa del objeto dentro de la ROI.
3: Activar el contador y el cronómetro si se cumple la condición de masa.
4: Realizar el conteo un número determinado de veces.
5: Cuando el contador llegue a dicho número, suspender el proceso de conteo y el proceso del cronómetro.
6: Extraer los datos del tiempo en el que se realizó la prueba
7: Calcular el error de medición.
8: Mostrar el error en pantalla.

4. Pruebas y resultados del algoritmo

La experimentación y evaluación del proyecto es la parte final, y de esta parte depende decidir si el algoritmo planteado y desarrollado es bueno para ser usado en el campo laboral por parte de los operarios de la CEO. Las pruebas al algoritmo se realizan primero a cada una de sus etapas, para determinar si estas cumplieron con su objetivo, y finalmente se realizan pruebas globales. Después de realizadas las pruebas resta concluir acerca de la eficiencia del algoritmo desarrollado.

4.1. Pruebas a la primera etapa

Para realizar las pruebas a la primera etapa se procedió a extraer ciertas cifras de los dos videos que se presentaron en la sección 3.1, estos datos son las transformadas rígidas para desplazamientos en ejes (x, y) y eje de rotación (θ) antes y después de ser suavizados, también se extrajeron las trayectorias antes y después de ser suavizados, y posteriormente se realizan gráficas y análisis a partir de ellas.

Extraer estos datos se hizo posible mediante la compilación del código en el computador, agregando algunas líneas de código que permiten su almacenamiento, para ello se usa el software “Qtcreator” en su versión de escritorio y para graficar estos valores se usa el software de Matlab.

Esta prueba se realizó dos veces con los datos de los dos videos presentados en la sección 3.1.

4.1.1. Primera prueba

Las figuras que se consideran a continuación pertenecen al primer video de la sección ya citada, cuando aún no se ha aplicado el algoritmo de estabilización, en primer lugar, la Figura 25 y la Figura 26 representan las variaciones, dadas en píxeles que se dan en los ejes de traslación (x, y) y en el eje de rotación θ respectivamente, como se observa en ambos casos presentan movimientos rudos, que son indeseables en cualquier video.

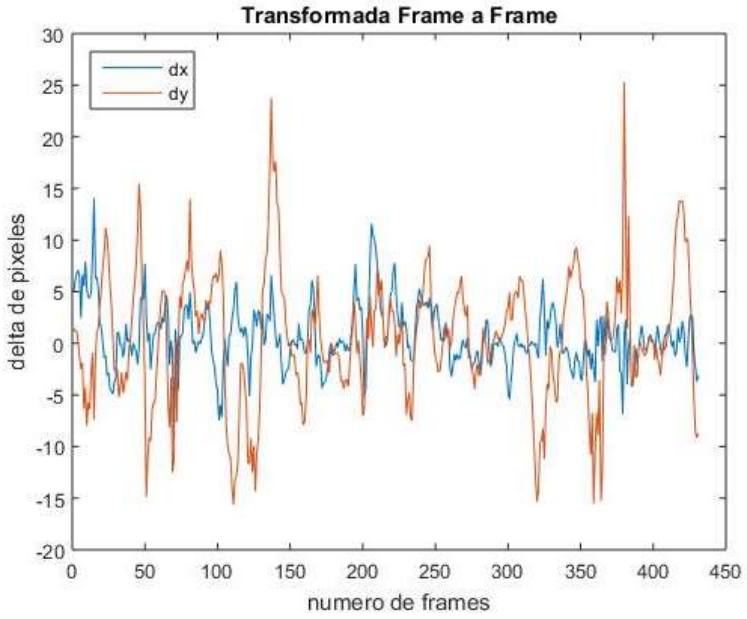


Figura 25: Transformada rígida en ejes (x, y)antes de ser suavizada (imagen realizada por el autor, prueba 1)

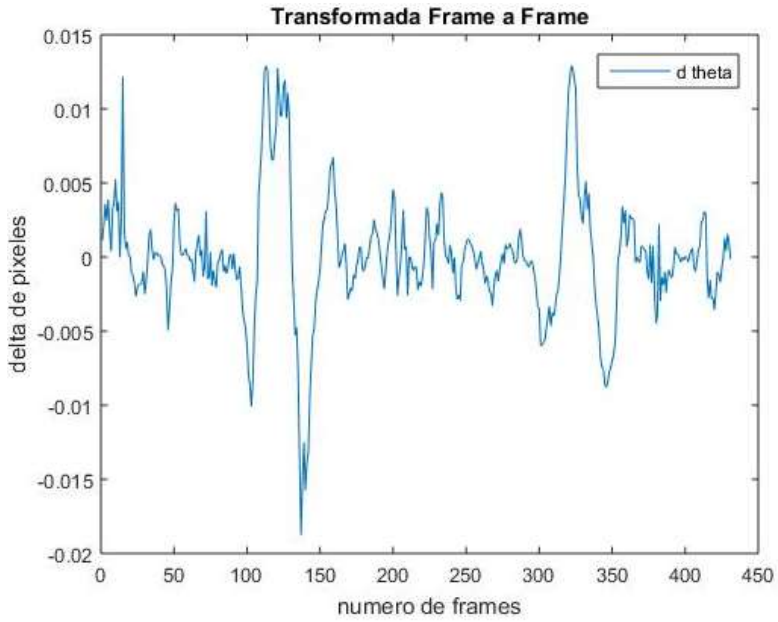


Figura 26: Transformada rígida en el eje θantes de ser suavizada (imagen realizada por el autor, prueba 1)

Las figuras que continúan en esta prueba corresponden al video cuando ya ha sido estabilizado por el algoritmo, para esta parte se presentan las comparaciones

de las trayectorias por separado en el eje x, eje y y eje θ antes y después de la aplicación del algoritmo.

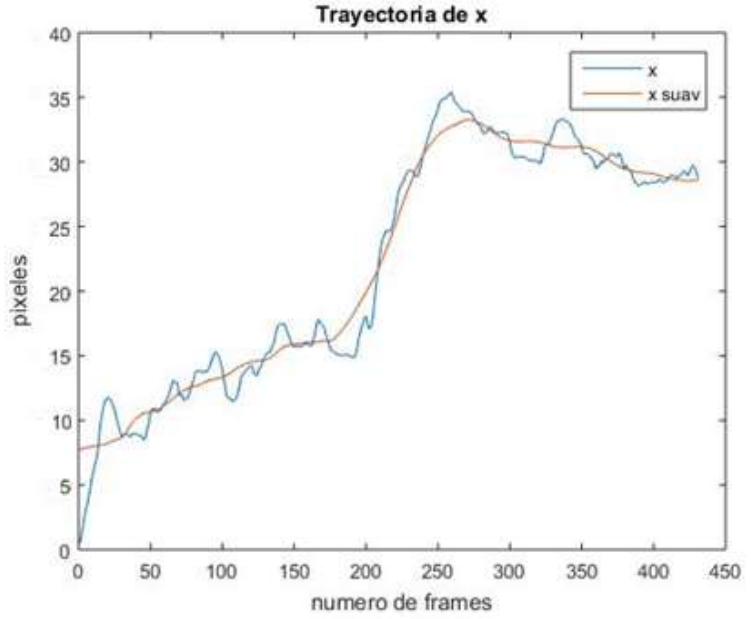


Figura 27: Trayectoria en el eje x. Antes y después de ser suavizada (imagen realizada por el autor, prueba 1)

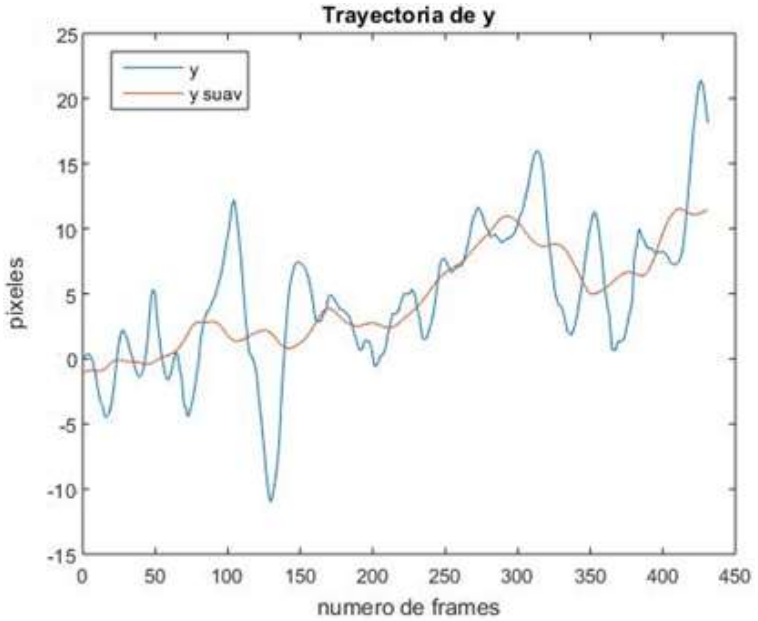


Figura 28: Trayectoria en el eje y. Antes y después de ser suavizada (imagen realizada por el autor, prueba 1)

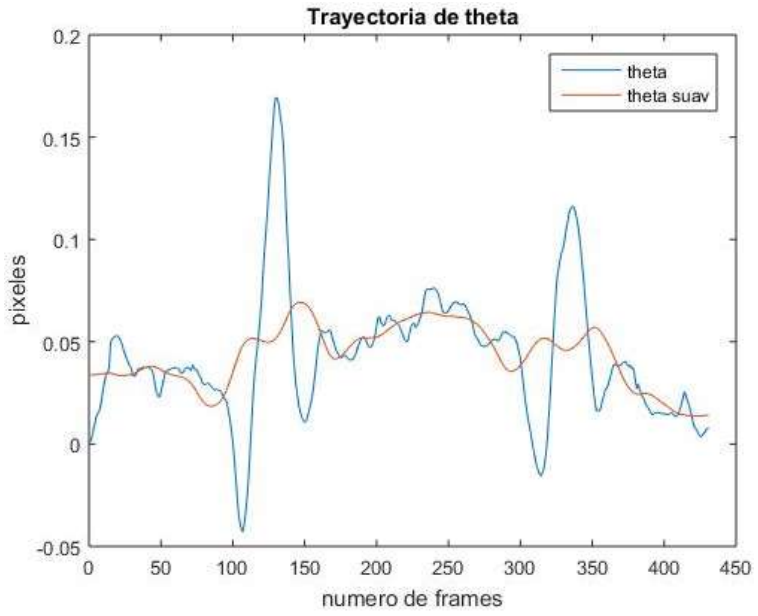


Figura 29: Trayectoria en el eje θ. Antes y después de ser suavizada (imagen realizada por el autor, prueba 1)

Como se observa, los resultados han sido positivos en las figuras, se contempla que para cada uno de los ejes la trayectoria ha sido suavizada, que es finalmente lo que se espera de esta parte del algoritmo.

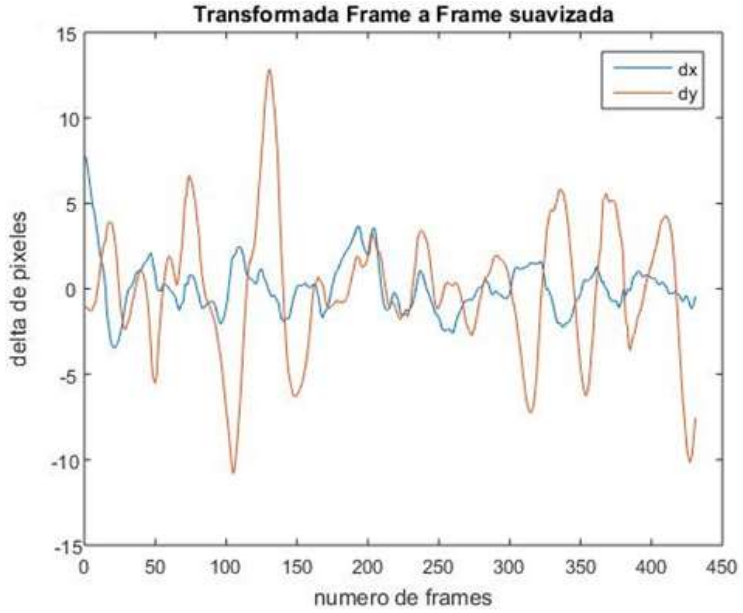


Figura 30: Transformada rígida en ejes (x, y)después de ser suavizada (imagen realizada por el autor, prueba 1)

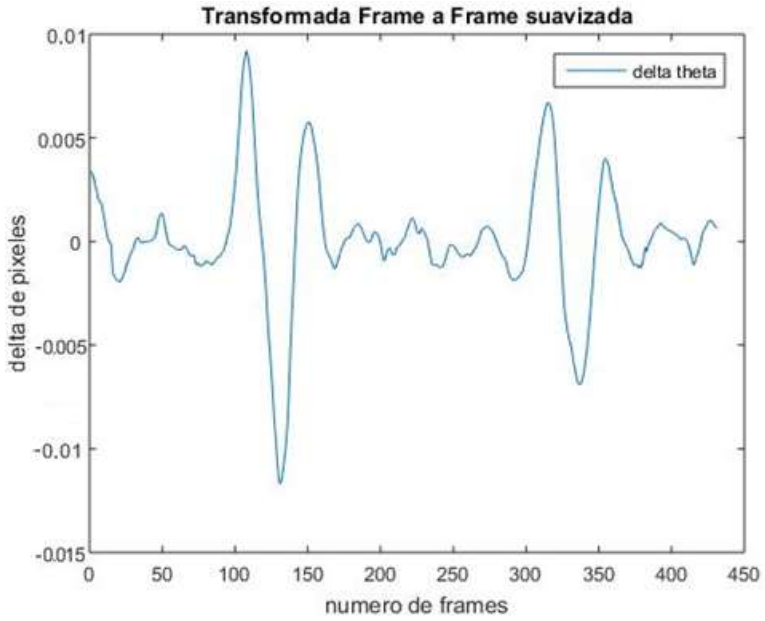


Figura 31: Transformada rígida en el eje θ, después de ser suavizada (imagen realizada por el autor, prueba 1)

Finalmente para la Figura 30 y la Figura 31 se ha vuelto a calcular la transformada rígida a los tres ejes que se están evaluando, la percepción es que dichas ilustraciones comparadas con la Figura 25 y la Figura 26 respectivamente se evidencian movimientos más armónicos, con lo que se concluyen resultados positivos.

También se anexan capturas de frames del video en cuestión, en donde se observa gráficamente resultados de lo explicado en la sección 3.1.

En la Figura 32, Figura 33 y Figura 34 se han ilustrado una secuencia de 9 frames tomados del video del cual se extrajeron los datos para las anteriores gráficas, en la parte izquierda de cada captura se observa el video original y en la derecha el video estabilizado, tal como se explica en la teoría el algoritmo trata de mantener los puntos característicos localizados (para el caso de los ejemplos los puntos no han sido graficados) en las mismas coordenadas en donde se encontraban en el frame anterior y es por eso que la escena tiende a salir del marco, los cambios más notorios se observan en los frames 132, 133, 134 y 135.

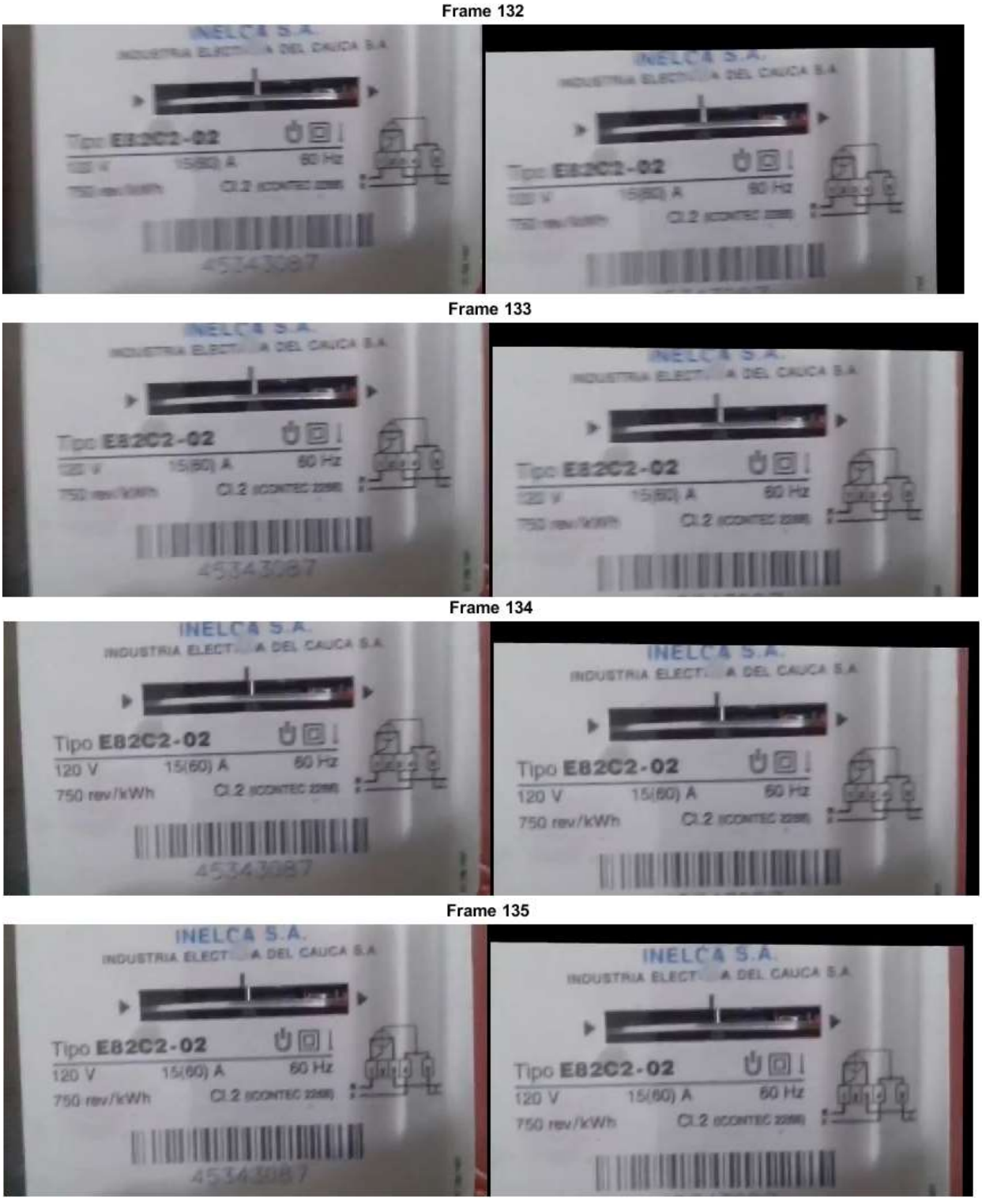


Figura 32: Captura de frames primer video estabilización 1 (imagen realizada por el autor)

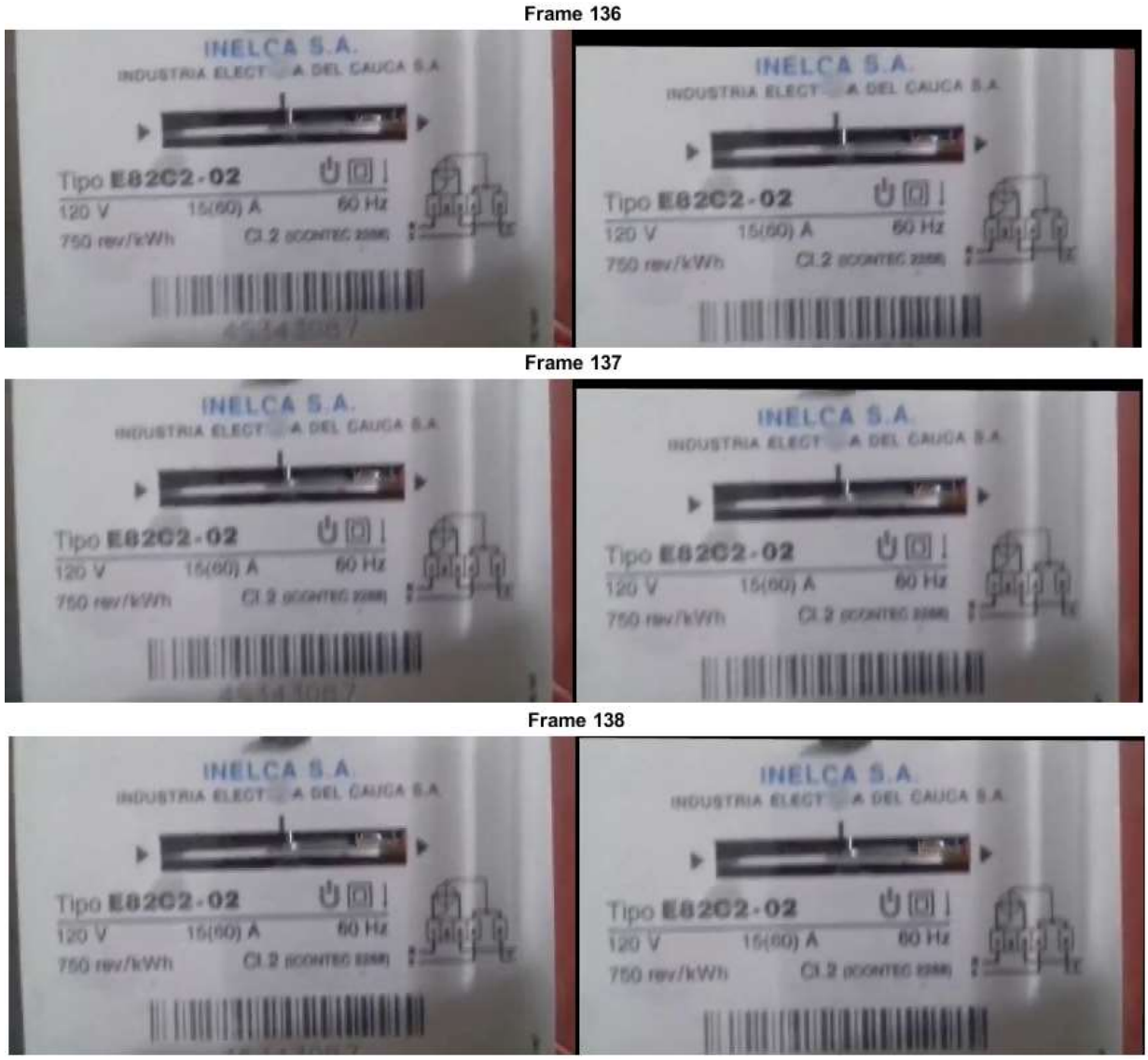


Figura 33: Captura de frames primer video estabilización 2 (imagen realizada por el autor)

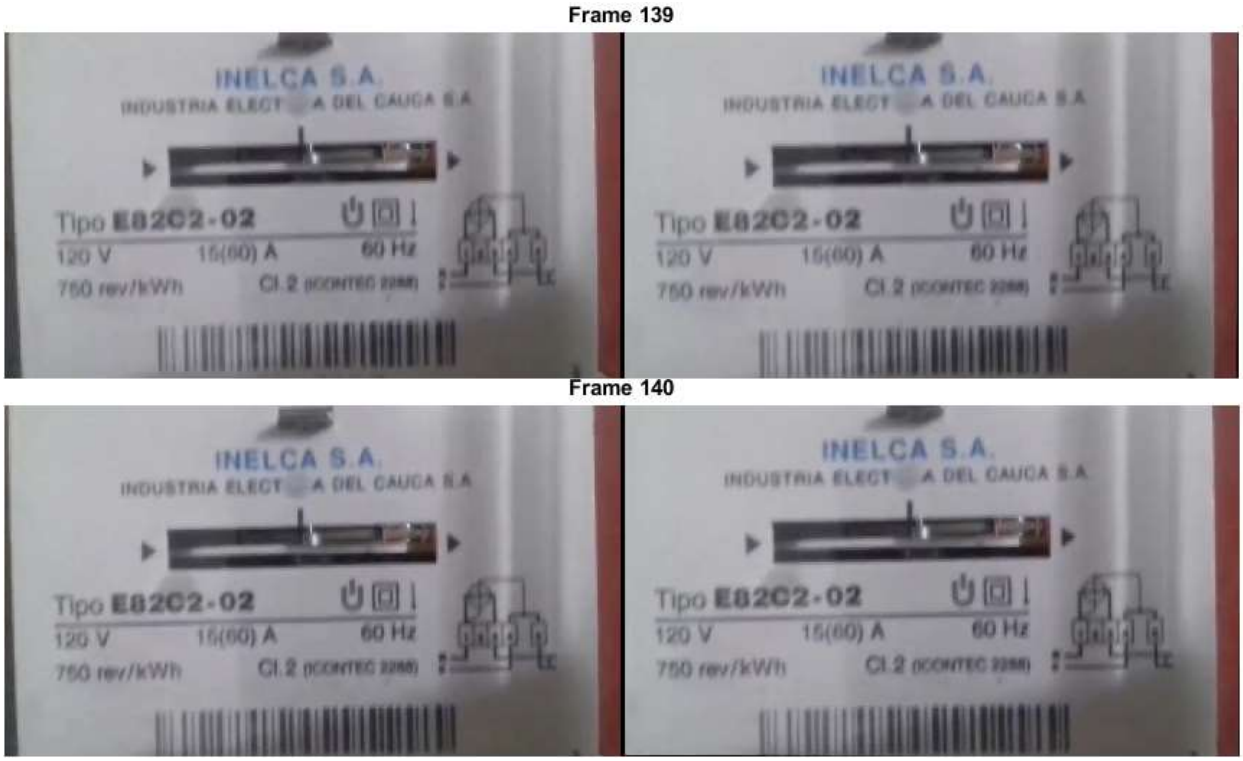


Figura 34: Captura de frames primer video estabilización 3 (imagen realizada por el autor)

4.1.2. Segunda prueba

La información con las que fueron realizadas las siguientes ilustraciones pertenecen al segundo video presentado en la sección 3.1, al igual que como se expuso en la anterior prueba, las primeras figuras pertenecen al cálculo de la transformada rígida para los ejes (x, y) y θ, Figura 35 y Figura 36 respectivamente. Donde nuevamente se perciben movimientos ásperos que hacen ver el video muy inestable.

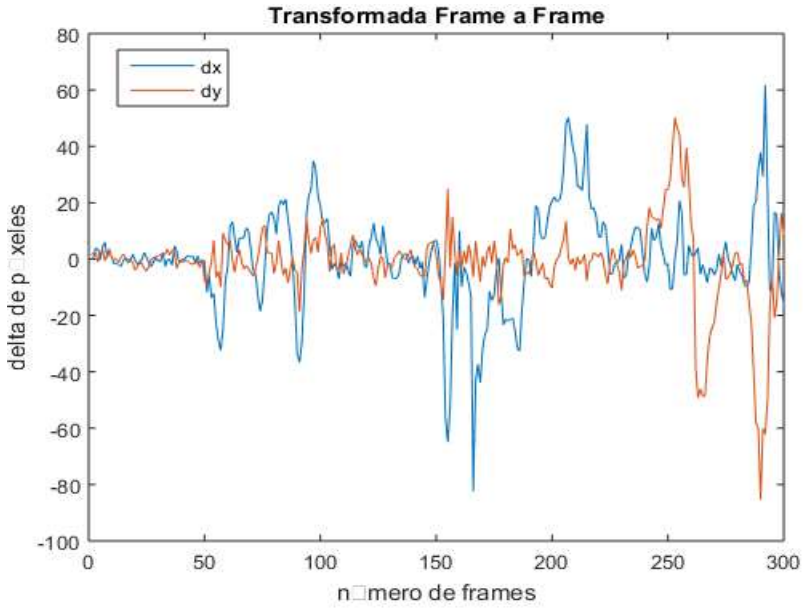


Figura 35: Transformada rígida en ejes (x, y) antes de ser suavizada (imagen realizada por el autor, prueba 2)

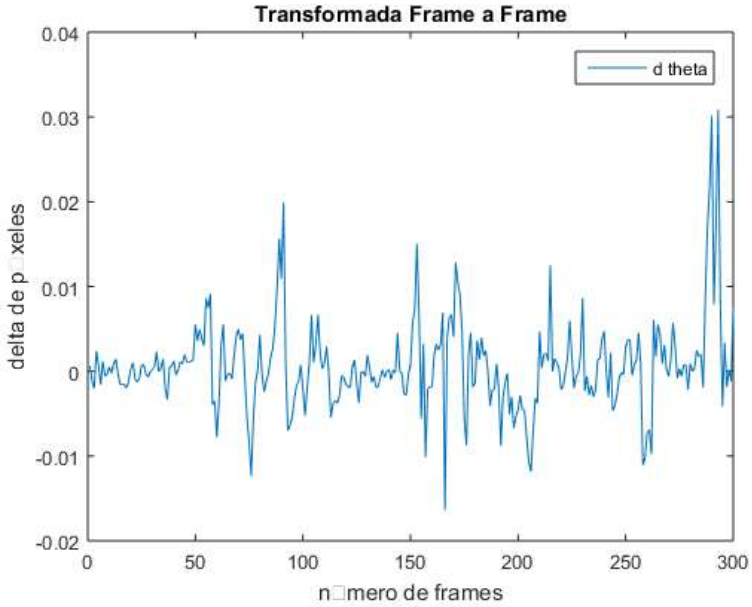


Figura 36: Transformada rígida en el eje θ antes de ser suavizada (imagen realizada por el autor, prueba 2)

El siguiente paso es ejecutar el algoritmo de estabilización sobre el video y recopilar la información. Posteriormente se grafican las trayectorias antes y después de ser suavizadas por separado, tal cual, en la anterior prueba, estas

ilustraciones están plasmadas en la Figura 37, Figura 38 y Figura 39, representando los ejes (x, y, θ) respectivamente.

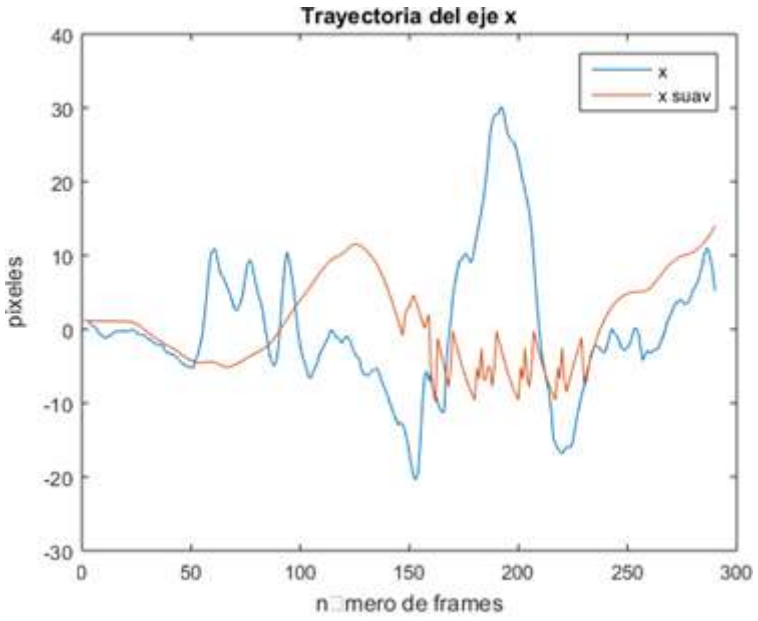


Figura 37: Trayectoria en el eje x. Antes y después de ser suavizada (imagen realizada por el autor, prueba 2)

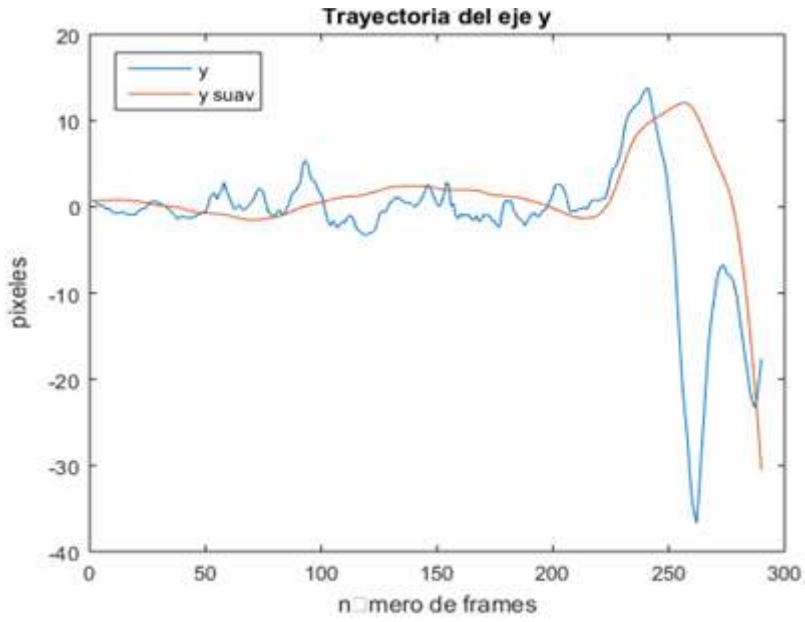


Figura 38: Trayectoria en el eje y. Antes y después de ser suavizada (imagen realizada por el autor, prueba 2)

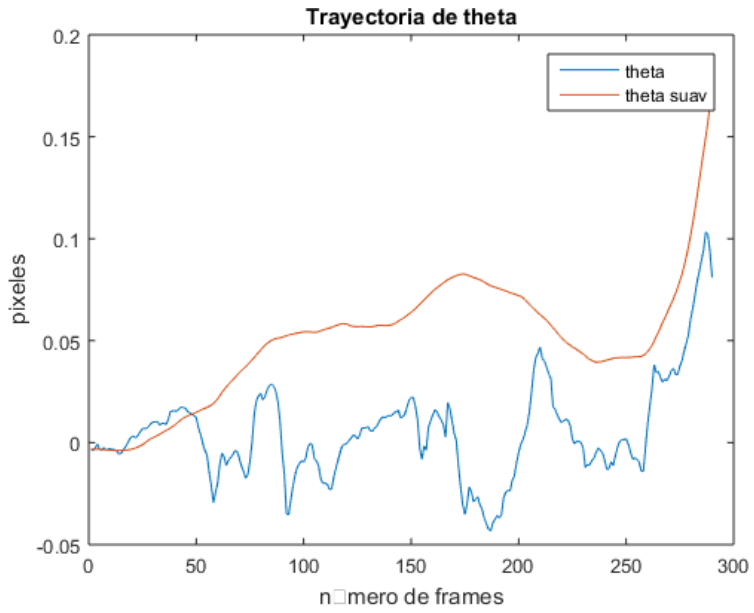


Figura 39: Trayectoria en el eje θ. Antes y después de ser suavizada (imagen realizada por el autor, prueba 2)

Con esto se observa que todas las trayectorias han sido suavizadas con la ayuda del algoritmo, vale la pena señalar que la trayectoria es una cantidad abstracta que no tiene necesariamente una relación directa con el movimiento inducido por la cámara, Lo importante es que la trayectoria puede ser suavizada, incluso si no tiene ninguna interpretación física.

Y finalmente para la Figura 40 y la Figura 41 se calcula nuevamente la transformada rígida al video ya estabilizado para observar el cambio que ha tenido.

En la Figura 40 se tienen las variaciones de la trayectoria expresadas en píxeles en los ejes (x, y) y comparada con Figura 35 existe la percepción de cambios menos bruscos, de igual manera es posible detectar que en la Figura 41 al ser comparada con la Figura 36, los movimientos son más armónicos.

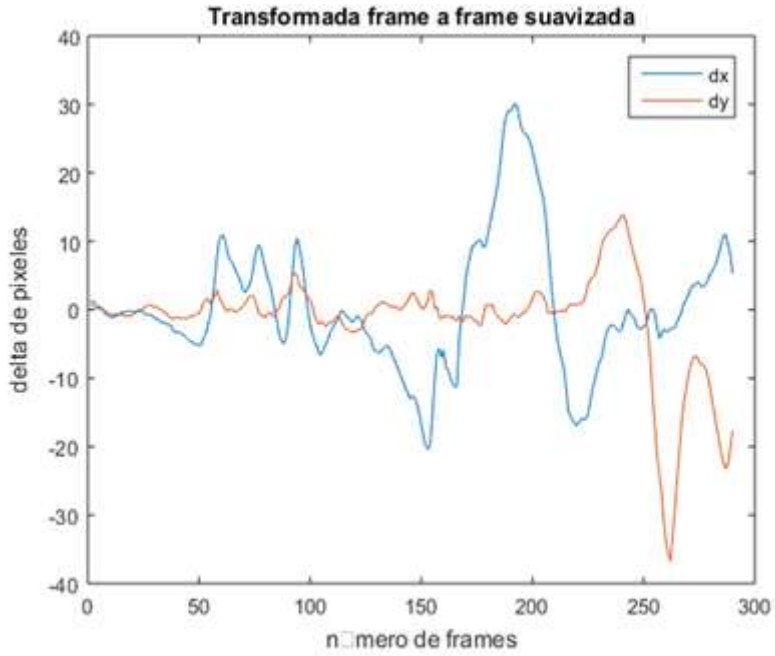


Figura 40: Transformada rígida en ejes (x, y)después de ser suavizada (imagen realizada por el autor, prueba 2)

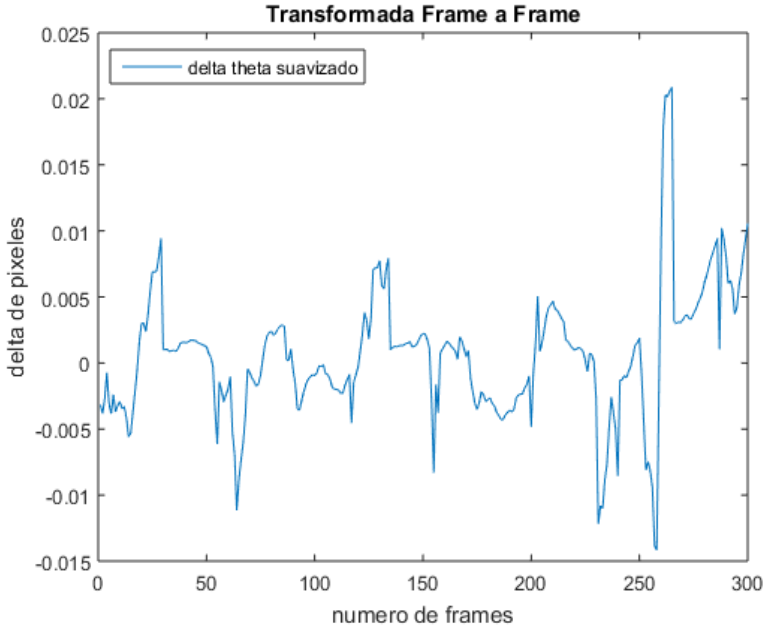


Figura 41: Transformada rígida en el eje θ, después de ser suavizada (imagen realizada por el autor, prueba 2)

Al igual que en la sección 4.1.1 se presentan capturas de pantalla pertenecientes al video del cual se extrajeron los datos para esta sección.

En la Figura 42, Figura 43 y Figura 44 se encuentran 8frames en secuencia, desde el frame 91 hasta el frame 99, esta vez con movimientos más bruscos y no solo en los ejes (x, y) sino también en el eje θ, donde la mayor variación observable se da a partir del frame 96, ya que el desplazamiento en el eje θ es muy alto, pero al igual que el ejemplo anterior el algoritmo contrarresta estos movimientos provocando que la escena se salga del marco de la pantalla, pero el objetivo permanezca prácticamente en las mismas coordenadas a lo largo de la secuencia.

Figura 42: Captura de frames segundo video estabilización 1 (imagen realizada por el autor)

Figura 43: Captura de frames segundo video estabilización 2 (imagen realizada por el autor)

Figura 44:Captura de frames segundo video estabilización 3 (imagen realizada por el autor)

4.2. Pruebas a la segunda etapa

En lo que concierne a la evaluación a la segunda etapa del algoritmo, lo que se hizo fue la muestra de dos videos, donde se comprueba que el algoritmo diseñado e implementado cumple con los objetivos de esta sección.

4.2.1. Primera prueba

En el enlace que se presenta a continuación (https://goo.gl/s9GdA4) se encuentra el video que fue el resultado de grabar la pantalla del dispositivo móvil mientras se ejecutaba la segunda etapa del algoritmo. En él se puede detallar que el reconocimiento a la región de interés es bastante robusto, a lo largo de la grabación se observan movimientos voluntarios para hacer que el rectángulo que se desea reconocer, visible en la Figura 22, cambie de coordenadas en la pantalla del móvil y a pesar de eso el algoritmo es capaz de hacer un seguimiento a esta zona y enmarcarla, para finalizar esta prueba se observa que entre los segundos 23 y 28 se aplican cambios de iluminación artificiales (en este caso con una linterna) para probar la eficiencia de la función implementada que los contrarresta, explicada en la sección 3.2, y se analiza que las variaciones son mínimamente perceptibles y el algoritmo aprende muy rápido, es decir toma los datos de iluminación y rápidamente hacen que estos pertenezcan al modelo de fondo, excluyéndolos como cambios significativos y haciendo que no se conviertan en falsos positivos. En la Figura 42 y Figura 43 y se ilustran tres frames consecutivos del video anteriormente mencionado, en el cual una muestra de cómo funciona el algoritmo de sustracción de fondo, en la primera parte se tiene la escena sin iluminación, y en ella se ve la región de interés enmarcada por un recuadro verde, a la segunda captura se le ha agregado iluminación, como se mencionó, genera algo de ruido mientras el algoritmo toma estos cambios como modelo de fondo y finalmente en la tercera escena se ve como el algoritmo rechaza dichos cambios, se puede comprobar porque ya no hay ruido como en la captura anterior.

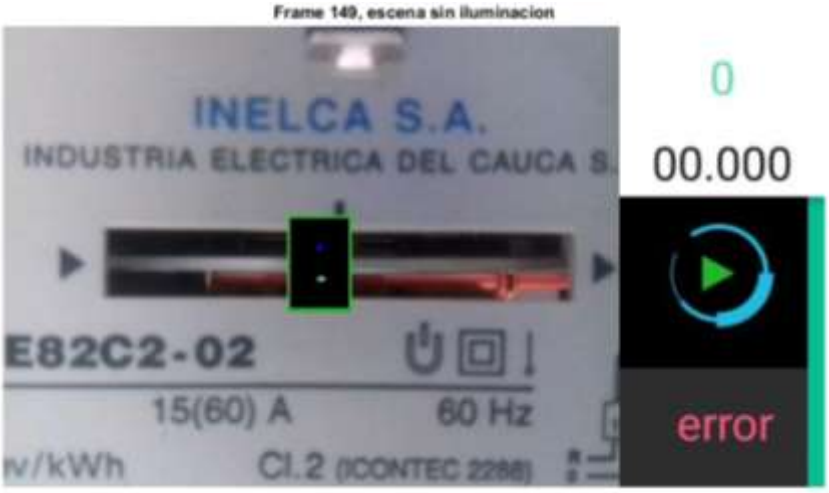


Figura 45:Captura de pantalla, rechazo a cambios de iluminación primer video, 1 (imagen realizada por el autor)

Figura 46:Captura de pantalla, rechazo a cambios de iluminación primer video, 2 (imagen realizada por el autor)

En la Figura 47 se presentan igualmente tres capturas, en esta se tiene que la linterna con la que se adicionan los cambios de iluminación artificiales es movida por lo ancho de la escena, en el frame 160 se tiene la región de interés sin ruido, al mover un poco la linterna esta genera el ruido que es indeseable, lo que se puede ver en el frame 161, pero igualmente el algoritmo rápidamente rechaza estos cambios y hace desaparecer el ruido, esto es visible en el frame 161.

Figura 47: Captura de pantalla, rechazo a cambios de iluminación primer video, 3 (imagen realizada por el autor)

4.2.2. Segunda prueba

Para esta segunda prueba al igual que en la primera se observan o descargan los videos en el siguiente enlace https://goo.gl/xNkBMG.

Con este video se pretenden hacer las mismas pruebas que en la sección (4.2.1), y al igual que en la anterior prueba se tiene detección robusta de la región de interés, en cuanto a los cambios de iluminación esta vez se realizaron con una linterna con mayor intensidad, estos cambios de iluminación provocados son observables entre los segundos 9 y 18, cabe notar que los cambios son tanto para aumentar como para disminuir la iluminación en la escena, y al igual que en la prueba realizada previamente estas variaciones no representan peligro para el resultado final ya que el algoritmo implementado es capaz de hacer que estos se vean disminuidos.

Al igual que en la prueba anterior en la se tiene que en el frame 57 aparece una escena sin iluminación, esto se puede ver en la Figura 48, en el frame que le sigue que es el 58 se adiciona luz a la escena, ilustrado en la Figura 49, como ya se explicó esta linterna tiene más intensidad lumínica, y es por esta razón que el ruido generado, visible en la dicho frame es mayor, pero eso no impide que el algoritmo realice un buen trabajo, como se observa en el frame 59, visible en la misma figura (Figura 49) el ruido ha sido eliminado satisfactoriamente.

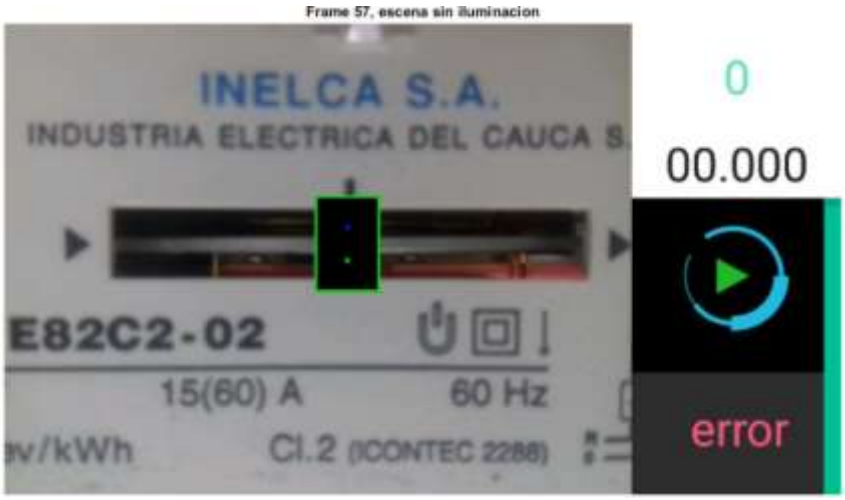


Figura 48: Captura de pantalla, rechazo a cambios de iluminación segundo video, 1 (imagen realizada por el autor)

Figura 49:Captura de pantalla, rechazo a cambios de iluminación segundo video, 2 (imagen realizada por el autor)

En la Figura 50 se tiene la misma prueba pero realizada inversamente, es decir que el lugar de agregar la iluminación artificial esta se quita, el frame 95 se puede observar que la iluminación está presente, el frame 96 se ha quitado totalmente la iluminación artificial, generando el mismo ruido que se genera como si se adicionara luz, pero el algoritmo nuevamente es capaz de repeler estos cambios y en el frame 97 se puede ver que nuevamente se tiene una imagen sin ruido en la región de interés.

Figura 50: Captura de pantalla, rechazo a cambios de iluminación segundo video, 3 (imagen realizada por el autor)

4.3. Prueba a la tercera etapa del algoritmo

Las pruebas que se realizan en esta sección no son solamente a la tercera etapa del algoritmo si no a su totalidad, para su realización no se hicieron necesarias conexiones físicas al contador (conectar cargas fantasmas), se adaptó un motor de corriente directa al tornillo sin fin del medidor, este se puede observar en la Figura 51.

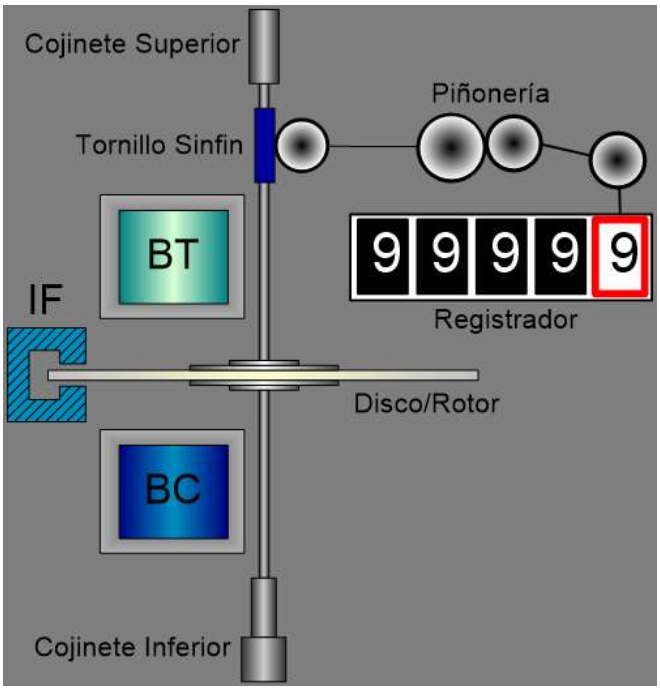


Figura 51: Mecanismo interno de un medidor (imagen tomada de la documentación de la CEO)

A su vez el motor de corriente directa está controlado por un dispositivo Arduino Uno y de esta manera se pueden regular las revoluciones del motor a través de variaciones del PWM, lo que permite de la misma manera controlar las revoluciones del Disco o Rotor del medidor y así tomar mediciones patrón que servirán de referencia para comprobar la robustez de la aplicación.

Las realizaciones de estas pruebas se hicieron bajo las siguientes condiciones:

Constante del medidor ISKRA tipo E82C2-02: 1000 imp/kWh

Corriente: 4 Amperios.

Voltaje 120 Voltios.

Los valores de corriente y voltaje son datos que se obtienen al realizar la prueba en fase baja, que es la fase usual que eligen los operarios para esta verificación.

Para calcular el tiempo patrón se eligieron tres valores de PWM, a estos se les midió el tiempo en el que tardaba en dar siete revoluciones, esto se hizo manualmente y diez veces para cada valor, después se promedió este tiempo. Los resultados se encuentran en la Tabla 5. Estos valores de PWM fueron elegidos a partir de que con un PWM alrededor del 25% se tiene un error aproximadamente de 0% según las ecuaciones (1) y (2) y se hicieron variaciones cercanas a este valor, de $\pm$6%, es decir valores de 19% y de 31%

	PWM 19%	PWM 25%	PWM 31%
Medición	Tiempo (s)	Tiempo (s)	Tiempo (s)
1	41.2	34.5	28.8
2	40.8	35.2	29.2
3	41.0	34.3	28.9
4	41.1	34.2	29.5
5	41.2	34.4	29.4
6	40.7	35.3	29.4
7	40.8	34.4	29.6
8	40.7	34.1	29.2
9	41.2	35.4	29.6
10	40.5	34.1	29.5
Promedio	40.92	34.59	29.31

Tabla 5: Mediciones de tiempo para diferentes valores de PWM (medida patrón)

Y en la Figura 52 se tiene la gráfica del promedio de estos tiempos medidos, contra los valores de PWM trazando una recta entre ellos, que representa al tiempo ideal al que se deben acercar las pruebas.

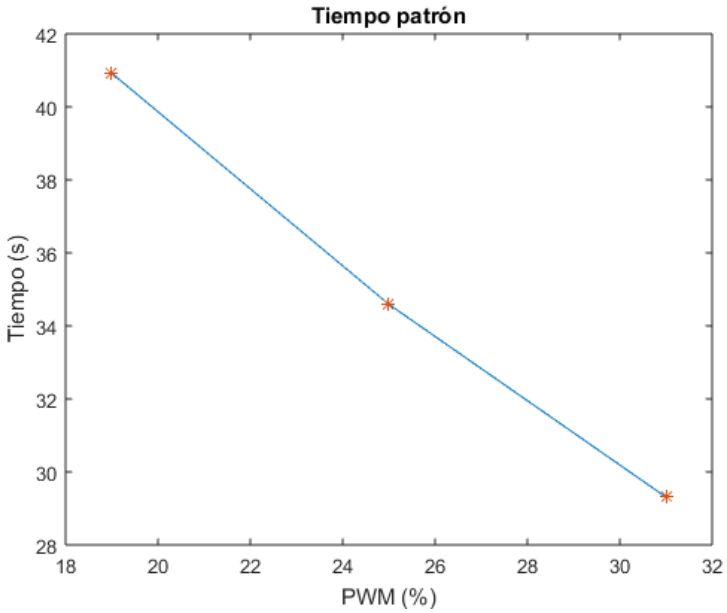


Figura 52: Aceptación de las pruebas realizadas (imagen realizada por el autor)

Lo siguiente es medir el tiempo con valores específicos de PWM, esto se hace de forma manual y con la aplicación y se comparan entre sí. Las pruebas que se hicieron con la aplicación móvil fueron sometidas a diferentes tipos de movimiento para comprobar que realmente contrarresta los movimientos involuntarios que pueda presentar el operario, los movimientos en las primeras cinco pruebas, es decir del PWM1 hasta PWM5, se hicieron bajo movimientos principalmente de escalamiento uniforme, y las siguientes cinco, es decir desde PWM6 hasta PWM10, se hicieron con movimientos de rotación. Los datos obtenidos se visualizan en la Tabla 6

	PWM1 (20%)	PWM2 (21%)	PWM3 (22%)	PWM4 (23%)	PWM5 (24%)	PWM6 (26%)	PWM7 (27%)	PWM8 (28%)	PWM9 (29%)	PWM10 (30%)
T. app (s)	39.75	38.96	37.96	37.09	36.02	34.25	33.06	31.56	31.07	30.52
T. cron (s)	40.42	38.89	38.15	37.55	36.25	34.26	33.19	32.05	30.70	30.25

Tabla 6: Comparación del tiempo medido por la aplicación versus el tiempo medido manualmente

Y en la Figura 53 se grafican estos valores obtenidos, el cuadrado de color negro representa el tiempo medido con la aplicación, mientras que la estrella de color azul representa el tiempo medido de manera manual, a simple vista se pueden

observar que los valores obtenidos mediante la aplicación, en su mayoría, se acercan más a los valores patrón.

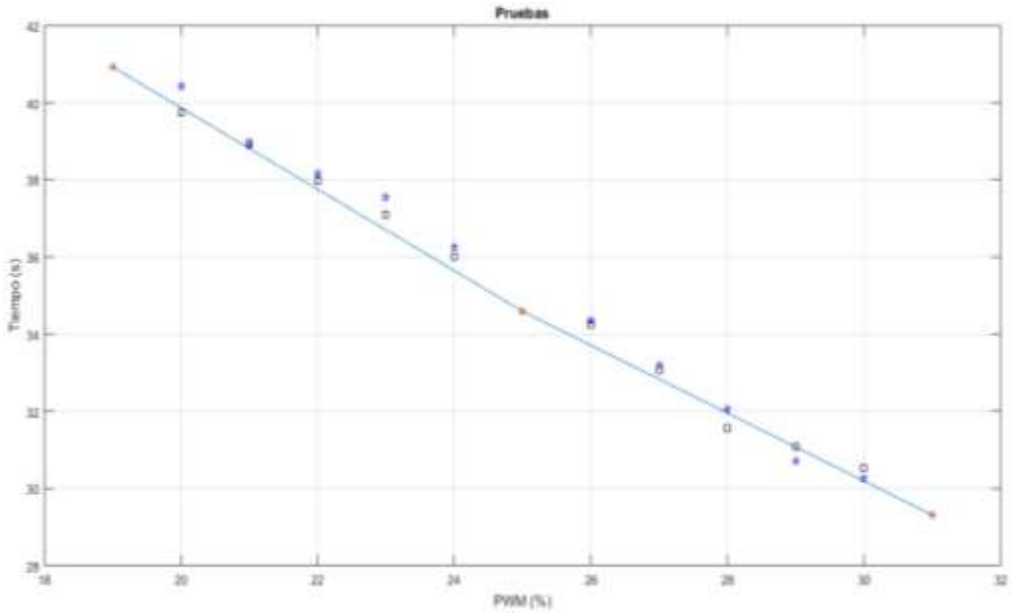


Figura 53: Comparación de los tiempos medidos manualmente y por la aplicación (imagen realizada por el autor)

4.4. Estimación temporal de las etapas del algoritmo

En esta sección hace el cálculo de cuánto tiempo le toma a cada etapa alcanzar el objetivo planteado localmente.

El cálculo temporal de cada etapa del algoritmo se llevó a cabo mediante el mismo IDE que se usó para implementar este mismo, es decir Android Studio, para esto se usó la función de java `System.nanoTime()`, la cual da un dato muy preciso del tiempo estimado de cada etapa de la función, ya que este dato es entregado en nanosegundos se debe realizar la conversión a segundos (al valor entregado por la función, dividirlo entre 1.000’000.000), los resultados obtenidos fueron los siguientes: primera etapa 0.029463 segundos, es decir que la aplicación se va a demorar este tiempo en hacer la comparación entre el frame actual y el frame anterior y hacer todos los cálculos explicados en la sección 3.1, para la segunda etapa el resultado fue de 0.009634 segundos, este es el tiempo en el que tarda el algoritmo en dibujar la región de interés y aplicar el sustractor de fondo consignado en la sección 3.2 y finalmente se tiene el tiempo del cálculo del error, es de 0.15452 segundos. Los anteriores tiempos presentados son muy bajos e indican un muy buen desempeño en todas las etapas del algoritmo.

características del computador

Marca	Toshiba
Modelo	C45-ASP4311FL
Procesador	Intel Core i5-3230M 2.6GHz (3.2GHz c/TB)
Memoria RAM	6 GB DDR3
Disco Duro	750GB 5400RPM

4.5. Interpretación de los resultados

Se calculó la desviación estándar de los datos obtenidos mediante la aplicación y manualmente, esto se hizo posible en parte gracias a la ayuda del programa Matlab, el cual permite ubicar un punto sobre la recta y saber su valor exacto, esto se hace para saber cuál es el tiempo ideal de cada valor de PWM, en la

	PWM1 (20%)	PWM2 (21%)	PWM3 (22%)	PWM4 (23%)	PWM5 (24%)	PWM6 (26%)	PWM7 (27%)	PWM8 (28%)	PWM9 (29%)	PWM10 (30%)
T. app (s)	39.75	38.96	37.96	37.09	36.02	34.25	33.06	31.56	31.07	30.52
T. cron (s)	40.42	38.89	38.15	37.55	36.25	34.26	33.19	32.05	30.70	30.25
T. patrón (s)	39.86	38.81	37.75	36.7	35.64	33.71	32.83	31.95	31.07	30.19
Desviación estándar app	0.054	0.074	0.104	0.194	0.190	0.269	0.114	0.195	0	0.160
Desviación estándar manual	0.280	0.039	0.199	0.424	0.305	0.274	0.180	0.050	0.185	0.030

Tabla 7: Comparaciones de desviaciones estándar

Se calculan los promedios de estas desviaciones estándar y se tiene:

Promedio desviación estándar, datos de la aplicación = 0.1354 y promedio de desviación estándar datos tomados manualmente = 0.1997, así podemos comprobar que la aplicación es más exacta en la toma de medidas, comparado a como se realizaría manualmente.

Otra cosa que se realizó para la validación de los datos fue graficar la correlación lineal que tenían los datos esperados contra los datos obtenidos, para este caso se correlacionan los tiempos, el obtenido con la aplicación y el obtenido manualmente.

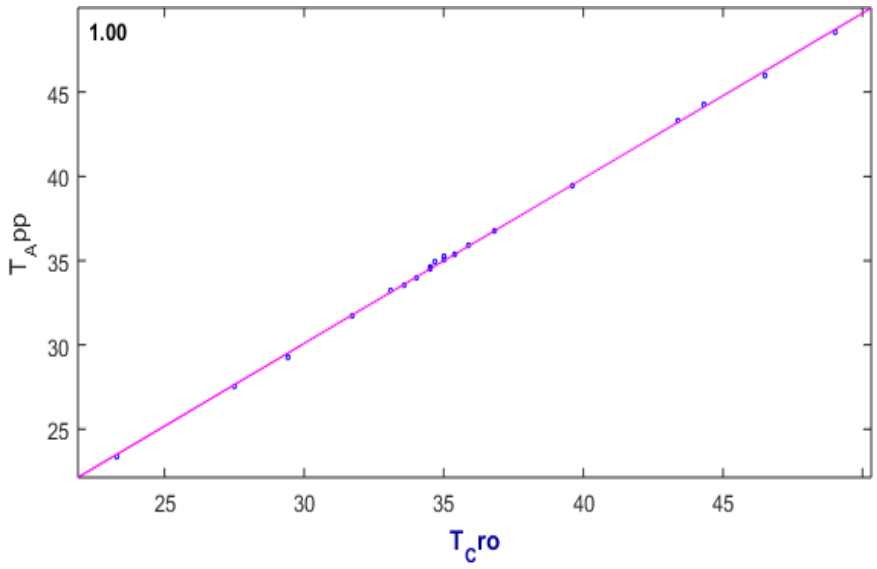


Figura 54: Correlación entre los tiempos calculados (imagen realizada por el autor)

En la Figura 54 se plasmaron estos datos y como se puede en la parte superior izquierda se tiene el valor de correlación lineal, el cual es igual a 1, lo que significa que existe una relación de dependencia total directa entre las variables. Es decir, si una de ellas aumenta (o disminuye), la otra aumenta (o disminuye) en igual proporción. Lo cual es bueno e indica un buen desempeño de la aplicación.

5. Conclusiones y trabajos futuros

En este capítulo se resumen las conclusiones a las que se han llegado después del desarrollo del trabajo que se realizó para satisfacer una necesidad de la empresa prestadora de servicio en el departamento del Cauca denominada CEO, de la misma manera se plantean trabajos que pueden llegar a mejorar este mismo.

5.1. Conclusiones

El reto principal del proyecto fue implementar el algoritmo diseñado con funciones que sean ejecutables en tiempo real, que no sobrecargue la aplicación, pero de igual manera que sean robustas para satisfacer los objetivos planteados, esto se llevó a cabo satisfactoriamente.

El algoritmo planteado puede ser de gran utilidad para el personal de la CEO ya que un dispositivo móvil es algo que tienen la mayoría de las personas en la actualidad o la empresa puede proporcionar al operario.

El escalamiento tecnológico facilita la aplicación de la prueba, ya que se ven disminuidos el número de dispositivos que se utilizan en ella, lo que al técnico encargado le da una mejor maniobrabilidad y esto puede llevar a que la prueba se aplique más rápido y por ende pueda cubrir más medidores en el mismo periodo de tiempo, caso útil cuando se realizan brigadas MACRO y se debe cubrir una mayor área que en las brigadas usuales.

Aunque los resultados de la aplicación fueron muy buenos, se ve algo lejana la utilización de esta misma en campo, ya que esta aplicación únicamente está diseñada para el tipo de medidores electromecánicos y con el valor de constante de medidor del ISKRA E8C2C2.

El desempeño del algoritmo puede variar dependiendo de las características del dispositivo móvil que se use.

5.2. Trabajos futuros

Para que la aplicación sea más completa se le podría adicionar una función que envíe los resultados de la prueba, junto con los datos del medidor y la persona propietaria a la impresora portátil con la que cuentan los operarios de la CEO y expuesta en la Tabla 2.

Se puede desarrollar una aplicación que realice la prueba de exactitud a los medidores electrónicos, que en general sigue el mismo procedimiento, pero en lugar de contar las revoluciones del disco cuenta los pulsos generados por un diodo led.

Se puede abarcar un número más amplio de contadores electromecánicos si el valor de la constante del medidor es introducido y no un valor fijo.

Se puede mejorar el desempeño del algoritmo haciendo que este no sea aplicado en tiempo real, es decir, grabar el video del funcionamiento del medidor, después cargar este video con la aplicación desarrollada y que esta entregue el resultado del error.

Acoplar la aplicación desarrollada con la aplicación implementada en el trabajo de grado “Sistema de lectura numérica de medidores de energía eléctrica para la prueba de dosificación de la Compañía Energética de Occidente” en la cual se asiste el desarrollo de la prueba de dosificación

REFERENCIAS

[1] F. Alegria and A. Serra, “Automatic calibration of analog and digital measuring instruments using computer vision,” *IEEE Trans. Instrum. Meas.*, vol. 49, no. 1, pp. 94–99, 2000.

[2] S. L. Pang and W. L. Chan, “Computer vision application in automatic meter calibration,” *Fourtieth IAS Annu. Meet. Conf. Rec. 2005 Ind. Appl. Conf. 2005.*, vol. 3, pp. 1731–1735, 2005.

[3] Q. Li, Y. Fang, Y. He, F. Yang, and Q. Li, “Automatic reading system based on automatic alignment control for pointer meter,” in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, 2014, pp. 3414–3418.

[4] G. Andria, G. Cavone, L. Fabbiano, N. Giaquinto, and M. Savino, “Automatic Calibration System for Digital Instruments Without Built-in Communication Interface,” pp. 857–860, 2009.

[5] O. Chang, E. Pruna, M. Pilatasig, I. Escobar, and L. Mena, “Calibration of residential water meters by using computer vision,” *CHILECON 2015 - 2015 IEEE Chil. Conf. Electr. Electron. Eng. Inf. Commun. Technol. Proc. IEEE Chilecon 2015*, pp. 351–356, 2016.

[6] Y. Yang *et al.*, “EAST-AIA deployment under vacuum: Calibration of laser diagnostic system using computer vision,” *Fusion Eng. Des.*, vol. 112, pp. 563–568, 2016.

[7] R. Lu and M. Shao, “Sphere-based calibration method for trinocular vision sensor,” *Opt. Lasers Eng.*, vol. 90, no. October 2016, pp. 119–127, 2017.

[8] P. A. Belan, S. A. Araujo, and A. F. H. Librantz, “Segmentation-free approaches of computer vision for automatic calibration of digital and analog instruments,” *Measurement*, vol. 46, no. 1, pp. 177–184, 2013.

[9] C. Zheng, S. Wang, Y. Zhang, P. Zhang, and Y. Zhao, “A robust and automatic recognition system of analog instruments in power system by using computer vision,” *Measurement*, vol. 92, pp. 413–420, 2016.

[10] S. Chang, Y. Zhong, Z. Quan, Y. Hong, J. Zeng, and D. Du, “A Real-time Object Tracking and Image Stabilization System for Photographing in Vibration Environment using OpenTLD Algorithm,” *Adv. Robot. its Soc. Impacts (ARSO), 2016 IEEE Work.*, pp. 0–4, 2016.

[11] L. Pettazzi, E. Fedrigo, R. Muradore, P. Haguenauer, and L. Pallanca, “Improving the accuracy of interferometric measurements through adaptive vibration cancellation,” in *2015 IEEE Conference on Control and*

Applications, CCA 2015 - Proceedings, 2015, no. July 2014, pp. 95–100.

[12] B. H. Chen *et al.*, “Improved global motion estimation via motion vector clustering for video stabilization,” *Eng. Appl. Artif. Intell.*, vol. 54, pp. 39–48, Sep. 2016.

[13] G. Puglisi and S. Battiato, “A robust image alignment algorithm for video stabilization purposes,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1390–1400, Oct. 2011.

[14] O. K. O. Kwon, J. S. J. Shin, and J. P. J. Paik, “Edge based adaptive Kalman filtering for real-time video stabilization,” *2006 Dig. Tech. Pap. Int. Conf. Consum. Electron.*, pp. 75–76, 2006.

[15] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, “SIFT features tracking for video stabilization,” in *Proceedings - 14th International conference on Image Analysis and Processing, ICIAP 2007*, 2007, pp. 825–830.

[16] K. Feng, H. Yonghua, and Z. Huaxiong, “Video stabilization based on multi-scale local color invariants,” in *Proceedings of the International Conference on Networking and Distributed Computing, ICNDC*, 2014, pp. 65–69.

[17] D. G. Lowe, “Object recognition from local scale-invariant features,” *Proc. Seventh IEEE Int. Conf. Comput. Vis.*, vol. 2, no. [8, pp. 1150–1157, 1999.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[19] C. T. Jianbo Shi, “Good Features to Track.pdf,” *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 593–600, 1994.

[20] J. V. C. I. R and Y. G. Lee, “Novel video stabilization for real-time optical character recognition applications q,” *J. Vis. Commun. Image Represent.*, vol. 44, pp. 148–155, 2017.

[21] L. Di Stefano, F. Tombari, and S. Mattoccia, “Robust and accurate change detection under sudden illumination variations,” *ACCV’07 Work. Multi-dimensional Multi-view Image Process. Tokyo, Nov., 2007 MM-P-02*, pp. 103–109, 2007.

[22] S. Kumar and J. Sen Yadav, “Video object extraction and its tracking using background subtraction in complex environments,” *Perspect. Sci.*, 2016.

[23] Z. Zeng, J. Jia, Z. Zhu, and D. Yu, “Adaptive maintenance scheme for codebook-based dynamic background subtraction,” *Comput. Vis. Image Underst.*, vol. 152, pp. 58–66, 2016.

[24] G. Gemignani and A. Rozza, “A Robust Approach for the Background Subtraction Based on Multi-Layered Self-Organizing Maps,” vol. 25, no. 11, pp. 5239–5251, 2016.

[25] K. . Saran and S. G, “Traffic Video Surveillance : Vehicle Detection and

Classification," *2015 Int. Conf. Control. Commun. Comput. India*, no. November, pp. 516–521, 2015.

[26] C. Y. Jang, S. J. Kang, and Y. H. Kim, "Adaptive contrast enhancement using edge-based lighting condition estimation," *Digit. Signal Process. A Rev. J.*, vol. 58, pp. 1–9, 2016.

[27] Electrificadora del Meta S.A., "Medición de energía eléctrica," vol. 1, p. 5, 2013.

[28] ICICONTEC. (1996). Norma técnica colombiana, "Norma técnica colombiana 5900." 2011.

[29] I. (1996). N. técnica Colombiana, "Norma técnica colombia 4856." 2015.

[30] C. P. Dueñas, "Apuntes de visión artificial," *Univ. Madrid*, vol. 1, pp. 11–32, 2008.

[31] R. Szeliski, "Computer Vision : Algorithms and Applications," *Computer (Long. Beach. Calif).*, vol. 5, p. 832, 2010.

[32] R. Castillo, J. Hernández, E. Inzunza, and J. Torres, "Procesamiento Digital de Imágenes Empleando Filtros Espaciales.," *Iiis.Org*, 2013.

[33] J. G. Proakis and D. Manolakis, "Digital Signal Processing." p. 1033, 1996.

[34] W. K. Pratt, "Digital image process," in *Digital Image Processing*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007, pp. i–xix.

[35] K. Y. Huang, Y. M. Tsai, C. C. Tsai, and L. G. Chen, "Video stabilization for vehicular applications using surf-like descriptor and KD-tree," *Proc. - Int. Conf. Image Process. ICIP*, no. 5, pp. 3517–3520, 2010.

[36] L. Araneda and M. Figueroa, "A compact hardware architecture for digital image stabilization using integral projections," *Microprocess. Microsyst.*, vol. 39, no. 8, pp. 987–997, 2015.

[37] D. G. Lowe, "Object recognition from local scale-invariant features," *Proc. Seventh IEEE Int. Conf. Comput. Vis.*, vol. 2, no. [8, pp. 1150–1157, 1999.

[38] H. Bay and A. Ess, "Speeded-Up Robust Features (SURF)," vol. 110, pp. 346–359, 2008.

[39] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Procedings Alvey Vis. Conf. 1988*, pp. 147–151, 1988.

[40] C. T. Jianbo Shi, "Good Features to Track," *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 593–600, 1994.

[41] "Affine Transformation - MATLAB & Simulink." [Online]. Available: https://es.mathworks.com/discovery/affine-transformation.html. [Accessed: 14-Aug-2017].

[42] J. J. Gibson, "of the Environment Are What It," 1977.

[43] F. Raudies, “Optic flow,” *Scholarpedia*, vol. 8, no. 7. p. 30724, 2013.

[44] A. Barriga-Rivera and G. J. Suaning, *Digital Image Processing using Matlab*, vol. 2011. 2011.

[45] C. Platero, “Procesamiento Morfológico,” *Madrid Univ.*, pp. 171–198, 2012.

[46] M. Piccardi, “Background subtraction techniques: a review,” *2004 IEEE Int. Conf. Syst. Man Cybern. (IEEE Cat. No.04CH37583)*, vol. 4, pp. 3099–3104, 2004.

[47] B. Tamersoy, “Background Subtraction,” *Human-Computer Interact.*, vol. 18, pp. 106–114, 2009.

[48] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 2, no. 2, p. 28–31 Vol.2, 2004.

[49] Z. Zivkovic and F. Van Der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction,” *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[50] Y. Wang, Z. Hou, K. Leman, and R. Chang, “Real-Time Video Stabilization for Unmanned Aerial Vehicles,” *Conf. Mach. Vis. Appl.*, no. April, pp. 336–339, 2011.

[51] N. Friedman and S. Russell, ““Image Segmentation in Video Sequences: A Probabilistic Approach,” *Thirteen. Conf. Uncertain. Artif. Intell.*, pp. 175–181, 1991.

[52] S. Suzuki and K. be, “Topological structural analysis of digitized binary images by border following,” *Comput. Vision, Graph. Image Process.*, vol. 30, no. 1, pp. 32–46, 1985.

Anexos

Anexo 1: Instalación de la librería OpenCV en Android Studio y ejemplo de cálculo del vector de flujo óptico entre dos fotogramas

Para la realización de esta guía se cuenta con que ya se tiene instalado el software Android Studio en su versión 2.3.3, en caso de no tenerlo se puede descargar del siguiente Link https://developer.android.com/studio/index.html, lo siguiente es instalar OpenCV, en este caso se contó con la versión 3.1.0 para la versión de Android, se puede descargar desde el siguiente enlace http://opencv.org/releases.html, se elige la opción “Android pack” de la versión ya mencionada como se muestra en la Figura 55. Lo que se va a descargar es un archivo Zip que contiene una carpeta de archivos, esta se debe pegar en un sitio fácil de memorizar, en mi caso se encuentra en la Disco local D (D:\)

Figura 55: OpenCV para Android (imagen realizada por el autor).

Para la instalación de la librería en el proyecto que se está realizando se debe generar un nuevo proyecto en Android Studio y seguir los pasos como se muestran desde la Figura 56 hasta la,

En la Figura 56 se da el nombre que va a recibir la aplicación y la ubicación del archivo, después de esto tal y como se observa en la Figura 57 se selecciona la

versión de Android MÍNIMA, en la que la aplicación puede ser soportada, en este caso se seleccionó la versión 4.4.

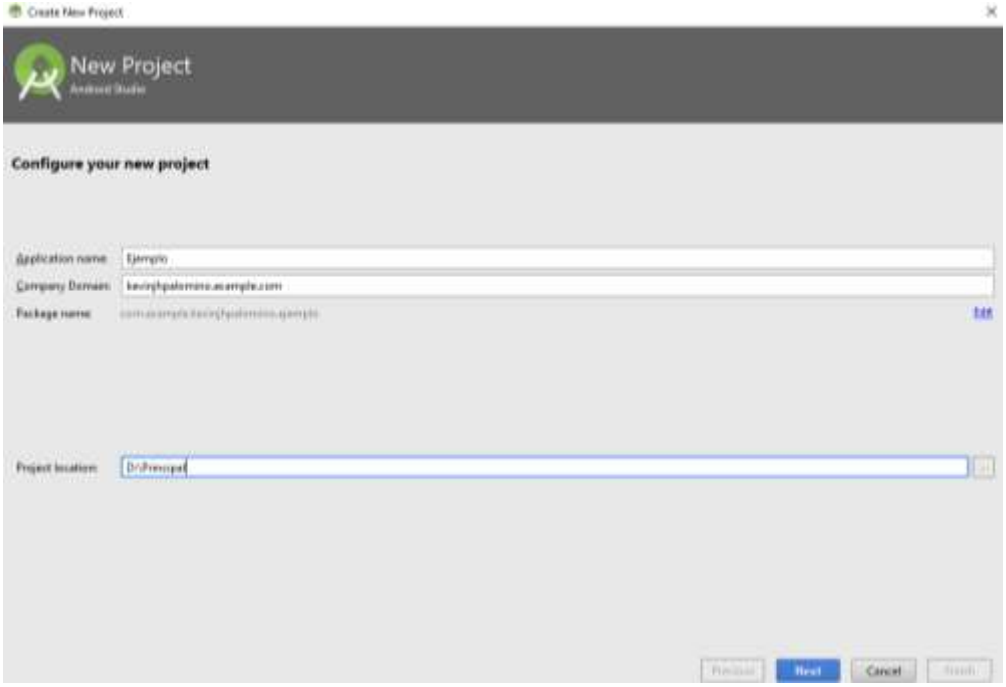


Figura 56: Creación de un nuevo proyecto en Android Studio (imagen realizada por el autor).

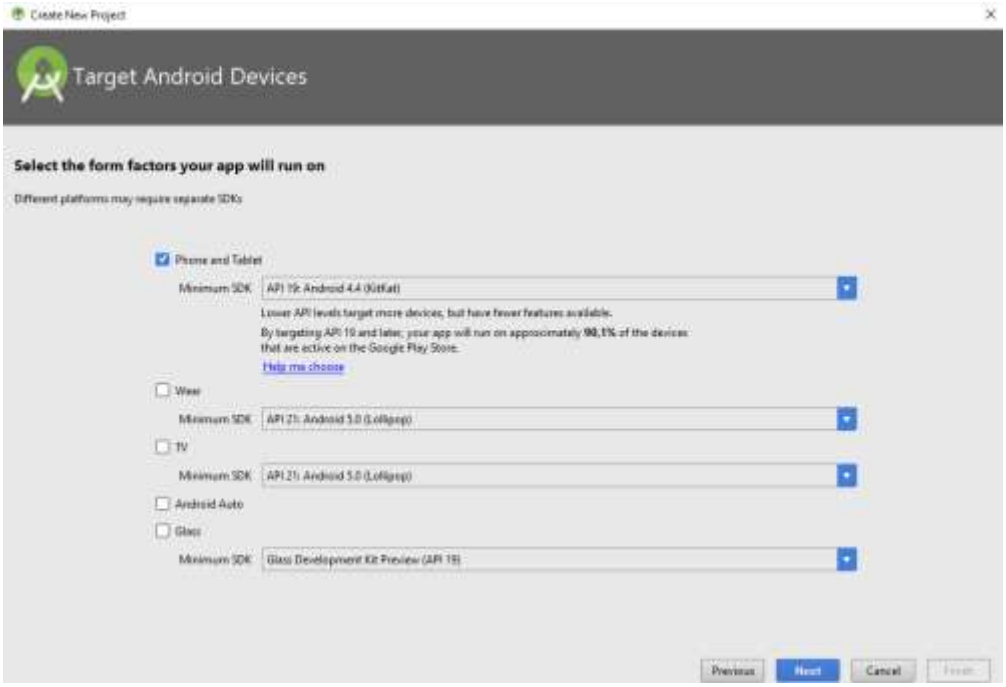


Figura 57: Seleccionar versión Android (imagen realizada por el autor).

Ahora se selecciona con que tipo de poryeto se quiere iniciar y para el caso del ejemplo se elige un proyecto vacío

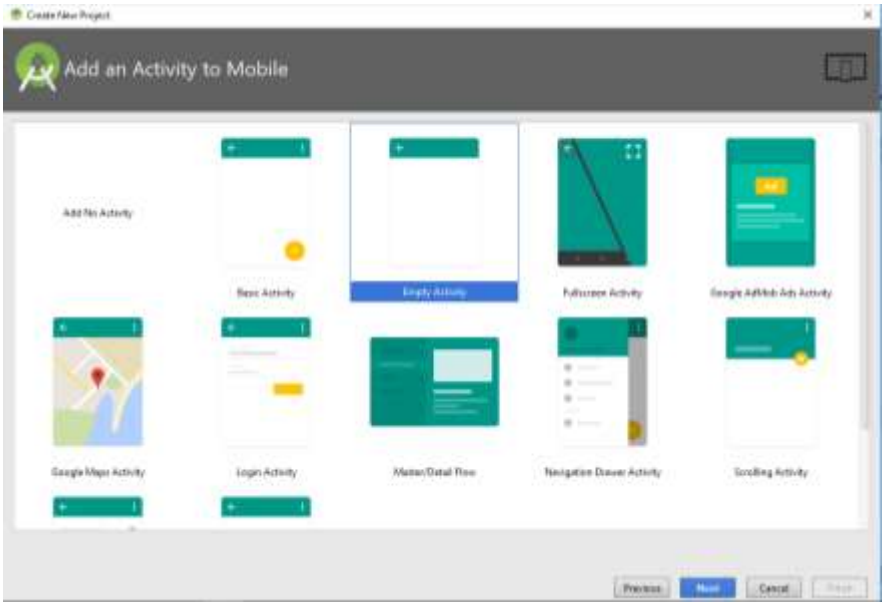


Figura 58: Tipo de proyecto (imagen realizada por el autor).

El siguiente paso es darle un nombre al archivo java, que es donde va a ir la parte lógica del código que se va a crear.

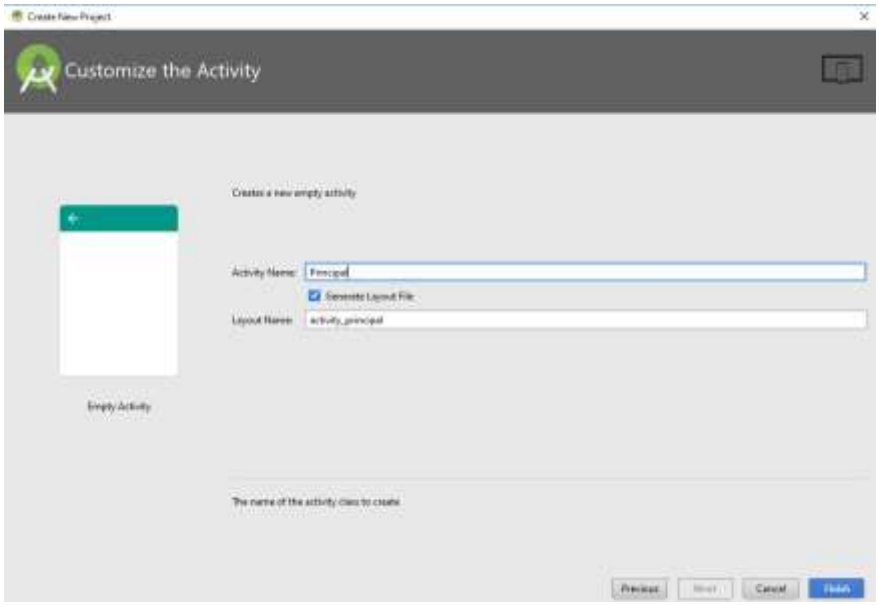


Figura 59: Nombre del archivo .java (imagen realizada por el autor)

Una vez hecho este procedimiento se le da clic en “Finish” y se crea el proyecto, después, tal y como lo muestra la Figura 60, se requiere importar el módulo de la librería OpenCV.

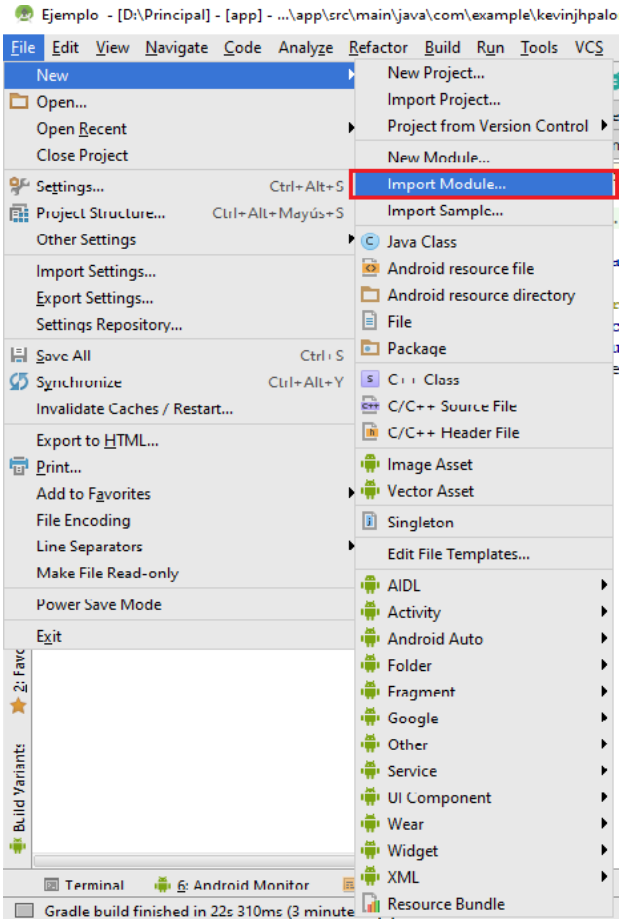


Figura 60: Importación del módulo (imagen realizada por el autor).

Una vez en esta ventana se selecciona la carpeta con el nombre de “java” que se encuentra en el paquete de archivos de OpenCV para Android descargados anteriormente, esta carpeta se encuentra ubicada en la siguiente dirección para mi caso. D:\OpenCV-android-sdk3.1.0\sdk

Esto se ilustra en la Figura 61, una vez se selecciona la carpeta mencionada se da clic en “Ok” → “Next” y finalmente “Finish”

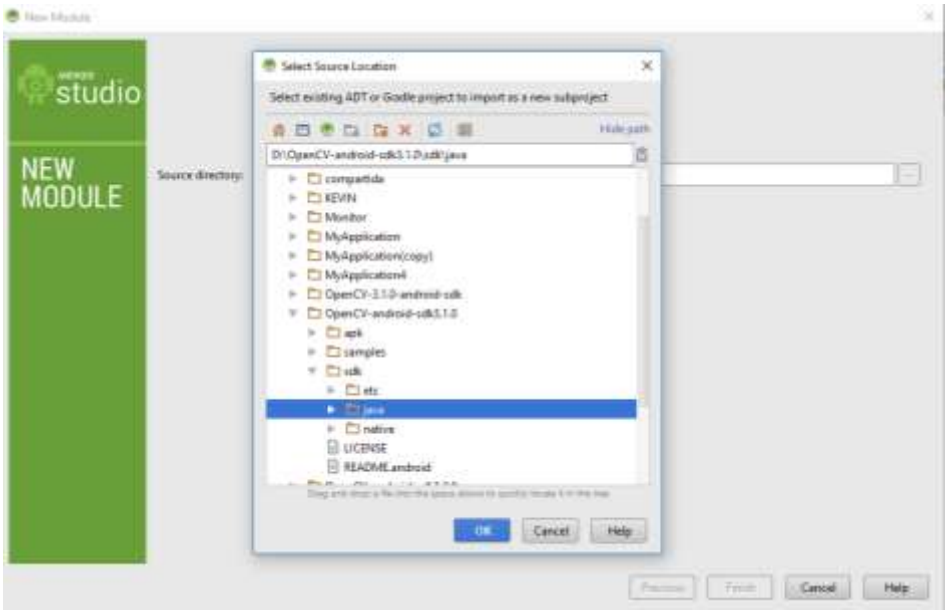


Figura 61: Selección del archivo de la librería(imagen realizada por el autor).

Realizado ya este paso van a aparecer múltiples errores en el proyecto, pero no son de preocuparse ya que no se ha completado el procedimiento completo. El paso que prosigue a esto es ir a la sección “Project Structure” que se encuentra en la pestaña “File”, acción que despliega una ventana como la que se ilustra en la Figura 62

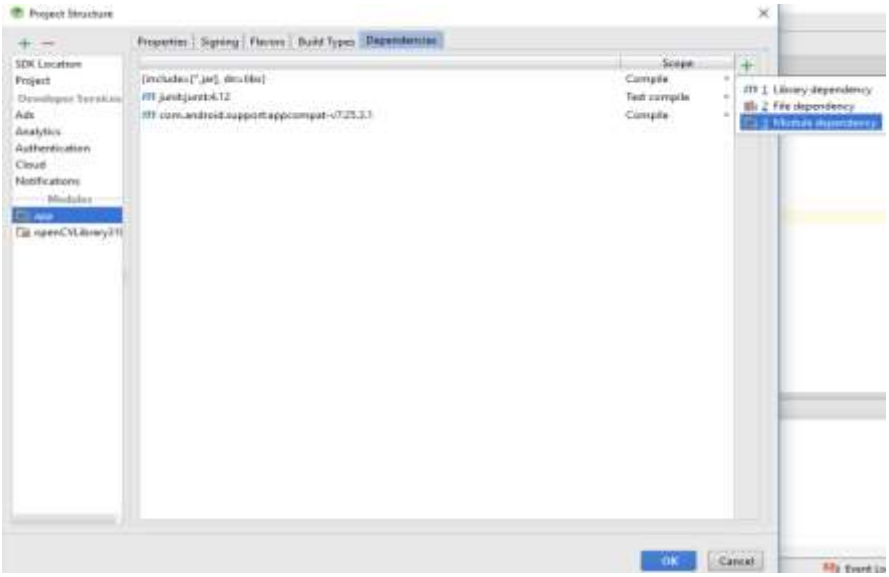


Figura 62: Estructura del proyecto (imagen realizada por el autor)

Ahí se selecciona la pestaña llamada “app” ubicada en el lado inferior izquierdo de la pantalla, después elige otra pestaña llamada “Dependencies”, se le da clic en el símbolo de suma (+) y en “Module dependency” y se selecciona el único archivo que contiene la pestaña llamado “openCVLibrary310” , clic en “Ok” y en “Ok” en este punto, en los archivos del proyecto que se ubican en la parte izquierda de la pantalla ya se debe contar con la Librería de OpenCV, tal como en la Figura 63

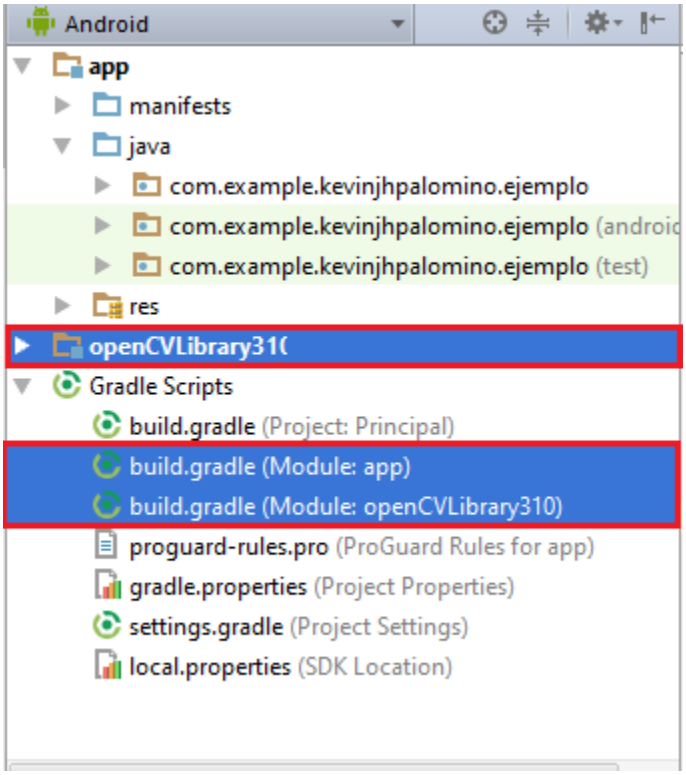


Figura 63: Librería OpenCV implementada (imagen realizada por el autor).

Ahora se deben copiar las librerías el OpenCV en el proyecto creado, para eso vamos al paquete que se descargó anteriormente, en mí caso D:\OpenCV-android-sdk3.1.0\sdk\nativey se copia la carpeta llamada “libs”, después se debe ir a la carpeta donde se creó el proyecto y pegarla en la siguiente ruta para mí caso D:\Principal\app\src\mainy pegarla, también se debe cambiar el nombre de la carpeta por “jniLibs” quedando como se muestra en la Figura 64.

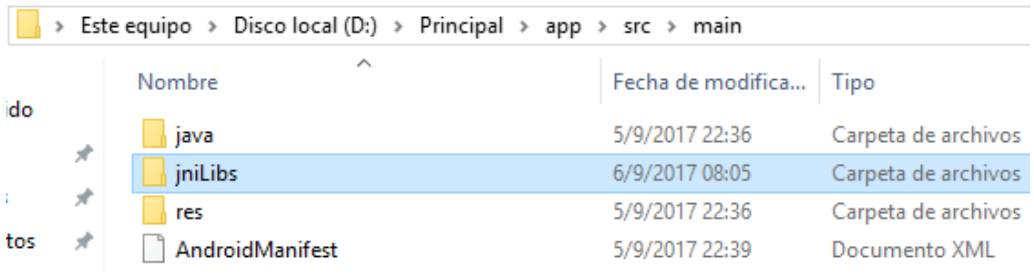


Figura 64: Librerías pegadas al proyecto (imagen realizada por el autor).

Y únicamente queda modificar los archivos “Gradle” para que el archivo compile correctamente que se encuentran encerrados en el segundo recuadro de la Figura 63. A estos archivos se les cambian los siguientes valores con los que aparecen tanto en la Figura 65 como en la Figura 66.

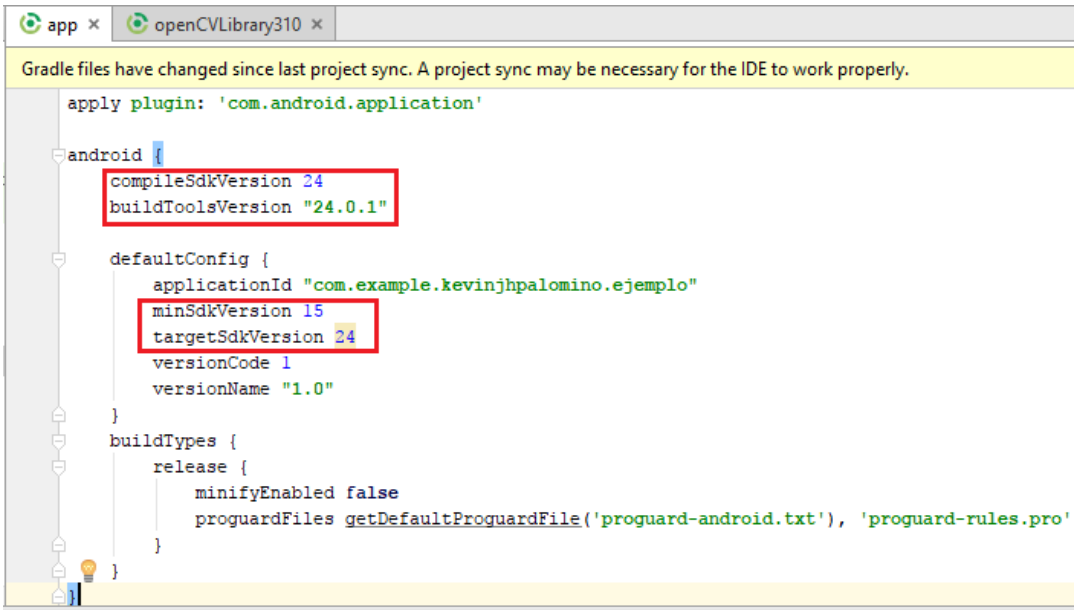


Figura 65: Primer archivo gradle (imagen realizada por el autor).

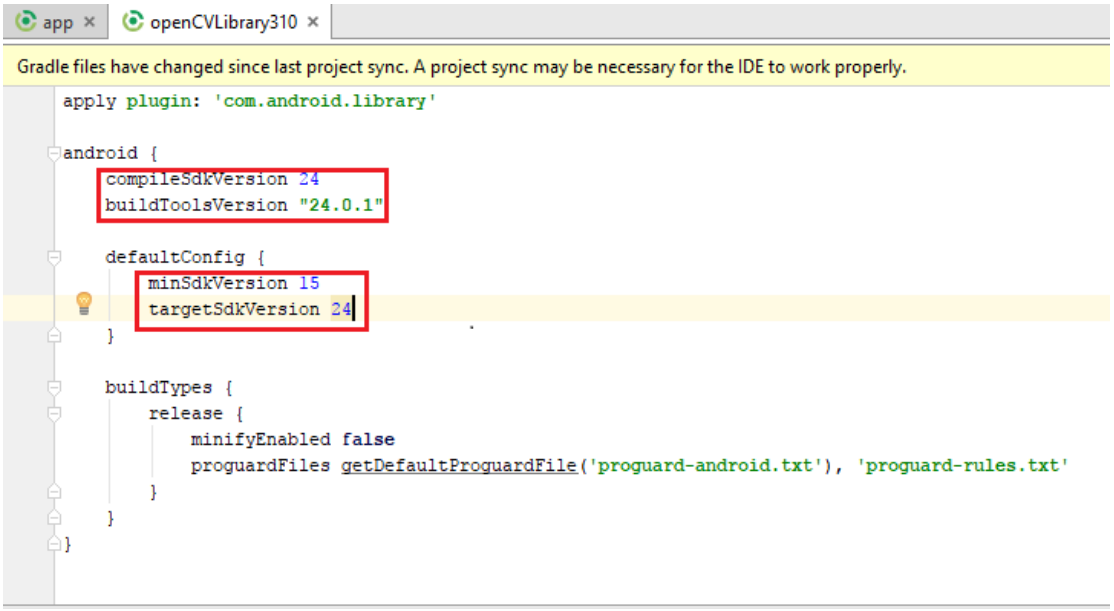


Figura 66: Segundo archivo gradle (imagen realizada por el autor).

Una vez realizado esto se sincroniza el proyecto, dándole clic en el logo  ubicado en la parte superior del proyecto, realizado esto para comprobar que la librería fue correctamente agregada al proyecto se observa algo como lo mostrado en la Figura 67 localizado en la parte inferior de la ventana.

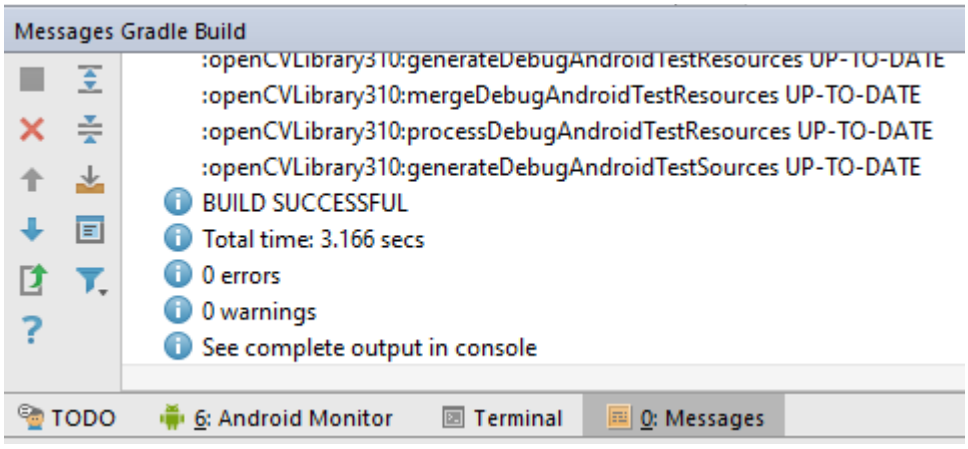


Figura 67: Mensaje de sincronización (imagen realizada por el autor).

Ya realizado esto lo que hay que hacer es dirigirse al siguiente enlace https://goo.gl/XXdhZjy descargar la carpeta que contiene tres archivos, que son los que se encuentran en la Figura 68.

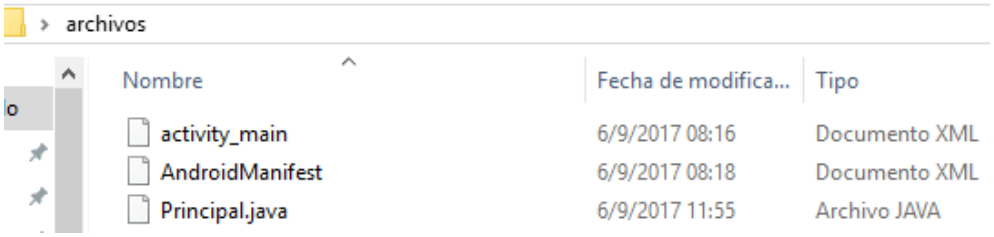


Figura 68: Archivos de flujo óptico (imagen realizada por el autor).

Lo que se debe hacer inicialmente es abrir el archivo llamado “Principal.java” con un editor de texto que puede ser el Bloc de notas o Notepad + +, copiar todo el código desde la línea dos y pegarlo en el archivo .java del proyecto (en el segundo recuadro de la Figura 69) también desde la línea dos.

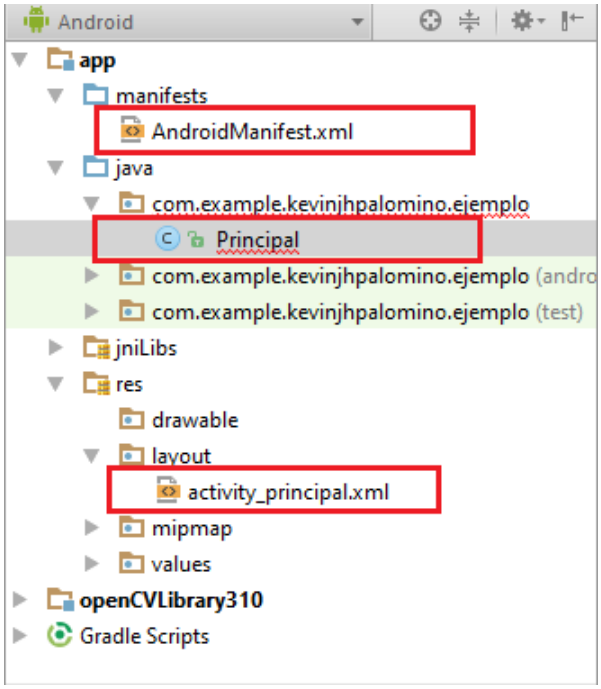


Figura 69: Archivo .java del proyecto (imagen realizada por el autor).

Después hacer lo mismo con el archivo llamado “AndroidManifest.xml” y únicamente copiar las cinco líneas que se muestran señaladas en la Figura 70 y

pegarla en el archivo “AndroidManifest.xml” señalado en el primer recuadro de la Figura 69.

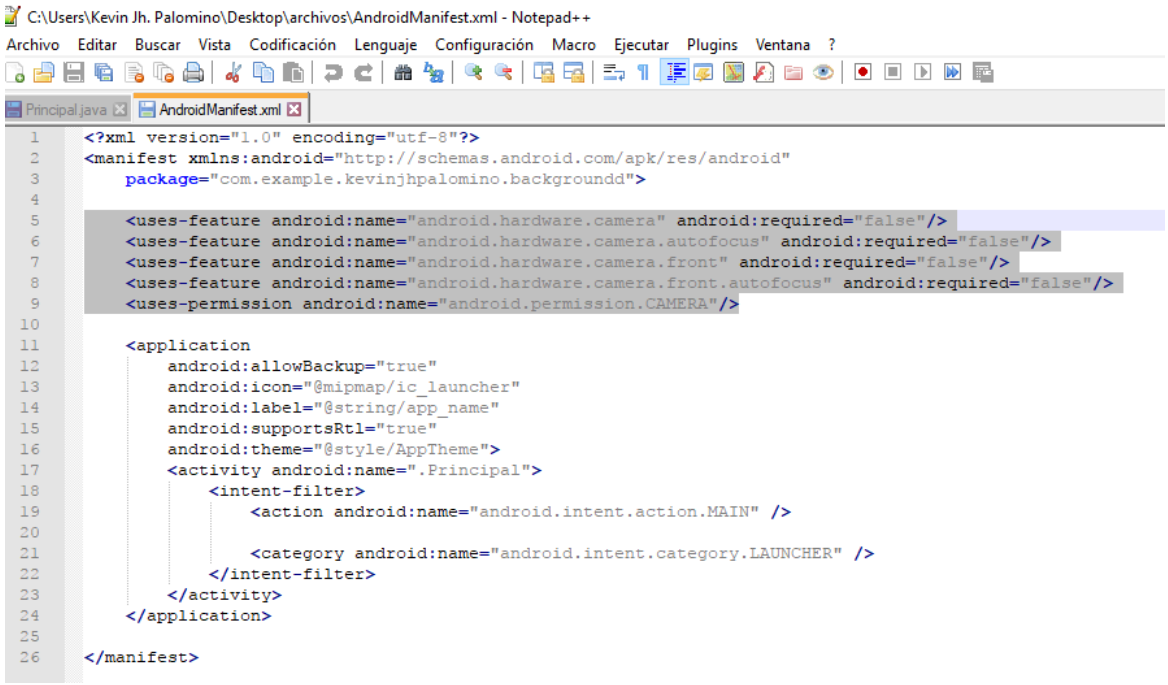


Figura 70: Android manifesf archivo .xml (imagen realizada por el autor).

Finalmente abrir el archivo “activity_main.xml” y copiar todas las líneas de código y pegarlas en el archivo que aparece en el tercer recuadro de la Figura 70.

El paso final es instalar la aplicación en el celular (asegurarse de tener la depuración activada), se debe conectar el celular al computador por medio del cable de datos, presionar la tecla ▶ ubicada en la parte superior de la ventana del proyecto, se desplegará una ventana con el nombre del dispositivo conectado, se selecciona y se presiona “Ok”. En este momento la aplicación empezará a ser instalada y cuando termine este procedimiento se abrirá automáticamente la aplicación, en ella observa la detección de los puntos característicos y al mover el objeto que se esté enfocando o la cámara del celular se genera el seguimiento a cada punto (siempre y cuando cada punto sea detectado en dos frames consecutivos), el punto rojo señala donde estaba el punto en el frame anterior y el verde indica donde se encuentra actualmente dicho punto, la línea azul que conecta los dos puntos es nombrada vector de flujo óptico.

Anexo 2: Manual de usuario de la aplicación

Descripción de la aplicación

La aplicación diseñada ha sido nombrada “AccTest”, este es un diminutivo de Accuracy Test, que es español significa prueba de exactitud, esta aplicación está diseñada para móviles con sistema operativo Android versión 4.4 KitKat, sin embargo, puede ser usada en versiones superiores de Android, esta aplicación permite realizar de forma automática la prueba de exactitud de que es realizada por los miembros de la CEO dentro de las verificaciones en sitio, estas brigadas pueden ser PQR o campañas MACRO, el logo que se diseñó para la aplicación es el que se muestra en la Figura 71.

Figura 71: Logo de AccTest (imagen realizada por el autor)

Requerimientos mínimos

El celular utilizado para el desarrollo de esta aplicación fue un Samsung Galaxy Grand Prime modelo SM-G531H y sus características esenciales son las siguientes:

Pantalla: 5’’, 540 x 960 píxeles.

Procesador: Snapdragon 410 1.2 GHz.

Memoria RAM: 1 GB.

Peso: 156 g.

Procesador gráfico: ARM Malí-400 MP2.

Cámara trasera: 8 Megapíxeles.

Cámara frontal: 5 Megapíxeles.

Este dispositivo móvil puede ser observado en la Figura 72.

Figura 72: Samsung Galaxy Grand Prime (imagen tomada de la Web)

Aunque no es un celular muy potente fue suficiente para la realización de este proyecto y se esperan mejores rendimientos en celulares con mejores características.

Para iniciar se debe tener la aplicación instalada y además se debe instalar OpenCVManager ya que el algoritmo desarrollado usa funciones de la librería de OpenCVy se hace necesaria su instalación, esta aplicación que puede ser encontrada en la Google PlayStore. El logo de esta aplicación es mostrado en la Figura 73.

Figura 73: OpenCV Manager + AccTest (imagen realizada por el autor)

Ya solo queda usar la aplicación desarrollada, para eso se da clic en el logo de AccTest.

Inicialmente se encontrará una pantalla de presentación con el nombre de la empresa para la cual se desarrolló la aplicación y también su logo, como se muestra en la

Figura 74: Nombre y logo de la CEO (Imagen tomada de la Web).

Después de esto, la aplicación automáticamente pasará a la pantalla de inicio, la cual contiene tres botones que redirigen a tres nuevas ventanas, dicha pantalla está ilustrada en la Figura 75.

Figura 75: Pantalla de inicio AccTest (imagen realizada por el autor)

Dando clic en el botón de ayuda se abre una ventana en la queel operario encontrará unos pasos para la correcta realización de la prueba (Figura 76).

Figura 76: Ventana de ayuda (imagen realizada por el autor)

También en el menú principal se encuentra el botón de Acerca de, este contiene la información básica de la aplicación, quienes fueron sus desarrolladores, cual fue el propósito de su realización, cual es la persona encargada de la supervisión del desarrollo del, etc. Esto está ilustrado en la Figura 77

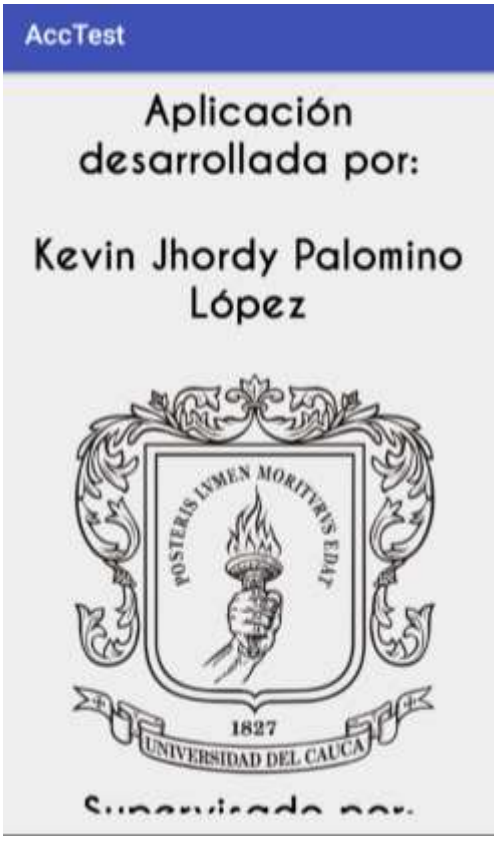


Figura 77: Ventana "Acerca de" (imagen realizada por el autor)

Y finalmente, en el botón de menú principal se encuentra el botón de realizar prueba, siendo este el más importante de la aplicación, al dar clic sobre este botón se abre automáticamente la cámara trasera del móvil, en este paso se debe tener conectado el medidor de energía para realizar la prueba y verificar que este esté funcionando correctamente, cuando eso ya sea asegurado se debe enfocar la cámara del móvil al rotor del medidor electromecánico, la aplicación automáticamente encontrará la zona de interés, y para que el algoritmo empiece a contar las revoluciones el operario debe presionar el botón  sobre dicha zona, como se muestra en la Figura 78.

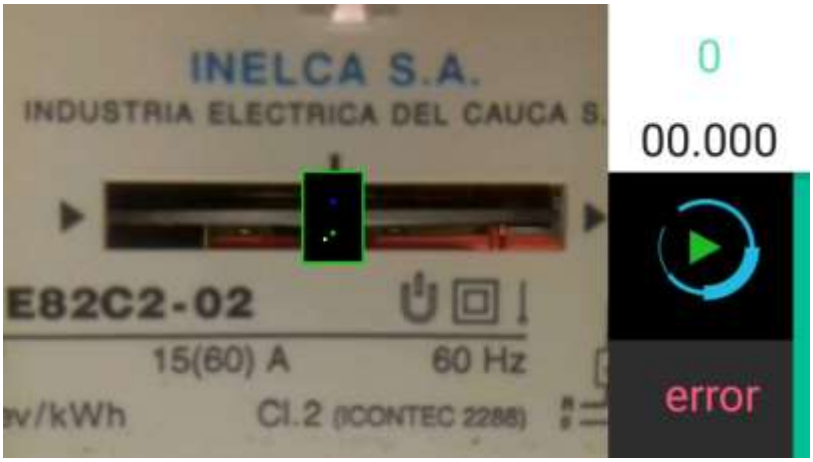


Figura 78: Prueba de exactitud (imagen realizada por el autor)

Cuando la prueba termine, automáticamente en la parte inferior derecha de la pantalla se muestra el error de medición que tiene el contador de energía, con lo que se puede deducir si el medidor es apto o no para seguir con su funcionamiento cabe recordar que para que el medidor sea apto el error no debe sobrepasar el $\pm 5\%$ como se explica en la sección 2.3, un ejemplo del error calculado, con los demás datos se puede observar en la Figura 79

Figura 79: Error calculado en un medidor (imagen realizada por el autor)

Y con esto se da por terminada la guía de uso de AccTest.