

**Instituto Tecnológico de Costa Rica**  
**Escuela de Computación**  
Programa de Maestría en Computación

**Un análisis comparativo de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* para la detección automática de granos de café en imágenes**

**(A comparative analysis of the algorithms *Fast Radial Symmetry Transform* and *Hough Transform* for the automatic counting of coffee grains in images)**

Tesis para optar al grado de Magíster Scientiae en Computación

**Autor:**  
**Marco León Sarkis**

**Profesor Asesor:**  
**Luis Alexánder Calvo Valverde**

Agosto, 2017

## Resumen

En este trabajo se presenta una estrategia que aporta en la solución de un problema presentado por el Instituto del Café de Costa Rica (ICAFFE). Esta organización cuenta con una serie de imágenes de granos de café y desea encontrar una manera automática, mediante visión por computador, de realizar el conteo de los granos con el fin de incrementar la eficiencia en el proceso de estimación de la producción.

Se propone una estrategia de detección de granos de café mediante la combinación de los algoritmos *Fast Radial Symmetry Transform*[8] y *Hough Transform*[19]. Posteriormente, se realiza la incorporación de la estrategia en el proceso de detección de granos de la herramienta P-TRAP[13] con el propósito de aumentar la exactitud en la detección de granos de café existentes a partir de imágenes de granos tomadas desde un dispositivo móvil en un ambiente no controlado y en la que los granos de café no son removidos de su ambiente natural.

Asimismo, se realiza un análisis comparativo de las detecciones realizadas por la versión de P-TRAP desarrollada en este trabajo, en donde se combinan ambos algoritmos, y las detecciones realizadas por los algoritmos *Fast Radial Symmetry Transform*[8] y *Hough Transform*[19] individualmente. Para este análisis se determina el número de granos existentes en cada imagen previamente. Posteriormente, se ejecuta el proceso de detección de granos sobre cada imagen y se toma registro de la cantidad de granos de café existentes detectados por cada algoritmo. Finalmente, se realiza un análisis detallado sobre los resultados obtenidos.

## Abstract

In this work we present a strategy that contributes to the overall solution of a problem presented by the Costa Rica Coffee Institute (ICAFFE). ICAFFE owns a set of coffee grains images and needs to find an automatic way, through computer vision, to detect and count the number of grains in each image in order to increase the efficiency in the process of estimating yield.

A strategy to detect coffee grains in images is proposed, by combining the algorithms *Fast Radial Symmetry Transform*[8] and *Hough Transform*[19]. Then, this strategy is incorporated in the grain detection process of P-TRAP[13], an open-source tool, to increase the precision in the detection of existing coffee grains. The images are taken with a mobile device in a non-controlled environment in which the grains are not pulled off their natural environment.

Likewise, a comparative analysis is done between the P-TRAP version developed in this study and both algorithms running individually. The number of existing grains in an image is determined manually. Then, the cherry detection process is executed over each image and results are collected. Finally, a detailed analysis is done over the results obtained.

### ACTA DE APROBACIÓN DE TESIS

#### APROBACIÓN DE LA TESIS

Con fundamento en lo que establece el artículo 20.4.1.3 del "Manual de Normas y Procedimientos para optar por el título de "MAGISTER ACADÉMICO EN COMPUTACIÓN" el Instituto Tecnológico de Costa Rica (ITEC), certifica con la aprobación de la tesis de grado:

**"Un análisis comparativo de los algoritmos Fast Radial Symmetry Transform y Hough Transform para la detección automática de granos de café en imágenes"**

Nombre

Apellido

Método de Votación

Fecha de Aprobación

#### TRIBUNAL EXAMINADOR

<input checked="" type="radio"/> APROBADO	<input type="radio"/> REPROBADO
<input type="radio"/> RECOMIENDA	<input type="radio"/> NO RECOMIENDA

Máster Luis Alexander Calvo Valverde  
Profesor Asesor

Doctor Esteban Meneses Rojas  
Profesor Lector

Máster Saul Calderón Ramírez  
Profesional Externo

Dr. Roberto Cortés Morales  
Coordinador/Programa De Maestría

Agosto, 2017

# Tabla de Contenido

1	<b>Introducción</b> . . . . .	11
2	<b>Marco teórico</b> . . . . .	13
2.1	Visión por computador . . . . .	13
2.2	Segmentación de una imagen . . . . .	15
2.3	Identificación de objetos . . . . .	15
2.4	P-TRAP . . . . .	16
2.5	Algoritmo Fast Radial Symmetry Transform . . . . .	19
2.6	Espacio de Hough . . . . .	21
2.6.1	Representación de líneas en el espacio de Hough . . . . .	21
2.6.2	Representación de círculos en el espacio de Hough . . . . .	23
2.6.3	Algoritmo . . . . .	25
3	<b>Descripción de la investigación</b> . . . . .	26
3.1	Planteamiento del problema . . . . .	26
3.2	Descripción general de la investigación . . . . .	30
3.3	Trabajos relacionados . . . . .	32
3.4	Hipótesis . . . . .	35
3.5	Objetivos . . . . .	35
3.5.1	Objetivo general . . . . .	35
3.5.2	Objetivos específicos . . . . .	35
3.6	Resumen de experimentos . . . . .	36
3.6.1	Declaración del problema . . . . .	36
3.6.2	Factores . . . . .	37
3.6.3	Variables de respuesta . . . . .	37

3.6.4	Recolección de datos . . . . .	38
3.6.5	Análisis estadístico . . . . .	38
3.7	Detalles de experimentos . . . . .	39
3.8	Ambiente de desarrollo . . . . .	40
3.9	Conjunto de datos . . . . .	40
3.10	Implementación de los algoritmos . . . . .	41
3.10.1	Implementación de Fast Radial Symmetry Transform	41
3.10.2	Implementación de Hough Transform . . . . .	43
3.10.3	Implementación en P-TRAP . . . . .	45
3.11	Selección de parámetros . . . . .	48
3.11.1	Parámetros de Fast Radial Symmetry Transform . .	48
3.11.2	Parámetros de Hough Transform . . . . .	48
3.12	Alcance del proyecto . . . . .	49
<b>4</b>	<b>Resultados . . . . .</b>	<b>50</b>
4.1	Prueba de hipótesis . . . . .	51
4.1.1	Prueba de Lilliefors . . . . .	51
4.1.2	Prueba de Kruskal-Wallis . . . . .	52
4.1.3	Prueba de pares . . . . .	53
4.2	Análisis de resultados . . . . .	54
<b>5</b>	<b>Conclusiones y trabajo futuro . . . . .</b>	<b>56</b>
5.1	Conclusiones . . . . .	56
5.2	Trabajo Futuro . . . . .	59
	<b>Bibliografía</b>	<b>60</b>
	<b>Apéndice A Selección de parámetros</b>	<b>63</b>
1	Tabla de selección de parámetros para cada imagen . . . . .	64
	<b>Apéndice B Resultados de experimentos</b>	<b>65</b>
1	Tabla de resultados para Hough Transform . . . . .	66
2	Tabla de resultados para Fast Radial Symmetry Transform . . . . .	67

3	Tabla de resultados para P-TRAP . . . . .	68
4	Tabla de resultados para Hough Transform + Fast Radial Symmetry Transform . . . . .	69
<b>Apéndice C Conjunto de datos y código fuente</b>		<b>70</b>
1	Imágenes y código fuente . . . . .	71

# Lista de Tablas

1	Resultados de experimentos . . . . .	50
---	--------------------------------------	----



# Lista de Figuras

1	Ejemplo de una panícula . . . . .	16
2	Ejemplo de P-TRAP en funcionamiento . . . . .	17
3	Ejemplo de conversión . . . . .	17
4	Grano sin ruido . . . . .	18
5	Grano delimitado . . . . .	18
6	Ejemplo de <i>pixel afectado-negativamente</i> y <i>pixel afectado-positivamente</i> [8] . . . . .	20
7	Ejemplo de caras [8] . . . . .	20
8	Ejemplo de líneas que pasan por el punto $(x_0, y_0)$ . . . . .	21
9	Ilustración de líneas en el espacio paramétrico a partir de dos puntos en el espacio de la imagen. . . . .	22
10	Ejemplo de curva sinusoidal en el espacio paramétrico. . . . .	23
11	Ejemplo de imagen con un círculo. . . . .	24
12	Ejemplo de imagen con un círculo en el espacio paramétrico. . . . .	24
13	Aplicación del algoritmo Canny para detección de bordes. . . . .	25
14	Imágenes de granos en formato RGB . . . . .	27
15	Imágenes de granos en formato binario . . . . .	28
16	Ejemplo de detección de granos en P-TRAP . . . . .	30
17	Ejemplo de interacción de módulos de P-TRAP . . . . .	32
18	Opción agregada en P-TRAP . . . . .	45
19	Pantalla para definir parámetro <i>modo</i> . . . . .	46
20	Resultado utilizando <i>Fast Radial Symmetry Transform</i> y <i>Hough Transform</i> . . . . .	47

21 Resultado utilizando P-TRAP . . . . . 47

22 Gráfico Q-Q de la métrica f1-score . . . . . 51

23 Resultado de la prueba Lilliefors . . . . . 52

24 Resultado de la prueba Kruskal-Wallis . . . . . 52

25 Resultado de la prueba de pares post-hoc Nemenyi . . . . . 53

26 Diagrama de caja para las mediciones f1-score . . . . . 55

# 1 Introducción

El café es una de las materias primas que más se comercializan en el mundo. Millones de personas dependen directa o indirectamente de la producción y venta del café para poder subsistir. En el mercado mundial, el café se caracteriza por precios volátiles y cambios en los niveles de producción, los cuales impactan en forma directa los ingresos y la subsistencia de los agricultores que lo cultivan. El café sigue siendo el producto tropical que más se comercializa en todo el mundo y se produce en más de 50 países en transición y desarrollo; ofrece un medio de vida a 25 millones de agricultores y sus familias y representa hasta el 50% de los ingresos de exportación de algunos países [32].

En Costa Rica la comercialización de café se encuentra en manos del sector privado. Sin embargo, el estado mantiene la supervisión y control por medio del Instituto del Café de Costa Rica (ICAFFE), en cuya Junta Directiva están representados todos los actores que intervienen en la actividad cafetalera.

ICAFFE es una institución pública de carácter no estatal, fundada en 1933 como rectora de la caficultura costarricense. Uno de los objetivos principales de ICAFFE es investigar y desarrollar tecnología agrícola e industrial [16]. Guiados por este objetivo, ICAFFE busca hacer uso de la tecnología para resolver problemas y mejorar procesos en el área de la producción agrícola. Uno de estos problemas es el conteo de granos de café en imágenes. Esta tarea se realiza de manera manual y se requiere de tiempo y personal. Realizar el conteo de granos manualmente en una imagen requiere de un esfuerzo visual considerable. En este caso, el ojo debe hacer un mayor esfuerzo para enfocar pequeños detalles en la imagen. Después de un tiempo realizando esta tarea, la musculatura del ojo empieza a manifestar cansancio y como consecuencia, da paso a posibles errores en el conteo de granos de café.

En este trabajo de investigación se busca hacer uso de la visión por computadora

para aportar en la solución a este problema de conteo de granos de café en un ambiente no controlado. Se aporta el estudio y análisis estadístico de los algoritmos Fast Radial Symmetry Transform y Hough Transform en la detección de granos de café y se realiza la implementación de ambos algoritmos en la herramienta de código abierto P-TRAP.

En la sección 2 se presenta el marco teórico. Posteriormente, en la sección 3, se expone la descripción, el planteamiento del problema de investigación, objetivos generales y específicos. Seguidamente se presenta la metodología que se utilizó en esta investigación y el detalle de los experimentos realizados. El alcance es presentado en la sección 3. Finalmente, en la sección 5 se presentan los resultados de la investigación, potenciales mejoras y trabajo futuro.

## 2 Marco teórico

Uno de los sentidos más importantes de los seres humanos es la visión. Ésta es empleada para obtener la información visual del entorno físico. De hecho, se calcula que más del 70% de las tareas del cerebro son empleadas en el análisis de la información visual [14].

A principios del siglo XIX, en el año 1824, el científico francés Nicéphore Niepce obtuvo las primeras imágenes fotográficas, inéditas. La fotografía más antigua que se conserva es una reproducción de la imagen conocida como *Vista desde la ventana en Le Gras*, obtenida en 1826 con la utilización de una cámara oscura y una placa de peltre recubierta en betún [15].

Desde la aparición de las primeras imágenes fotográficas, se ha intentado extraer características físicas de las imágenes. La *fotogrametría* dió sus primeros pasos desde imágenes capturadas en globos. La astronomía avanzó enormemente con el análisis de imágenes recibidas por los telescopios. El análisis de radiografías transformó la medicina y así sucedió en una gran cantidad de áreas [15].

Por los años de 1980 estas técnicas confluyen y dan luz a un cuerpo de conocimiento propio. La revolución de la electrónica, con las cámaras de vídeo CCD y los microprocesadores, junto con la evolución de las Ciencias de la Computación hace que sea factible la *Visión por Computador* [14].

### 2.1 Visión por computador

La visión por computador es una disciplina científica que utiliza distintos métodos para adquirir, procesar, analizar y comprender imágenes del mundo real [2].

Hay muchas tecnologías que utilizan la visión por computador, entre ellas: reconocimiento de objetos, detección de eventos, reconstrucción de una escena y

restauración de imágenes.

Este trabajo se enfocará específicamente en la detección de objetos. En visión por computador, la detección de objetos es la tarea para encontrar e identificar objetos en una imagen [2]. Los humanos reconocemos una multitud de objetos en imágenes con poco esfuerzo, a pesar del hecho que la imagen del objeto puede variar un poco en diferentes puntos de vista, en diferentes tamaños o escalas e incluso cuando están trasladados o rotados. Sin embargo, no es posible que una máquina realice esta tarea con la facilidad con la que es realizada por los humanos y por el contrario es una tarea compleja computacionalmente.

En la industria podemos encontrar una serie de aplicaciones de los diferentes métodos de reconocimiento de objetos. A continuación se presenta una lista de algunas aplicaciones:

- Android Eyes - Object Recognition
- Detección de caras
- Control de calidad en procesos de manufactura
- Reconocimiento óptico de caracteres
- Marca de agua digital
- Parqueo automático de vehículos

El reconocimiento de objetos puede realizar un aporte valioso en el área de la agronomía. El análisis de imágenes de granos y semillas es una parte importante de diferentes procesos. Por ejemplo, incremento de la producción, aumento del entendimiento sobre el cultivo, predicción en una fase temprana de cual va a ser la producción final, elaboración de análisis genéticos, mejora del proceso de selección, entre muchos otros. Para realizar dicho reconocimiento de objetos es importante definir la diferencia entre segmentación y detección en imágenes. A continuación se

explican estos conceptos, seguidamente se presenta una herramienta que permite realizar este tipo de reconocimiento en imágenes de granos.

## 2.2 Segmentación de una imagen

En visión por computador, la segmentación de una imagen es un proceso en el cual se particiona una imagen digital en múltiples segmentos con el objetivo de simplificar la representación original de la imagen en otra más significativa y fácil de analizar [33]. Típicamente, el proceso de segmentación de imágenes es utilizado para localizar objetos y bordes en imágenes.

Los algoritmos de segmentación se basan en dos propiedades básicas de los valores del nivel de gris: discontinuidad o similitud entre los niveles de gris de píxeles vecinos. En el caso de la discontinuidad se busca segmentar la imagen a partir de los cambios grandes en los niveles de gris entre los píxeles. Las técnicas que utilizan las discontinuidades como base son la detección de líneas, bordes o puntos aislados. Por el contrario, la similitud busca segmentar la imagen mediante la agrupación de los píxeles que tienen características similares. Algunas técnicas que usan esto son la umbralización o el crecimiento de regiones [34].

## 2.3 Identificación de objetos

La identificación de objetos es la tarea de encontrar automáticamente objetos en una imagen. Los humanos detectamos y reconocemos objetos en imágenes con extrema facilidad, inclusive si los objetos sufren variaciones de forma, tamaño, localización, color, textura, brillo o están parcialmente obstruidos [35]. Cuando se trabaja en un problema de identificación de objetos en imágenes se sigue un proceso de cinco pasos explicado a continuación [35]:

1. Adquisición de la imagen: capturar la escena del mundo real a través de sensores y digitalizarla para su procesamiento, almacenamiento y transmisión.

2. Preprocesamiento: aplicar técnicas de mejoramiento de contraste, reducción de ruido, realce de características, entre otras. De modo que la imagen se adecúe para los siguientes pasos.
3. Segmentación: aislar los objetos de interés de la imagen.
4. Extracción de rasgos: describir numéricamente la naturaleza de los objetos segmentados como su forma, color y textura.
5. Clasificación: asignar una clase o categoría a cada objeto de la imagen basado en sus rasgos.

## 2.4 P-TRAP

En los cultivos, el tamaño y la forma de los granos son dos de las características más importantes que influyen la producción [17]. Por ejemplo, las panículas de arroz varían considerablemente en el número y orden de las ramas, alargamiento del eje, y la forma y tamaño del grano. El análisis manual de estas características consume bastante tiempo y los resultados son poco confiables. Sin embargo, el análisis cuantitativo y cualitativo de los rasgos en las panículas de arroz es esencial para el entendimiento de la diversidad de una panícula. La figura 1 es un ejemplo de una panícula en la cual podemos observar 3 aristas y un total de 20 granos.

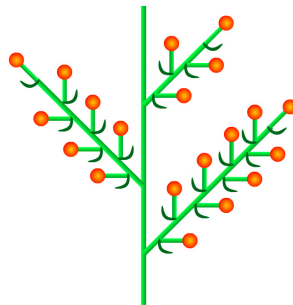


Figura 1: Ejemplo de una panícula

P-TRAP (Panicle Traits Phenotyping) es una herramienta de código abierto, escrita en Java, que dada una imagen, detecta y analiza automáticamente la estructura de



una panícula. Además cuenta y recolecta una serie de rasgos o parámetros relacionados con la forma de los granos [13]. La herramienta ofrece tres funciones principales: análisis de la panícula, conteo de granos y análisis de la forma del grano. Las tres funcionalidades pueden ser utilizadas de manera independiente o bien en simultáneo. Este trabajo se enfocará de ahora en adelante en la función que realiza la detección de granos. La figura 2 es un ejemplo de la herramienta P-TRAP detectando objetos circulares separados.

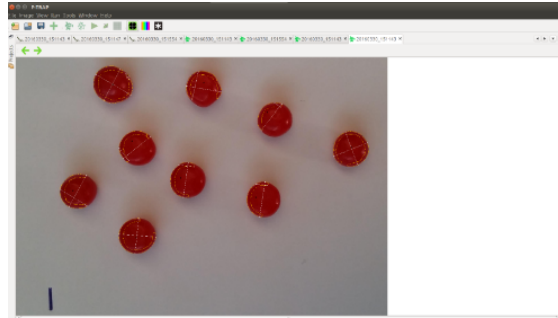


Figura 2: Ejemplo de P-TRAP en funcionamiento

Para realizar la detección de los granos se realizan 3 pasos principales. En el primer paso se convierte la imagen de RGB a binario utilizando un método de segmentación para obtener una imagen con granos blancos y fondo negro. En la figura 3 se presenta un ejemplo de dicha conversión.



(a) Grano en formato RGB (b) Grano en formato binario

Figura 3: Ejemplo de conversión

El segundo paso elimina cualquier arista o ruido que pueda tener el grano. En la figura 3a podemos observar una pequeña arista en el grano (apuntada con una flecha

roja), esto es lo que queremos eliminar en el segundo paso. Para esto se aplican los algoritmos "cvErode" y "cvDilate" de la biblioteca OpenCV [17]. En la figura 4 se muestra el resultado obtenido al finalizar el segundo paso. Se puede apreciar como la arista que habíamos visto en la figura 3a fue eliminada.



Figura 4: Grano sin ruido

El tercer paso utiliza la función "cvFindContour", basada en el algoritmo de Suzuki y Abe [18], la cual automáticamente detecta el perímetro de cada grano en la imagen binaria. En la figura 5 podemos observar como se delimita el grano con una línea roja.



Figura 5: Grano delimitado

Seguidamente, de acuerdo a los lineamientos de P-TRAP, se definen los siguientes rasgos de los granos para ser utilizados en el análisis que se desea realizar [13]:

- *Longitud*: La longitud de la línea más larga entre dos puntos en el contorno. Se utiliza la distancia euclidiana.

- *Anchura*: Para dos puntos en el contorno, la anchura es la longitud de la línea más larga perpendicular a la línea de longitud.
- *Área*: El número de píxeles del grano en la imagen binaria.
- *Compacidad*: La relación entre el area del grano y el contorno.
- *Elepticidad*: Mide la compresión de la figura circular a lo largo de su diámetro.
- *AR (Aspect Ratio)*: Es la relación entre la longitud y la anchura.

Los rasgos definidos anteriormente parten del hecho que cada grano ya fue detectado mediante el proceso de 3 pasos descrito previamente. Sin embargo, en algunas ocasiones, la conversión a binario de una imagen puede degradar el contorno de los granos complicando posteriormente la detección de los mismos.

## 2.5 Algoritmo Fast Radial Symmetry Transform

Existen muchas técnicas para encontrar puntos de interés en una imagen. Una de estas técnicas es la detección de puntos de interés a partir de la simetría radial local. En [8] se presenta el algoritmo *Fast Radial Symmetry Transform*, desarrollado en MATLAB[11], el cual realiza esta función.

En dicho trabajo se presenta un nuevo punto de interés basado en gradientes que detecta puntos de alta simetría radial. Este nuevo enfoque determina la contribución que cada pixel realiza a la simetría de píxeles a su alrededor, en lugar de considerar la contribución de un vecindario local a un pixel central.

La transformada presentada en [8] es calculada sobre un conjunto de uno o más rangos  $N$ , dependiendo de la escala de las características que se quieran detectar. El valor de la transformada en el rango  $n \in N$  indica la contribución a la simetría radial de los gradientes a una distancia  $n$  de cada punto. En cada rango  $n$  una *imagen de proyección orientación*  $O_n$  y una *imagen de proyección magnitud*  $M_n$  son formadas. Estas imágenes son generadas al examinar el gradiente  $\mathbf{g}$  en cada punto  $\mathbf{p}$

del cual se determina un *pixel afectado-positivamente*  $\mathbf{p}_{+ve}(\mathbf{p})$  y un *pixel afectado-negativamente*  $\mathbf{p}_{-ve}(\mathbf{p})$ . El *pixel afectado-positivamente* es definido como el pixel al que esta apuntando el vector de gradientes  $\mathbf{g}(\mathbf{p})$ , una distancia  $n$  de  $\mathbf{p}$ . El *pixel afectado-negativamente* es el pixel que está a una distancia  $n$  del gradiente apuntando en dirección opuesta.

La figura 6 muestra la ubicación de los pixeles  $\mathbf{p}_{+ve}(\mathbf{p})$  y  $\mathbf{p}_{-ve}(\mathbf{p})$  afectados por el elemento gradiente  $\mathbf{g}(\mathbf{p})$  para un rango de  $n = 2$ . El círculo punteado muestra todos los pixeles que podrían ser afectados por el gradiente en  $\mathbf{p}$  para un rango  $n$ .

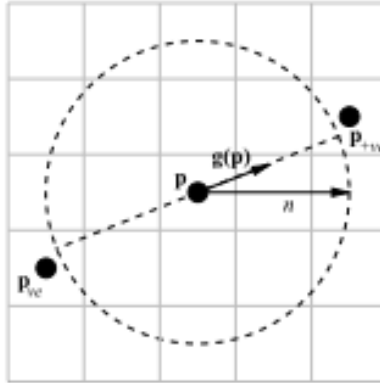
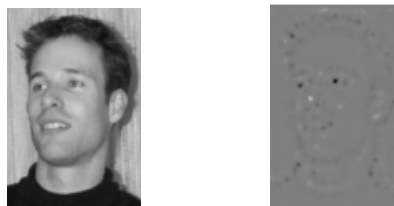


Figura 6: Ejemplo de *pixel afectado-negativamente* y *pixel afectado-positivamente* [8]

En la figura 7 se presenta un ejemplo para un conjunto de rangos  $N$  en el cual se desea resaltar los ojos de la cara. Se puede observar en la figura 7b como los ojos se resaltan del resto de la cara con 2 puntos negros.



(a) Foto original (b) Ojos detectados

Figura 7: Ejemplo de caras [8]

## 2.6 Espacio de Hough

La transformada de Hough [19] fue propuesta por Paul Hough en 1962 con el propósito de detectar rectas en una imagen. Más adelante, dicha transformada se extendió para identificar contornos cuya forma básica es conocida y puede ser representada por una curva paramétrica. Esto permitió que la transformada de Hough pudiera ser utilizada para la detección de círculos, elipses, parábolas, entre otras.

### 2.6.1 Representación de líneas en el espacio de Hough

Las líneas pueden ser representadas de manera única en un espacio bidimensional a partir de la ecuación 1.

$$y = ax + b \quad (1)$$

En la figura 8 se presenta un punto en el espacio de la imagen, en la posición  $(x_0, y_0)$  y en rojo se marcan algunas líneas que pasan por dicho punto. Todas aquellas líneas (a y b) que satisfacen  $y_0 = ax_0 + b$  pasan por el punto  $(x_0, y_0)$ .

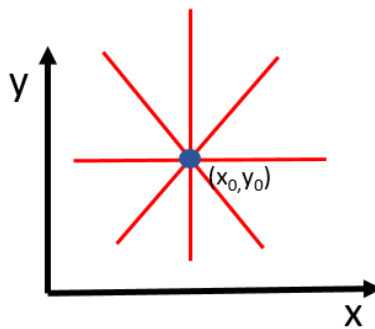
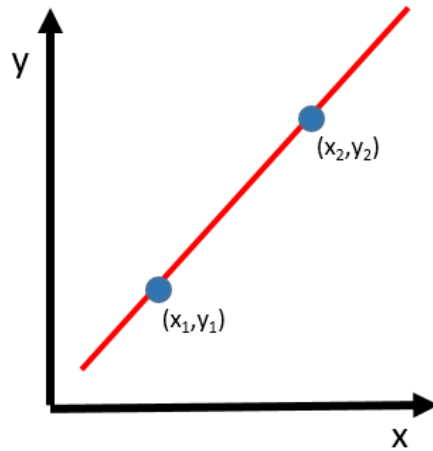


Figura 8: Ejemplo de líneas que pasan por el punto  $(x_0, y_0)$ .

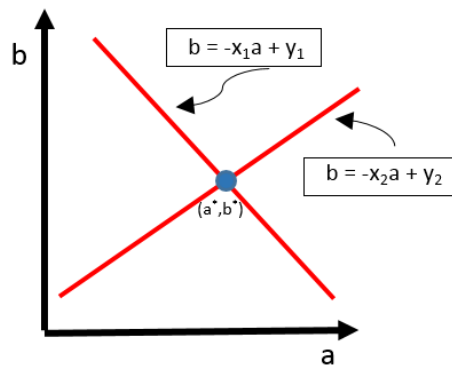
También, es posible definir la ecuación en términos de b, como se muestra en la ecuación 2. La ecuación 2 representa una línea en el espacio paramétrico de a y b. Es decir, un sólo punto en el espacio de la imagen corresponde a un número infinito de puntos en el espacio paramétrico (una línea), como se muestra en la figura 9. El

punto donde se intersectan las líneas (en el espacio paramétrico), son los parámetros de la recta que los une en el espacio de la imagen.

$$b = -x_0a + y_0 \tag{2}$$



(a) Dos puntos en el espacio de la imagen.



(b) Las dos líneas y su intersección en el espacio paramétrico que representan los dos puntos de la figura 9a, en el espacio de la imagen.

Figura 9: Ilustración de líneas en el espacio paramétrico a partir de dos puntos en el espacio de la imagen.

No obstante, con la ecuación 1 no es posible la representación de líneas verticales. Por esta razón la transformada de Hough utiliza una representación de dicha ecuación en términos de coordenadas polares, como se muestra en la ecuación 3.

$$\rho = x \cos \theta + y \sin \theta \quad (3)$$

El uso de coordenadas polares implica que los puntos en el espacio de la imagen ya no representarán una línea en el espacio paramétrico si no una curva sinusoidal como se muestra en la figura 10.

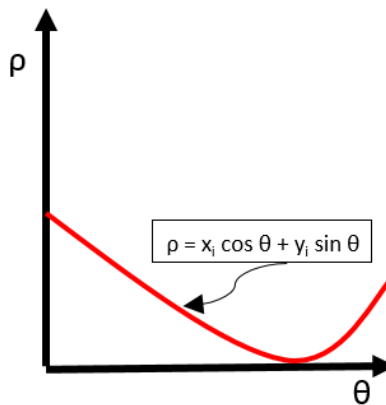


Figura 10: Ejemplo de curva sinusoidal en el espacio paramétrico.

### 2.6.2 Representación de círculos en el espacio de Hough

Como fue mencionado anteriormente, los círculos también pueden ser representados de manera única en un espacio bidimensional a partir de la ecuación 4 y aplicar Hough Transform para la detección de los mismos. En la ecuación 4, el punto  $(a,b)$  representa el centro del círculo y  $r$  su respectivo radio.

$$r^2 = (x - a)^2 + (y - b)^2 \quad (4)$$

En la figura 11 se observa un círculo en el espacio de la imagen. Para cada punto  $(x,y)$  del círculo original en el espacio de la imagen se puede definir un círculo con centro  $(x,y)$  y radio  $r$  (de acuerdo a 4) en el espacio paramétrico. El punto de intersección

de todos los círculos definidos en el espacio paramétrico corresponde al punto central del círculo original.

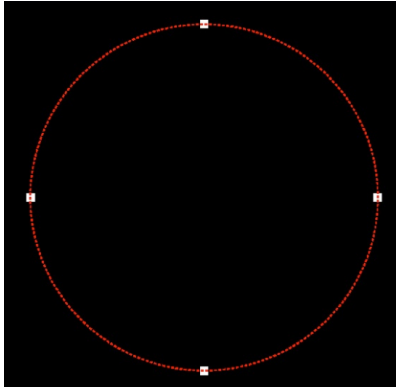


Figura 11: Ejemplo de imagen con un círculo.

En la figura 11 se marcan 4 puntos en el círculo original. Para cada uno de estos puntos se define un círculo en el espacio paramétrico. En la figura 12 se muestra el espacio paramétrico con los 4 círculos que representan los 4 puntos en el círculo original. Cada intersección entre los círculos fue marcada con un punto. Mediante el uso de una matriz acumulador se mantiene el registro de cada intersección. Luego, se procede a analizar dicha matriz para encontrar el punto en el cual se intersecan todos los círculos, este punto representa el centro del círculo original en el espacio de la imagen.

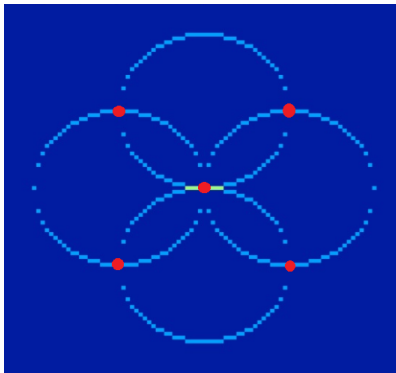


Figura 12: Ejemplo de imagen con un círculo en el espacio paramétrico.



### 2.6.3 Algoritmo

El algoritmo para detectar curvas paramétricas puede ser dividido en los siguientes pasos:

1. **Detección de orillas:** se aplica el algoritmo Canny [21] para detectar las orillas de la imagen. En la figura 13 se muestra el resultado que se obtiene después de aplicar el algoritmo Canny en una imagen, el resultado es una imagen binaria.

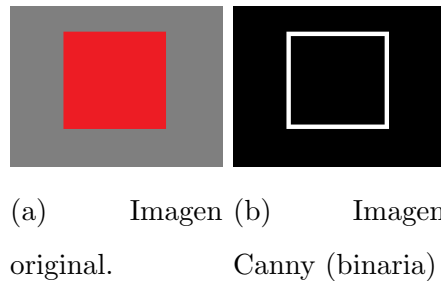


Figura 13: Aplicación del algoritmo Canny para detección de bordes.

2. **Mapeo en espacio Hough:** como fue detallado en la sección 2.6.1, cada punto orilla de la imagen binaria es mapeado al espacio de Hough como curvas sinusoidales. Los valores  $a$  y  $b$  son discretizados en un número limitado de valores, formando una matriz bidimensional en el espacio paramétrico llamado *acumulador*. Dicho acumulador se inicializa en cero. Cada orilla (mayor a un límite o como el máximo en la dirección del gradiente) contribuye a una serie de valores en el acumulador, sumándole una unidad a las combinaciones posibles de  $(a, b)$ . El elemento del acumulador que tenga un número mayor (más votos) corresponde a la ecuación de la línea deseada  $(a, b)$ .
3. **Interpretación de resultados:** se analiza el acumulador de resultados y se extraen los puntos con mayor cantidad de votos.
4. **Conversión:** las líneas infinitas se convierten en líneas finitas y se mapean en la imagen original.

## 3 Descripción de la investigación

### 3.1 Planteamiento del problema

Durante la última década se ha observado el impresionante avance tecnológico de las cámaras fotográficas. Actualmente, vivimos en una era donde contamos con dispositivos móviles capaces de tomar fotografías de alta calidad en cualquier lugar y momento.

Este avance ha sido el detonante para que en años recientes la visión por computadora se haya convertido en una tecnología clave en diferentes campos. En la industria automotriz son cada vez más populares los carros asistidos por computadoras. También, en la industria de videojuegos la visión por computadora ha sido utilizada para mejorar la interacción entre el usuario y el juego, creando una experiencia cada vez más real. En el área de la producción agrícola, la visión por computadora puede ser utilizada para mejorar la eficiencia y eficacia en los procesos de detección y conteo de granos. Ambos procesos, de ser realizados de forma manual, consumen mucho tiempo y existe cierto grado de error humano que afecta el resultado final [2].

En el Instituto del Café de Costa Rica (ICAFE) cuenta con un proceso para estimar la producción de una planta en particular a partir del número de granos de café en la planta en cierto punto de su desarrollo. Actualmente, el proceso de conteo de granos se realiza de forma manual, consumiendo una gran cantidad de horas hombre. ICAFE cuenta con un conjunto de 50 imágenes de granos de café y se desea encontrar una manera automática de realizar el conteo de granos con el fin de incrementar la eficiencia en el proceso.

En la literatura, se han propuesto varias soluciones para resolver este problema [3],[4],[5],[6],[7]. Sin embargo, la gran mayoría utiliza como entrada imágenes de granos en un ambiente controlado. Este es el caso de P-TRAP [13], una herramienta

de código abierto que implementa una serie de algoritmos para la detección de granos en un ambiente controlado. P-TRAP fue diseñado con un propósito en particular: analizar la estructura de panículas de arroz en imágenes para entender la diversidad y mejorar la producción de arroz. Para realizar esta labor, la plataforma cuenta con tres componentes diferentes: análisis de la estructura de la panícula, conteo de granos de arroz y análisis de la forma [13]. Este estudio se enfoca en el componente para el conteo de granos.

El componente de conteo de granos de P-TRAP utiliza un enfoque granulométrico el cual consiste en obtener la distribución por tamaño de los granos presentes. El primer paso del proceso de detección de granos de P-TRAP es convertir la imagen de entrada de RGB a binario, para luego seguir con el enfoque granulométrico.

En la figura 14 se presentan dos ejemplos de imágenes de granos. La figura 14a muestra una de las imágenes proporcionadas por ICAFE para las pruebas de detección de granos y la figura 14b es el tipo de imagen para el cual P-TRAP fue creado originalmente y sobre el que hace una detección de granos precisa.



(a) Granos de ICAFE



(b) Granos de P-TRAP

Figura 14: Imágenes de granos en formato RGB

Como se mencionó previamente, el primer paso en el proceso de detección de granos en P-TRAP es la conversión de RGB a binario. En la figura 15 se presentan las imágenes después de realizar la conversión respectiva.



(a) Granos de ICAFE



(b) Granos de P-TRAP

Figura 15: Imágenes de granos en formato binario

Como se puede observar en la figura 15b, los granos quedan bien delimitados y en un formato adecuado para continuar con el proceso de detección y aplicar el enfoque granulométrico. Sin embargo, en el caso de la figura 15a se perdió por completo el contorno de los granos, en este caso cualquier estrategia de detección que se aplique provocará resultados imprecisos.

Realizar la detección de granos de café a partir de imágenes tomadas fuera de un ambiente controlado agrega una serie de dificultades al proceso. Algunas de estas dificultades son:

1. Los granos traslapados imponen una dificultad ya que el grano pierde su forma y en ocasiones solo una mínima porción del grano queda a la vista.
2. Los granos que están detrás del racimo, que no se encuentran a la vista. Detras del racimo visible en la imagen existen mas granos de cafe pero se desconoce la cantidad.

3. Distancia de los granos con respecto a la cámara. En un ambiente controlado es posible tratar esto como una constante y definir un rango de tamaños. Sin embargo, en el caso de las imágenes tomadas en su ambiente natural no se tiene control sobre esto. El tamaño de los granos de café en la imagen va estar determinada por la distancia entre el racimo y el lente de la camara.
4. Ruido en el fondo de la imagen. En un ambiente controlado el fondo de la imagen tiene un mismo color y textura. En el caso de las proporcionadas por ICAFE, en el fondo se observan hojas, ramas y otro tipo de objetos que le agregan ruido a la imagen.

En el caso especfico de P-TRAP y como se mostr anteriormente, el proceso que permite realizar la deteccin de granos no es capaz de resolver dichas dificultades por lo que la precisin de la deteccin de granos de caf es muy baja. En la figura 16 se presenta un ejemplo en P-TRAP donde se ejecut el proceso de deteccin y conteo de granos sobre una de las imgenes de ICAFE. Como se puede observar, la deteccin realizada no fue exacta e incluso se marcan reas en las hojas que no pertenecen a los granos de caf. Algunas de las reas ms relevantes marcadas son apuntadas por flechas rojas.

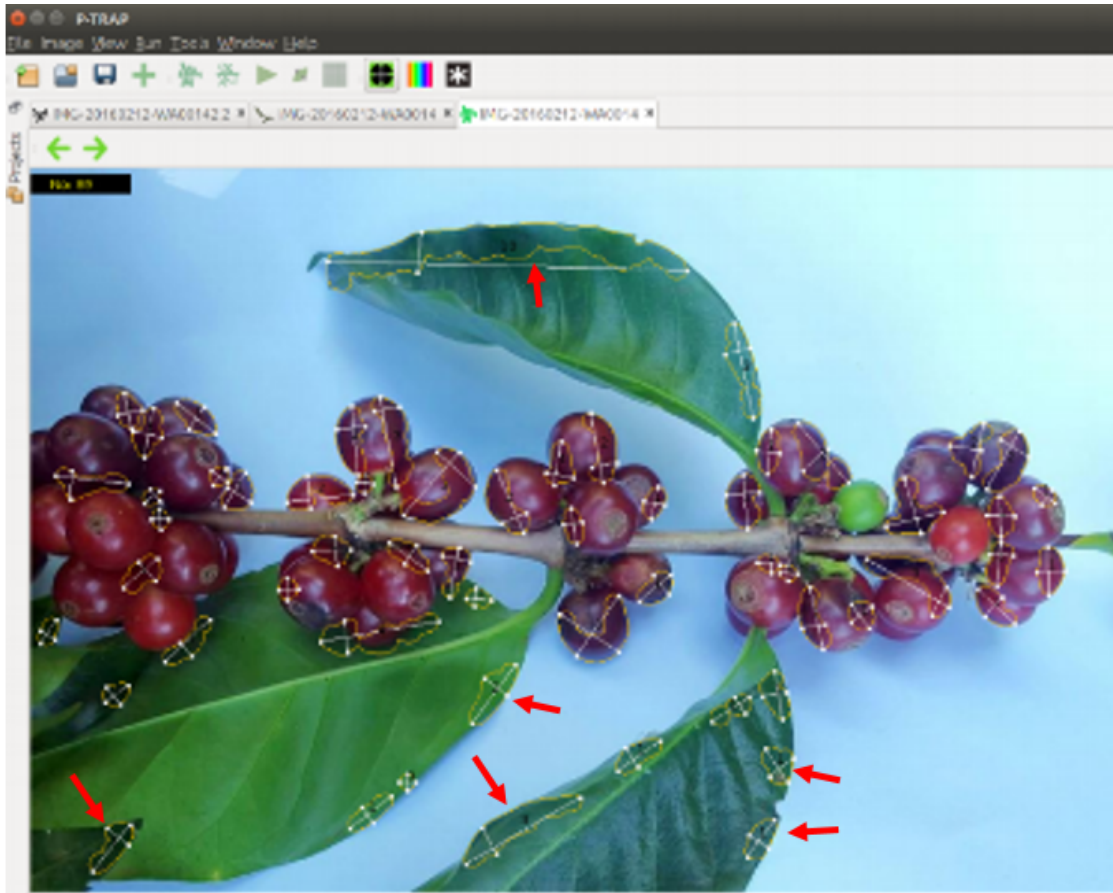


Figura 16: Ejemplo de detección de granos en P-TRAP

### 3.2 Descripción general de la investigación

El algoritmo *Hough Transform* es capaz de detectar curvas paramétricas, como fue explicado en 2.6. Asimismo, el algoritmo *Fast Radial Symmetry Transform* permite identificar regiones de interés en una imagen mediante el análisis de la simetría radial local, como se detalló en 2.5.

Dada la naturaleza simétrica de los granos de café y su figura paramétrica (círculo) es posible que se pueda aplicar *Hough Transform* o *Fast Radial Symmetry Transform* al proceso de detección de granos de P-TRAP. Específicamente en el inicio, antes del primer paso (conversión de RGB a binario). Esto permitiría que los granos existentes en la imagen original no se pierdan cuando se realice la conversión

de RGB a binario puesto que habrían quedado previamente demarcados por el paso anterior donde se aplicó alguno de los dos algoritmos. Sin embargo, aún existen factores en la imagen que podrían afectar negativamente las detecciones realizadas por ambos algoritmos. Por ejemplo, el ruido en la imagen incrementa los falsos positivos en las detecciones realizadas por el algoritmo *Hough Transform* y las diferentes texturas en la imagen pueden ocasionar falsos negativos en las detecciones del algoritmo *Fast Radial Symmetry Transform*. Por estas razones, la implementación de ambos algoritmos en combinación podría potencialmente reducir los falsos positivos y falsos negativos en las detecciones realizadas. De manera que, solamente se tomen en cuenta las detecciones realizadas por ambos y no aquellas detecciones realizadas solo por uno u otro.

A partir de lo descrito anteriormente, este proyecto pretende estudiar la exactitud, precisión, exhaustividad y f1-score en la detección de granos a partir de la incorporación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform*, individualmente y en combinación, al proceso de detección y conteo de granos ya existente en la herramienta P-TRAP[13].

Como se ha explicado anteriormente, la herramienta de código abierto P-TRAP[13] ofrece más funciones a parte de la función de detección y conteo de granos. Dichas funciones hacen que la herramienta sea más compleja desde un punto de vista técnico. Por esta razón, la solución cuenta con una gran cantidad de módulos que permiten la reutilización de código entre las diferentes funciones. Si bien es una buena práctica de programación, antes de realizar cualquier cambio en el código se requiere de un análisis y estudio a fondo de la solución con el fin de evitar cualquier tipo de daño colateral. En el caso particular de este proyecto, se debe realizar este estudio a fondo de la solución actual para poder realizar la incorporación de los algoritmos *Hough Transform* y *Fast Radial Symmetry Transform* de manera exitosa. En la figura 17 se muestra la compleja interacción entre los diferentes módulos de la herramienta P-TRAP a la hora de analizar una imagen.

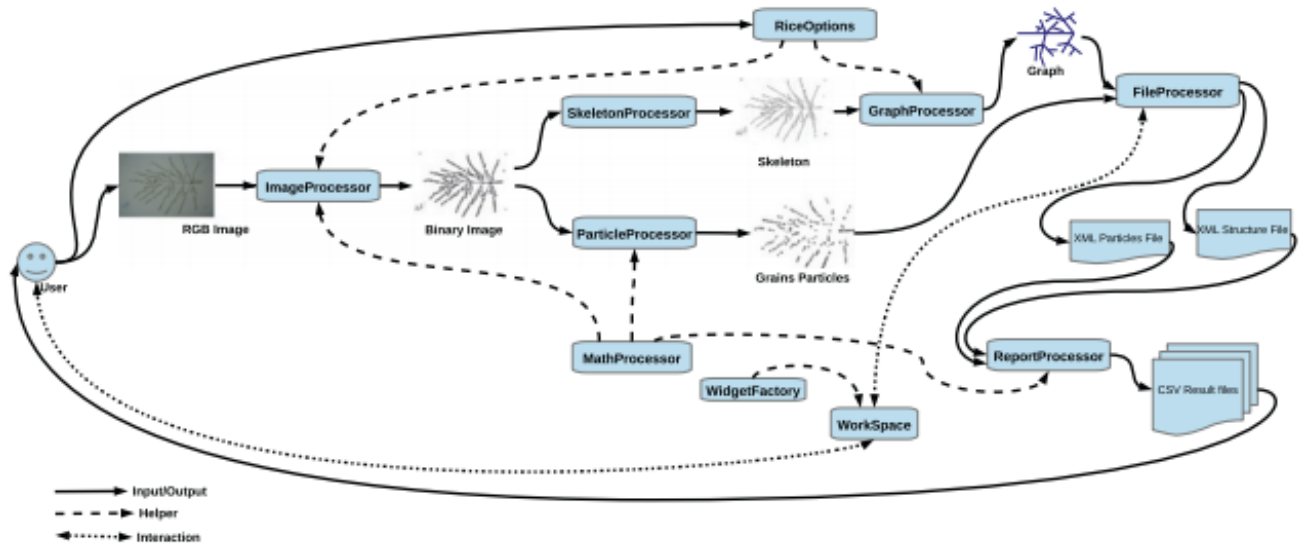


Figura 17: Ejemplo de interacción de módulos de P-TRAP

Asimismo, a parte del complejo análisis de la solución que se debe hacer, se debe realizar el desarrollo en Java del algoritmo *Fast Radial Symmetry Transform*. Como fue explicado en la sección 2.5, dicho algoritmo cuenta con un proceso complejo y detallado, lo cual incrementa la dificultad en la programación del mismo. Tanto el análisis detallado de la herramienta y la interacción entre todos los módulos, así como la compleja implementación en Java del algoritmo *Fast Radial Symmetry Transform* y *Hough Transform* hacen que dicho proyecto requiera de un trabajo de programación delicado y complejo.

Este estudio omite y deja como parte del posible trabajo futuro, la detección de granos de café traslapados y aquellos no visibles en la imagen. Asimismo, queda para trabajo futuro la estimación de la producción de la planta de café a partir de los datos recolectados.

### 3.3 Trabajos relacionados

A continuación se presentan los trabajos más relevantes relacionados con este proyecto de investigación.



En [5] se propone un marco de trabajo para el conteo automático de granos. Este framework esta compuesto de hardware y software. En cuanto a hardware, se propone un mecanismo de vibración para que los granos sean esparcidos de forma uniforme y no queden traslapados. Luego, se toma una foto que será convertida a escala de grises y pasará por un proceso de eliminacion de ruido, conversión binaria y morfología matemática. Ese sistema de vibración parte del supuesto que los granos están separados pero en el caso de las imágenes de ICAFE, los granos aún están adheridos a la planta.

En [6] se presenta un método automático que utiliza visión por computadora para identificar y contar uvas. En este estudio se crea un ambiente semi-controlado para la captura de imágenes donde un carro recorre lateralmente las plantaciones con una camara de alta tecnología y un sistema de luces para asegurar la calidad de las imágenes. Seguidamente, se detectan las uvas con base en la forma y textura. En el caso particular de ICAFE, las imágenes son tomadas en un ambiente no controlado y agregar cualquier tipo de hardware incrementa el costo de la solución.

En [7] desarrollan una aplicación para dispositivos móviles capaz de identificar especies de árboles a partir de fotos de sus hojas. Entre los aportes principales desarrollados en este trabajo destaca el componente de visión por computadora que descarta imágenes que no son de hojas. Para procesar las imágenes se procede a segmentar la hoja del fondo de la imagen, se extrae las características de la curvatura de la hoja y se realiza una comparación contra un conjunto de entrenamiento para determinar la especie de árbol. Las características analizadas (curvatura) en esta investigación son diferentes a las que se requieren analizar en el caso de las imágenes de granos de café.

En [4] se analizan las manchas en las hojas a partir de imágenes con el propósito de diagnosticar posibles enfermedades en la planta. En este proyecto se desarrolla un sistema de procesamiento de imágenes integrado. Este sistema cuenta con cuatro

etapas: preprocesamiento, segmentación, extracción de características y clasificación. En esta investigación se conoce de antemano las manchas que se están buscando por lo que se cuenta con un conjunto de entrenamiento, en esta propuesta no requeriríamos conjunto de entrenamiento para realizar la detección.

En [3] se presenta una propuesta para investigar diferentes tipo de características en imágenes de granos de arroz para determinar la variedad.

En [27] se presenta una propuesta para el conteo y detección de granos de café. En este proyecto se propone una estrategia que cuenta con cinco fases. En la primera fase se recolectan y se ajustan las imágenes. En la segunda fase se segmentan regiones homogéneas con el fin de detectar zonas que puedan brindar información sobre la textura. En la tercer fase se detectan arcos que podrían pertenecer o no a granos de café. La cuarta fase consiste en ajustar los puntos encontrados en la fase anterior a una elipse para evaluar si existe un grano o no. Finalmente, en la quinta fase se detectan, clasifican y cuentan los granos. Este trabajo brinda una solución al mismo problema planteado en este trabajo de investigación. Sin embargo, no se obtienen buenos resultados cuando hay una distancia considerable entre los granos, cuando el racimo no esta en el centro de la imagen y cuando hay un exceso de luz o oscuridad en la imagen.

Se analizaron multiples plataformas que implementan funciones de vision por computador. Sin embargo, se decidio hacer uso de P-TRAP porque ya cuenta con una interfaz grafica de usuario amigable, esta desarrollada en un lenguaje de programacion multiplataforma, ofrece componentes de reporte de resultados que pueden ser reutilizados y fue desarrollada exclusivamente para el analisis de granos por lo que los usuarios actuales podran tambien hacer uso de la version desarrollada en este trabajo sin tener que instalar otra herramienta.

## 3.4 Hipótesis

Con base en la definición del problema planteado y el marco teórico descrito anteriormente, se define la siguiente hipótesis:

*La incorporación de los algoritmos Fast Radial Symmetry Transform y Hough Transform al proceso propuesto por P-TRAP, mejora el desempeño, medido por el promedio armónica de precisión y de exhaustividad, en la detección de granos de café existentes en una imagen.*

## 3.5 Objetivos

### 3.5.1 Objetivo general

El objetivo general de la presente investigación es mejorar el desempeño, obtenido por el *f1-score*, en la detección de granos de café de la herramienta P-TRAP[13] al incorporar los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform*.

### 3.5.2 Objetivos específicos

Los objetivos específicos de este proyecto son los siguientes:

1. Implementar los algoritmos *Fast Radial Symmetry Transform*[8] y *Hough Transform* en el proceso de análisis de imágenes de la herramienta de código abierto P-TRAP[13].
2. Evaluar el desempeño, obtenido por el *f1-score*, en la detección de granos de café existentes en una imagen utilizando P-TRAP[13] (en su versión original) versus la versión desarrollada incorporando los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform*.
3. Brindar una explicación sobre los resultados obtenidos.

## 3.6 Resumen de experimentos

Para el diseño e implementación de la parte experimental de este trabajo se usan los *lineamientos* desarrollados en el libro de Douglas C. Montgomery [1]. El esquema del procedimiento recomendado para el desarrollo de esta fase incluyó lo siguiente:

1. **Reconocimiento y definición del problema:** desarrollar una declaración clara y sencilla del problema.
2. **Selección de factores, niveles y rangos:** enumerar todos los posibles factores que pueden influenciar el experimento. Incluye los factores de diseño potencial y los factores perturbadores. Asimismo, se deben seleccionar los rangos sobre los que varían los distintos factores y los niveles específicos sobre los que se aplicarán las iteraciones del experimento.
3. **Selección de la variable de respuesta:** proveer información útil sobre el fenómeno que está siendo estudiado.
4. **Selección del diseño de experimental:** hace referencia a aspectos claves del experimento tales como el tamaño de la muestra, selección del orden adecuado para la ejecución de los intentos experimentales y la decisión de bloquear o no algunas de las restricciones de aleatoriedad en la pruebas.
5. **Llevar a cabo el experimento:** monitorear el proceso para asegurar la correcta ejecución del experimento con respecto a lo planeado.

### 3.6.1 Declaración del problema

Estudiar el *f1-score* en la detección de granos de café existentes por parte de la versión por defecto de P-TRAP[13] y la versión desarrollada en este estudio incorporando los algoritmos *Fast Radial Symmetry Transform*[8] y *Hough Transform*.

### 3.6.2 Factores

En el diseño de experimentos, un factor es aquel componente que tiene cierta influencia en las variables de respuesta [1]. El objetivo de un experimento es determinar esta influencia.

En el desarrollo de este experimento se seleccionaron los siguientes factores para su estudio:

1. Proceso de detección
  - Proceso por defecto de P-TRAP
  - Proceso por defecto de P-TRAP más la incorporación del algoritmo *Fast Radial Symmetry Transform*
  - Proceso por defecto de P-TRAP más la incorporación del algoritmo *Hough Transform*
  - Proceso por defecto de P-TRAP más la incorporación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* combinados
2. Imágenes de granos de café

### 3.6.3 Variables de respuesta

A continuación se detallan las variables a considerar para evaluar los experimentos de este proyecto:

- Exactitud (accuracy): Corresponde al porcentaje de detecciones correctas, tomando en cuenta el total de la población. Se calcula de la siguiente forma:

$$\frac{total\_aciertos}{total\_predicciones} \quad (5)$$

- Precisión: Corresponde al porcentaje de granos detectados que sí existen en la imagen. Se calcula de la siguiente forma:

$$\frac{verdaderos\_positivos}{verdaderos\_positivos + falsos\_positivos} \quad (6)$$

- Exhaustividad (recall): Corresponde al porcentaje de granos existentes que sí fueron detectados. Se calcula de la siguiente forma:

$$\frac{\textit{verdaderos\_positivos}}{\textit{verdaderos\_positivos} + \textit{falsos\_negativos}} \quad (7)$$

- f1-score: Promedio ponderado de la precisión y la exhaustividad. Se calcula de la siguiente forma:

$$2 * \left( \frac{\textit{precision} * \textit{exhaustividad}}{\textit{precision} + \textit{exhaustividad}} \right) \quad (8)$$

La variable de respuesta utilizada para seleccionar y comparar los distintos algoritmos es la métrica *f1-score*, sin embargo se recolectaron las demás variables de respuesta para su respectivo análisis.

### 3.6.4 Recolección de datos

La obtención de las distintas variables de respuesta fue realizada de forma automática por el programa durante la realización de cada experimento.

### 3.6.5 Análisis estadístico

Se utilizaron bibliotecas en R [28] para el análisis estadístico con el propósito de validar si las distribuciones obtenidas como resultado de los experimentos mostraron o no diferencias significativas de acuerdo a la estrategia utilizada.

En este análisis se utilizó la prueba de normalidad Lilliefors [22] y se determinó que, en la mayoría de los casos a evaluar, los valores de *f1-score* no estaban distribuidos normalmente.

Para la comparación estadística de los resultados, no se puede utilizar el procedimiento paramétrico de ANOVA [30] porque dicho método asume la distribución normal de los valores. De esta forma, se procedió a utilizar el método de varianza unifactorial por rangos de Kruskal-Wallis[9], que no asume una distribución normal de los valores [29].

La técnica de Kruskal-Wallis[9] prueba la hipótesis nula de que las  $k$  muestras provienen de la misma población o de poblaciones idénticas con la misma mediana. Para especificar explícitamente la hipótesis nula y alterna,  $\theta_j$  debe ser la mediana de la población para el  $j$ -ésimo grupo o muestra [10]. La comparación de las diferentes versiones del algoritmo estudiadas en el experimento, surgieron a partir de las siguientes hipótesis:

$\mathbf{H}_0$  : La distribución de la métrica f1-score para el par de algoritmos es la misma.

$\mathbf{H}_1$  : La distribución de la métrica f1-score para el par de algoritmos no es la misma.

Al encontrar que al menos una de las distribuciones de f1-score fue diferente, se usó la prueba post-hoc de Nemenyi [24] en pares con la distribución de Tukey, para comparar los pares de algoritmos.

### 3.7 Detalles de experimentos

Para llevar a cabo los experimentos descritos en la anteriormente, se requirieron una serie de pasos de preparativos que incluyen:

1. Instalación y preparación del ambiente de desarrollo.
2. Preparación del conjuntos de datos de imágenes.
3. Implementación de los algoritmos.
4. Selección de parámetros para los algoritmos.
5. Comparación del desempeño (f1-score) de los distintos algoritmos.

### 3.8 Ambiente de desarrollo

El desarrollo de este proyecto se realizó en una máquina de 64 bits con Windows 8.1 Enterprise. La herramienta P-TRAP[13] fue desarrollada en el lenguaje de programación Java por lo que se utilizó NetBeans [12] en su versión 8.0.1 como ambiente integrado de desarrollo. Asimismo, se utilizó la versión 9.0 de MATLAB [11] como referencia a la implementación del algoritmo *Fast Radial Symmetry Transform*[8].

Durante la etapa de desarrollo y posteriormente durante la ejecución de las pruebas se utilizaron bibliotecas de Java y R. A continuación se detalla el uso de cada una:

- **OpenCV 3.2 [20]**: Requerido para la implementación del algoritmo *Hough Transform* y la conversión de imágenes.
- **Nortest 1.0-4 [23]**: Requerido para realizar la prueba de normalidad Lilliefors.
- **PMCMR 4.1 [31]**: Requerido para la ejecución de pruebas Kruskal-Wallis y posteriormente la comparación de pares utilizando Nemenyi.

### 3.9 Conjunto de datos

Para la ejecución de la fase experimental de este trabajo de investigación se utilizó un conjunto de imágenes proporcionado por ICAFE. Este conjunto está conformado por 50 imágenes en formato PNG. A continuación se presentan las principales características que describen al conjunto de imágenes:

- Las imágenes fueron tomadas desde un dispositivo móvil en un ambiente no controlado.
- Las imágenes fueron tomadas en diferentes puntos en el tiempo por lo que hay imágenes de granos de café en diferentes etapas de maduración.
- Los granos no fueron separados de la planta.



- Las imágenes fueron tomadas durante el día por lo que no fue necesario la utilización de luces artificiales para mejorar la visibilidad de los granos.
- La distancia entre la cámara y los granos varía.

## 3.10 Implementación de los algoritmos

### 3.10.1 Implementación de Fast Radial Symmetry Transform

La programación del algoritmo *Fast Radial Symmetry Transform* se realizó en el lenguaje de programación Java con el propósito de ser integrado posteriormente a la herramienta P-TRAP. Existía previamente una versión de dicho algoritmo en la herramienta MATLAB. Esta versión de MATLAB fue utilizada como referencia durante el desarrollo de su versión en Java. Una vez finalizado el desarrollo del algoritmo se realizaron pruebas en paralelo a la versión de MATLAB para corroborar que los resultados obtenidos eran los mismos. Este algoritmo, recibe los siguientes datos de entrada:

- **Imagen:** Imagen en escala de grises que va ser utilizada para la detección de regiones.
- **Radio:** A partir del radio mínimo y máximo se define el conjunto de rangos que va ser utilizado para la detección de regiones.
- **Alfa:** Define el nivel de rigor utilizado durante la detección de regiones simétricas. Un valor muy alto podría ocasionar gran cantidad de regiones detectadas y el efecto contrario si el valor es bajo. Los autores recomiendan un valor fijo de 2.
- **Modo:** Existen 3 tipos de modo (claro, oscuro o ambos) con los cuales es posible ejecutar el algoritmo dependiendo de las regiones simétricas que se desean detectar.

A continuación se explica en más detalle los pasos de la implementación.

---

**Algoritmo 1** Fast Radial Symmetry Transform

---

```
1: procedure APPLYFASTRADIALSYMMETRYTRANSFORM
2:   Convertir la imagen de entrada a escala de grises
3:   Obtener la altura y el ancho de la imagen
4:   Determinar el rango de radios
5:   Definir el gradiente para X y Y
6:   while  $i < radius\_range$  do
7:     Ciclo por cada radio en el rango
8:     Definir las matrices de orientacion y magnitud en cero
9:     while  $x < image\_height$  do
10:      while  $y < image\_width$  do
11:        Ciclo por cada pixel en la imagen
12:        Obtener el gradiente normalizado (gnorm) para (x,y)
13:        if  $gnorm > 0$  then
14:          if  $mode = bright$  then
15:            Definicion de pixel afectado positivamente
16:            Sumar un punto al pixel afectado positivamente en la matriz
17:            orientacion
18:            Sumar el gradiente normalizado al pixel afectado positivamente
19:            en la matriz magnitud
20:          end if
21:          if  $mode = dark$  then
22:            Definir de pixel afectado negativamente
23:            Restar un punto al pixel afectado negativamente en la matriz
24:            orientacion
25:            Restar el gradiente normalizado al pixel afectado negativamente
26:            en la matriz magnitud
27:          end if
28:        end if
29:      end while
30:    end while
31:  end while
32:  Obtener el valor absoluto de la matriz orientacion y magnitud y cada valor
33:  se divide entre
34:  el maximo
35:  Multiplicar ambas matrices (O y M)
36:  Aplicar un filtro gaussiano
37:  Guardar los resultados en la dimension correspondiente al radio que se esta
38:  procesando
39:  Sumar todos los resultados de cada radio
40:  Aplicar escala al resultado
41: end procedure
```

---

### 3.10.2 Implementación de Hough Transform

Se utilizó la biblioteca OpenCV para la utilización del algoritmo *Hough Transform* mediante la función *HoughCircles*. Previo al uso esta función, las imágenes fueron convertidas a escala de grises y se aplicó un filtro gaussiano de 5x5 mediante el uso de la función *medianBlur*. Esto con el fin de disminuir el ruido general en la imagen y obtener mejores resultados. Este algoritmo, recibe los siguientes datos de entrada:

- **Imagen:** Imagen en escala de grises que va ser utilizada para la detección de círculos.
- **Método:** El método de detección utilizado. La versión de OpenCV utilizada solo es compatible con el método (*CV\_HOUGH\_GRADIENT*) descrito en [25].
- **dp:** Proporción inversa de la resolución del acumulador. Si *dp* tiene un valor de 1, el acumulador tiene la misma resolución que la imagen de entrada, si tiene un valor de 2 la altura y el ancho del acumulador son la mitad de grande de la imagen de entrada.
- **minDist:** Distancia mínima entre los centros de los círculos detectados. Si el valor es bajo se van a obtener muchos falsos positivos alrededor de un verdadero positivo.
- **Radio:** A partir del radio mínimo y máximo se define el conjunto de rangos que va ser utilizado para la detección de regiones.

A continuación se explica en más detalle los pasos de la implementación.

---

**Algoritmo 2** Hough Transform

---

```
1: procedure APPLYHOUGHTRANSFORM
2:   Convertir la imagen de entrada a escala de grises
3:   Aplicar Canny para detectar orillas
4:   Obtener la altura y el ancho de la imagen
5:   while  $x < image\_height$  do
6:     while  $y < image\_width$  do
7:       Ciclo por cada pixel en la imagen
8:       Obtener el gradiente normalizado (gnorm) para (x,y)
9:       if pixelesorilla then
10:        while Para todo theta do
11:          Obtener el angulo en radianes
12:          Calcular la distancia desde el origen
13:          Calcular el valor de rho
14:          Incrementar la celda respectiva en la matriz acumulador
15:        end while
16:      end if
17:    end while
18:  end while
19: end procedure
```

---

### 3.10.3 Implementación en P-TRAP

Como parte de esta investigación, la combinación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* fue implementada en el código de la herramienta *P-TRAP*. Esta implementación facilitó la ejecución de los experimentos y agregó una nueva funcionalidad para su potencial uso por parte de otros usuarios. Para realizar este trabajo se analizó la estructura y dependencias del programa y se procedió a realizar las siguientes modificaciones:

1. La herramienta hace uso del patrón de diseño *observador* para el manejo de las acciones que ocurren en la interfaz gráfica. En este caso, se procedió a crear una clase que representa la acción que se desea agregar y a registrar esa clase a un control específico de la interfaz gráfica. De esta manera, cuando el control cambie de estado, se procede a ejecutar el proceso de detección que implementa los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform*. En la figura 18 se muestra como se le presenta la opción al usuario. Se decidió colocar esta opción por debajo de la acción predeterminada de P-TRAP para la detección de granos para seguir el orden de funciones.

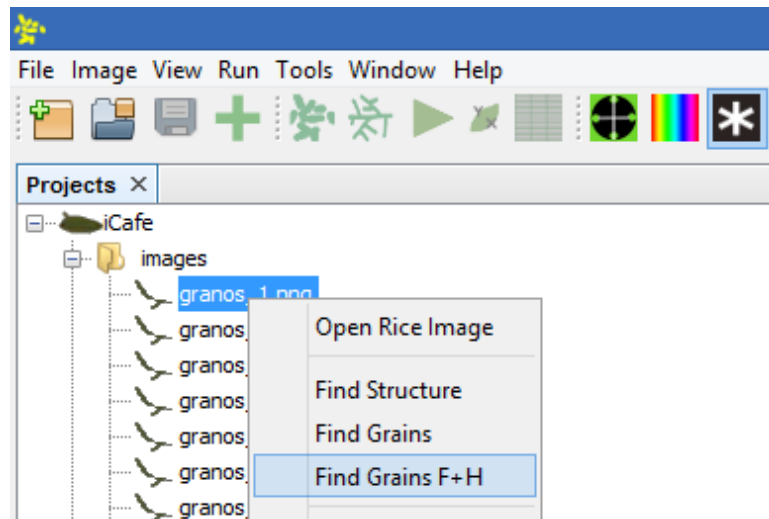


Figura 18: Opción agregada en P-TRAP

2. Se aplica el patrón de diseño *singleton* en la clase creada anteriormente para restringir la creación de objetos.

3. Mediante el uso de la biblioteca `javax.swing.JOptionPane` se agrega la funcionalidad que le permite al usuario definir los parámetros. En la figura 19 se muestra la pantalla que le permite al usuario definir el parámetro *modo*.

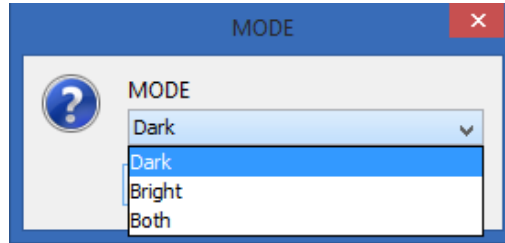


Figura 19: Pantalla para definir parámetro *modo*

4. Se agregan las clases que contienen el código de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* al módulo de procesamiento de imágenes de *P-TRAP*. Una vez que las clases son agregadas a este módulo, es posible la invocación de los métodos desde el módulo de trabajo (workspace).
5. Se agrega el llamado a los métodos agregados en el paso anterior en el módulo de trabajo (workspace), permitiendo la integración de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* con el resto.

En la figura 20 se observa el resultado, en *P-TRAP*, obtenido después de aplicar la detección de granos utilizando la combinación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform*. Se observa que se logró detectar los cuatro granos, sin detecciones negativas. En la figura 21 se observa el resultado de la misma imagen pero utilizando la función de detección de granos predefinida en *P-TRAP*, donde queda un grano sin detectar y se realiza una detección falsa.

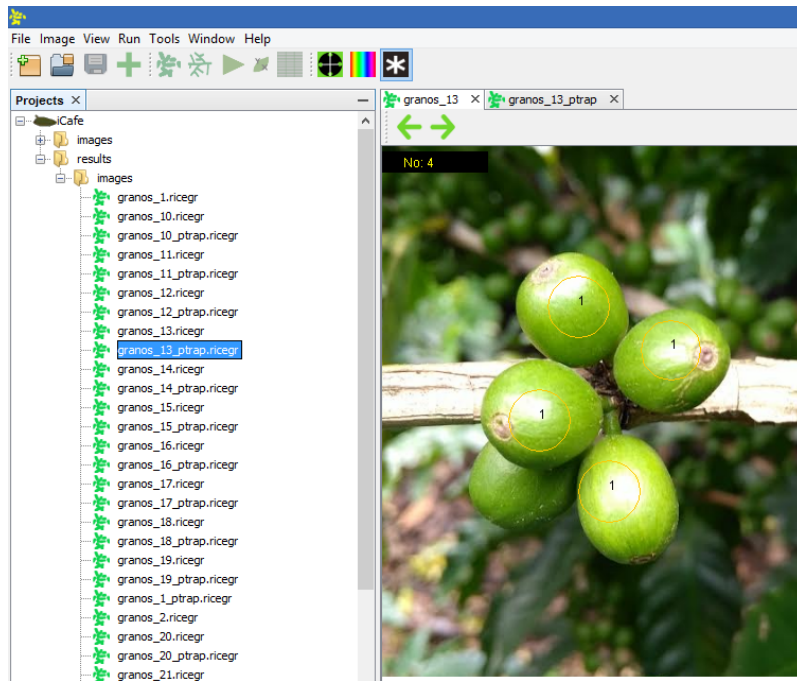


Figura 20: Resultado utilizando *Fast Radial Symmetry Transform* y *Hough Transform*

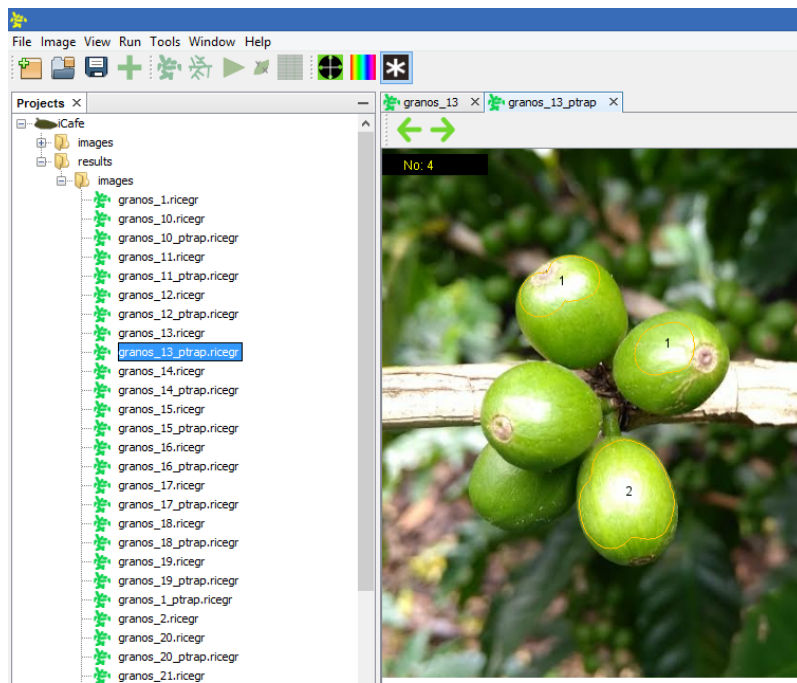


Figura 21: Resultado utilizando P-TRAP

## 3.11 Selección de parámetros

Como fue descrito en la sección 3.10, cada uno de los algoritmos utilizados en la fase experimental de esta investigación para la detección de granos de café requiere de la definición previa de una serie de parámetros. Con la selección precisa de los parámetros se busca obtener un alto desempeño en las detecciones. Los valores evaluados en cada parámetro se eligieron según los valores observados en la literatura y pruebas preliminares. Como se detalló en la sección 3.9, cada imagen fue tomada a distancias diferentes, por lo que el diámetro de los granos varía entre las imágenes y por ende su radio. Por esta razón, el radio de los granos en cada imagen fue determinado manualmente.

### 3.11.1 Parámetros de Fast Radial Symmetry Transform

- **Radio:** Las combinaciones de parámetros utilizadas se encuentran en el apéndice A.1.
- **Alfa:** Se determinó un valor fijo de 2 a partir de la recomendación de los autores en [8].
- **Modo:** Se utilizó el valor *ambos* para todas las imágenes. De utilizar el parámetro *claro* o *oscuro* se estaría realizando una detección parcial de los granos. En el conjunto de imágenes se pueden encontrar granos que poseen tonos claros y oscuros por lo que la opción *ambos* permite realizar una detección más precisa.

### 3.11.2 Parámetros de Hough Transform

- **Radio:** Las combinaciones de parámetros utilizadas se encuentran en el apéndice A.1.
- **Método:** Se utilizó el método *CV\_HOUGH\_GRADIENT* de acuerdo a lo descrito en [25].



- **dp**: Se determinó un valor de 1 para que el acumulador tuviera la misma resolución que la imagen de entrada.
- **minDist**: Se utiliza el mismo valor de radio mínimo con el fin de reducir los falsos positivos alrededor de una detección verdadero positivo.

### 3.12 Alcance del proyecto

Los siguientes aspectos resumen el alcance del presente proyecto de investigación a ser considerado para la evaluación del mismo:

- Se aporta una versión de P-TRAP[13] que incluye el algoritmo *Fast Radial Symmetry Transform* y *Hough Transform* en el proceso de detección de granos y el conjunto de imágenes utilizado para la realización de los experimentos
- Todas las funcionalidades de P-TRAP[13] están disponibles en la nueva versión.
- Análisis estadístico para contrastar los resultados de los experimentos.
- El análisis de resultados se basó únicamente en la métrica de puntaje f1-score.
- Explicación sobre los resultados obtenidos.
- Solo se cuentan los granos de café visibles en la foto.
- Solo se realizó el conteo de granos de café para imágenes RGB.
- Se debe proporcionar el radio mínimo y máximo para la detección de granos.
- Quedó por fuera de este trabajo el conteo de granos traslapados
- No se cuentan granos de café no visibles en la imagen. Es decir, aquellos granos de café que se encuentran detrás del rácimo en la foto.
- Si bien la lista de entregables no incluye un artículo científico, existe el acuerdo moral con el tutor de preparar un artículo científico en inglés para ser presentado a una revista que trate un tema afín, esto como máximo en los siguientes tres meses a la defensa.

## 4 Resultados

En esta sección se presentan los resultados de los experimentos y su respectivo análisis. Las métricas obtenidas por cada algoritmo, sobre el conjunto de imágenes proporcionado por ICAFE, con los parámetros seleccionados detallados en la sección 3.11 se resumen en la tabla 1. En esta tabla se resaltó en color verde y en negrita el mejor resultado obtenido en cada métrica. Los resultados de las métricas por imagen pueden ser consultados en el apéndice B.

	HOUGH + FAST	HOUGH	FAST	P-TRAP
Precision	<b>86.05%</b>	28.73%	71.97%	39.30%
Recall	75.70%	<b>95.44%</b>	77.10%	91.60%
Accuracy	<b>67.43%</b>	28.35%	59.30%	37.93%
f1-score	<b>80.54%</b>	44.17%	74.45%	55.00%

Tabla 1: Resultados de experimentos

De acuerdo con los datos mostrados en la tabla 1, el algoritmo *Hough Transform* obtuvo un buen resultado detectando granos ya que tiene un recall del 95%. Es decir, difícilmente deja un grano sin detectar. Sin embargo, realiza muchas detecciones falsas (falsos-positivos) por lo que la precisión y el accuracy son extremadamente bajos, apenas de un 28%. El f1-score, promedio ponderado de precisión y el recall, es bastante bajo, apenas de un 44%.

Por otro lado, *Fast Radial Symmetry Transform* tiene un recall y una precisión mayor al 70%. Tiene una mayor precisión en comparación a *Hough Transform*. Sin embargo, cuando se toma en cuenta la cantidad de granos no detectados y las falsas detecciones con la métrica accuracy obtenemos un resultado bajo, apenas un 59%. Si bien el f1-score es superior al obtenido por *Hough Transform*, sigue siendo bajo.

P-TRAP obtiene los resultados más bajos con excepción de la métrica recall, la cual es la segunda más alta. Cabe resaltar los bajos resultados obtenidos en las métricas precisión y accuracy, con apenas 39.30% y 37.93% respectivamente.

La combinación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* obtiene los mejores resultados con excepción de la métrica recall. Cabe destacar una precisión bastante alta, 86%, muy superior a la obtenida por los demás algoritmos.

## 4.1 Prueba de hipótesis

La figura 22 corresponde a un gráfico Q-Q [26], conocido también como gráfico cuantil-cuantil, donde se muestran las mediciones de f1-score obtenidas contra una distribución normal, representada por la línea diagonal. Es evidente como las mediciones de f1-score se desvían de la distribución normal.

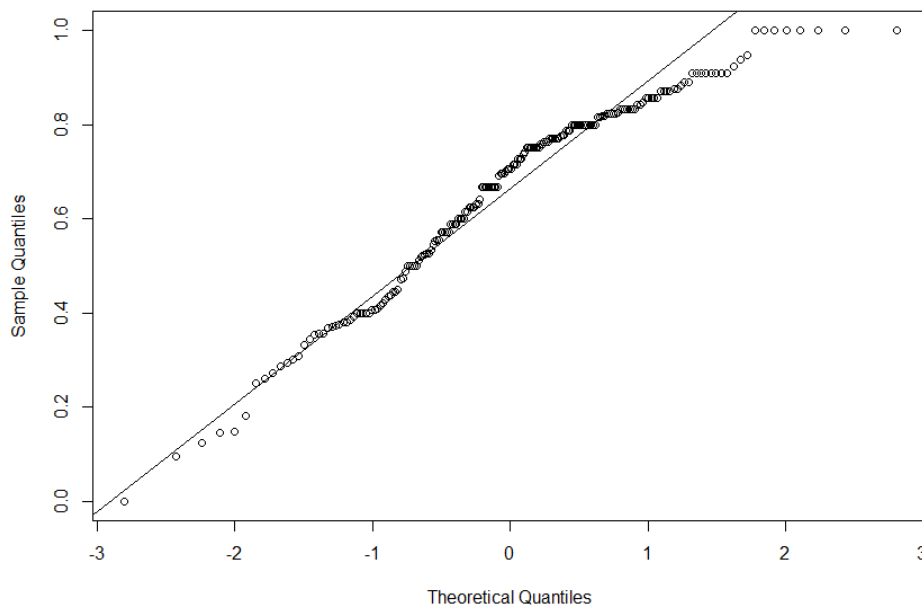


Figura 22: Gráfico Q-Q de la métrica f1-score

### 4.1.1 Prueba de Lilliefors

Para reforzar la observación presentada anteriormente se procedió a realizar una prueba de Lilliefors [22], la cual evalúa la hipótesis nula de que los valores están distribuidos normalmente, contra la hipótesis alternativa de que no lo están. El resultado de esta prueba se puede observar en la figura 23, donde se obtuvo un

valor  $p$  de 5.346e-08, el cual indica que los valores no siguen una distribución normal considerando un nivel de significancia de 5%.

```
Lilliefors (Kolmogorov-Smirnov) normality test
data: Q
D = 0.12454, p-value = 5.346e-08
```

Figura 23: Resultado de la prueba Lilliefors

Por esta razón se rechaza la hipótesis de que las mediciones de f1-score se distribuyen similarmente a una distribución normal. Dado que las mediciones no siguen una distribución normal, se optó por utilizar la prueba no paramétrica de Kruskal-Wallis [9] para la prueba de hipótesis.

#### 4.1.2 Prueba de Kruskal-Wallis

En la figura 24 se muestra el resultado de la prueba no paramétrica de Kruskal-Wallis [9] sobre las mediciones de f1-score. En este caso, se pone a prueba la hipótesis nula de que los datos vienen de la misma distribución, contra la hipótesis alternativa de que existe una diferencia verdaderamente significativa entre las poblaciones del conjunto de datos, que en este caso, corresponden a las mediciones de f1-score que fueron recolectadas.

```
Kruskal-Wallis rank sum test
data: Q by Algorithm
Kruskal-Wallis chi-squared = 66.149, df = 3, p-value = 2.848e-14
```

Figura 24: Resultado de la prueba Kruskal-Wallis

Como se observa en la figura 24, se obtuvo un valor  $p$  de 2.848e-14, muy cerca de cero y por debajo del nivel de significancia utilizado de 5%. Se concluye que existe evidencia estadística para el rechazo de la hipótesis nula planteada en Kruskal-Wallis. Es decir, sí existe una diferencia verdaderamente significativa entre las poblaciones del conjunto de datos.

### 4.1.3 Prueba de pares

Se procede a aplicar la prueba de pares *post-hoc* utilizando Nemenyi [24] con la distancia de Tukey. El objetivo de esta prueba es encontrar diferencias estadísticamente significativas entre las distribuciones de f1-score, para cada uno de los algoritmos. En la figura 25 se muestran los resultados obtenidos después de realizar esta prueba.

```
Pairwise comparisons using Tukey and Kramer (Nemenyi) test
with Tukey-Dist approximation for independent samples

data: mydata$Q and mydata$Algorithm

      Fast      Hough      Hough+Fast
Hough  2.8e-07 -          -
Hough+Fast 0.2686  2.2e-12 -
P-TRAP  0.0016  0.2605  3.1e-07
```

Figura 25: Resultado de la prueba de pares post-hoc Nemenyi

En la intercepción *Hough+Fast* y *P-TRAP*, se obtuvo un valor  $p$  de  $3.1e-07$ , casi cero y muy por debajo del nivel de significancia utilizado de 5%. Por lo tanto, existe suficiente evidencia estadística para rechazar la hipótesis nula planteada en la prueba Nemenyi. Esto nos indica que existe una diferencia significativa para las mediciones de f1-score, a la hora de comparar ambos algoritmos. Con esta información y la presentada en la tabla 1 donde la combinación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* obtiene un valor de f1-score de 80.54% contra un valor de 55% de P-TRAP se concluye que la combinación de algoritmos propuesta sí mejora el desempeño de la medida f1-score y por consiguiente se acepta la hipótesis planteada en la sección 3.4.

En la figura 25 también cabe resaltar las intersecciones entre *P-TRAP* y *Hough* y *Hough+Fast* y *Fast*. En la intersección *P-TRAP* y *Hough* se obtiene un valor  $p$  de 0.26, por encima del nivel de significancia de 5%. Por lo tanto, no existe una diferencia significativa para las mediciones de f1-score, a la hora de comparar ambos algoritmos. En la intersección *Hough* y *Hough+Fast* se obtiene un valor  $p$  de 0.0016, por debajo del nivel de significancia de 5%. Esto nos indica que existe una

diferencia significativa para las mediciones de f1-score, a la hora de comparar ambos algoritmos.

## 4.2 Análisis de resultados

Como se constató anteriormente, estadísticamente sí existen diferencias significativas, con un nivel de confianza del 95%, que determina la superioridad de la medida f1-score obtenida por la combinación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* sobre *P-TRAP*.

El algoritmo *Hough Transform* y *P-TRAP* logran detectar la gran mayoría de granos en las imágenes y por ello obtienen un recall bastante alto. Sin embargo, realizan una gran cantidad de detecciones falsas por lo que el resto de medidas, accuracy, precisión y f1-score obtienen resultados bastante pobres. Esto se debe a la cantidad de ruido en cada imagen del conjunto de datos. Al ser imágenes tomadas en un ambiente no controlado, cuentan con mucho ruido en el fondo y de acuerdo a lo explicado en las secciones 2.6 y 2.4, *Hough Transform* y *P-TRAP* son susceptibles a estos factores.

En la figura 26 se presenta un diagrama de caja en el cual se visualiza la distribución de f1-score para cada algoritmo. Podemos observar como la mediana de *Hough Transform* y la de *P-TRAP* están cerca y por debajo de *Fast Radial Symmetry Transform* y la combinación de *Fast Radial Symmetry Transform* y *Hough Transform*. Esto nos indica que *P-TRAP* obtiene un mejor desempeño en la detección que *Hough Transform*. También podemos ver que la mediana de *Fast Radial Symmetry Transform* está cerca de la mediana de la combinación de *Fast Radial Symmetry Transform* y *Hough Transform*. Esto nos indica que el algoritmo *Fast Radial Symmetry Transform* obtiene un desempeño similar a cuando se combina con *Hough Transform*, es decir, la combinación sí mejora el desempeño en la detección pero no de manera considerable.

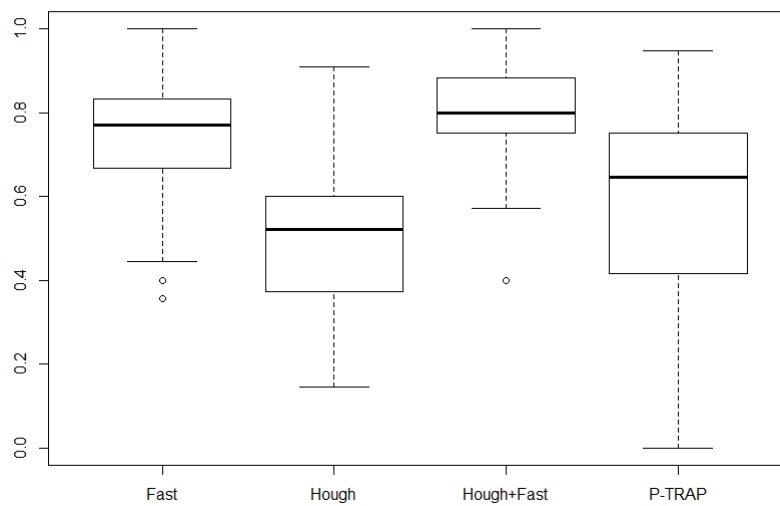


Figura 26: Diagrama de caja para las mediciones f1-score

## 5 Conclusiones y trabajo futuro

### 5.1 Conclusiones

En este trabajo de investigación se estudió mediante diferentes métricas la incorporación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform*, en combinación e individualmente, al proceso de detección de granos de la herramienta de código abierto *P-TRAP* y se planteó la siguiente hipótesis de investigación:

*La incorporación de los algoritmos Fast Radial Symmetry Transform y Hough Transform al proceso propuesto por P-TRAP, mejora el desempeño, medido por el promedio armónica de precisión y de exhaustividad, en la detección de granos de café existentes en una imagen.*

En la sección 4 se presentaron los resultados de los experimentos realizados. En estos resultados se destaca el puntaje obtenido en la métrica *f1-score*, 80.54%, por parte de la combinación *Fast+Hough* sobre el obtenido por *P-TRAP*, 55%. Seguidamente, se presentan los resultados de las pruebas estadísticas realizadas que permiten, con un nivel de confianza del 95%, aceptar la hipótesis de investigación planteada.

Con el fin de llevar a cabo esta investigación se desarrolló lo siguiente:

- Utilizando el lenguaje de programación Java, se realiza de manera exitosa la programación del algoritmo *Fast Radial Symmetry Transform* y del algoritmo *Hough Transform*. Esta implementación de los algoritmos nos permite realizar pruebas preliminares para su posterior incorporación a la herramienta *P-TRAP*.
- Después de un análisis minucioso de las dependencias, se realizó la incorporación de los algoritmos *Fast Radial Symmetry Transform* y del



algoritmo *Hough Transform* al proceso de detección de granos de la herramienta *P-TRAP*.

- Se creó un ambiente de pruebas para la ejecución de la fase experimental de esta investigación.
- Utilizando el ambiente de pruebas se llevó a cabo la ejecución de los experimentos con el propósito de recolectar las métricas *f1-score*, exactitud, precisión y exhaustividad.
- Mediante la prueba de hipótesis se comparó el *f1-score* entre todos los algoritmos sobre el conjunto de imágenes proporcionado por ICAFE.
- Se entrega un análisis detallado de los resultados obtenidos a partir de los experimentos realizados.

Como resultado de este procedimiento, se obtuvo una mejora significativa en el nivel de *f1-score* entre la versión predeterminada de *P-TRAP* y la versión desarrollada en este proyecto de investigación que combina los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform*. Asimismo, se observó una mejora significativa del algoritmo *Fast Radial Symmetry Transform* individualmente. A partir de los resultados recolectados y posteriormente su análisis se obtienen las siguientes conclusiones:

1. Es posible realizar detecciones de granos de café en imágenes tomadas en un ambiente no controlado y con presencia de ruido.
2. El algoritmo *Hough Transform* y la herramienta *P-TRAP* son susceptibles al ruido en la imagen. Esto genera una gran cantidad de detecciones falsas. Esto se observó de manera repetitiva en cada imagen del conjunto de datos y mediante el análisis de la métrica *f1-score*.
3. El algoritmo *Hough Transform* y la herramienta *P-TRAP* son capaces de detectar la gran mayoría de los granos existentes en una imagen. Esto fue observado mediante los resultados obtenidos por la métrica *exhaustividad*.

4. La combinación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* obtiene un mejor desempeño, medido por la métrica *f1-score*, en comparación con *P-TRAP*.
5. El algoritmo *Fast Radial Symmetry Transform* individualmente obtiene un desempeño similar al obtenido cuando es combinado con el algoritmo *Hough Transform*.
6. Si bien la combinación de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform* obtuvo el mejor resultado de *f1-score* en los experimentos con un 80.54% aún es un puntaje bajo para la puesta en práctica y su potencial uso en la industria.

Es de interés hacer de uso público la versión de *P-TRAP* desarrollada en este trabajo. Por esta razón se iniciaron conversaciones con los desarrolladores de la herramienta con el fin de realizar este aporte a la versión oficial.

## 5.2 Trabajo Futuro

A continuación se propone una serie de acciones y temas a realizar en un futuro para la detección y conteo de granos de café en imágenes:

- Analizar cómo es posible integrar el trabajo desarrollado en esta investigación con el trabajo propuesto en [27]. En dicho trabajo se propone una estrategia de detección de cinco fases y donde se detectan granos de café mediante el análisis de los arcos presentes en la imagen. Existen algunos tipos de imágenes que la solución propuesta en [27] no logra detectar pero lo desarrollado en este trabajo sí. Por ejemplo, cuando la rama no está en el centro de la imagen.
- Estimación de la cantidad de granos de café que están detrás de la rama. Mediante el uso de aprendizaje máquina sería interesante poder estimar la cantidad de granos que existen detrás de un rácimo a partir de los rasgos y características de la parte frontal que se puede observar en la imagen.
- Detección de granos de café traslapados. Los granos que se encuentran parcialmente traslapados pierden su figura circular y solo una porción es visible, dificultando el proceso de detección. Mediante el análisis de la posición de los granos y la textura, potencialmente se podría determinar la presencia de un grano traslapado.
- La recolección de imágenes es un proceso difícil ya que las plantaciones de café pueden ser grandes y densas. Sería útil el desarrollo de una aplicación móvil que analice la calidad de las imágenes recolectadas en tiempo real. Esto con el fin de informar al usuario en el sitio si existe la necesidad de realizar otras tomas y se evite otra visita al campo para recolectar nuevamente las imágenes porque las imágenes tomadas inicialmente no cuentan con la calidad deseada.

# Bibliografía

- [1] D. C. Montgomery. *"Guidelines for designing experiments, design and analysis of experiments."* 5th Edition, 2000.
- [2] Reinhard Klette. *"Concise Computer Vision: An Introduction into Theory and Algorithms"* First Edition, 2014.
- [3] Periasamy, A. *"Matching of Different Rice Grains Using Digital Image Processing"*, 2013.
- [4] El-Helly, M, Ahmed, R, El-Gammal, S. *"An Integrated Image Processing System for Leaf Disease Detection and Diagnosis"*, 2003.
- [5] Wang, W., Wang, Y., Ji, T. *"Grains Automatic Counting Method Based on Computer Vision"*, 2012.
- [6] Nuske, S, Achar, S, Bates, T, Narasimhan, S, Singh, S. *Yield Estimation in Vineyards by Visual Grape Detection*, 2011.
- [7] Kumar, N, Belhumeur, P, Biswas, A, Jacobs, D, Kress, W, Lopez, I, Soares, J. *Leafsnap: A Computer Vision System for Automatic Plant Species Identification*, 2012.
- [8] Loy, G., Zelinsky, A. *"A Fast Radial Symmetry Transform for Detecting Points of Interest"*, 2002.
- [9] William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *"Journal of the American Statistical Association"*, 1952.

- [10] Montaña, M. *Estadística no paramétrica para la toma de decisiones del ingeniero industrial en la industria maquiladora*.
- [11] MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.
- [12] Netbeans IDE. URL <http://www.netbeans.org>.
- [13] AL-Tam, F, Adam, H, dos Anjos, A, Lorieux, M, Larmande, P, Ghesquiere, A, Jouannic, S, Reza, H. "*P-TRAP: a Panicle Trait Phenotyping tool*", 2013.
- [14] Platero, C. "*Apuntes de Visión Artificial*", 2009.
- [15] Newhall, B. "*Historia de la Fotografía*", 2002.
- [16] Instituto del Cafe de Costa Rica. <http://www.icafe.cr/>
- [17] Tanabata, T, Shibaya, T, Hori, K, Ebana, K, Yano, M. *SmartGrain: High-Throughput Phenotyping Software for Measuring Seed Shape through Image Analysis*, 2012.
- [18] Suzuki, S, Abe, K. *Topological Structural Analysis of Digitized Binary Images by Border Following*. CVGIP 30 1, 1985.
- [19] P.V.C. Hough. *Method and means for recognizing complex patterns*, 1962. U.S. patent 3069654.
- [20] Bradski, G. *The OpenCV Library*, 2000.
- [21] Canny, J. *A computational approach to edge detection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6), 1986.
- [22] Lilliefors, H. *On the Kolmogorov-Smirnov test for normality with mean and variance unknown*. Journal of the American statistical Association, 1967.
- [23] Gross, J. *Package Nortest*, 2015.
- [24] Nemenyi, P. *Distribution-free Multiple Comparisons*. Princeton University, 1963.

- [25] Yuen, K, Princen, J, Illingworth, J, Kittler, J. *Comparative study of Hough transform methods for circle finding*. Image Vision Comput. 8 1, 1990.
- [26] Wilk, M, Gnanadesikan, R. *Probability plotting methods for the analysis of data*. Biometrika, 1968.
- [27] Ramos, P.J, Prieto, F.A, Montoya, E.C, Oliveros, C.E. *Automatic fruit count on coffee branches using computer vision*, 2017.
- [28] Venables, W.N, Smith D.M. *An Introduction to R*, 2017.
- [29] Howell, D. *Statistical Methods for Psychology*, 2002.
- [30] Fisher, R. *The Correlation Between Relatives on the Supposition of Mendelian Inheritance*. Philosophical Transactions of the Royal Society of Edinburgh, volume 52, 1918.
- [31] Pohlert, T. *Package PMCMR*, 2016.
- [32] ICAFE. *Informe sobre la actividad cafetalera de Costa Rica*, 2016.
- [33] Linda G. Shapiro, George C. Stockman (2001): *Computer Vision*, pp 279-325, New Jersey, Prentice-Hall.
- [34] Barghout, Lauren, Lawrence W. Lee. *Perceptual information processing system.*, 2003.
- [35] Gomez, W. *Reconocimiento de objetos en fotografías*, 2015.

# Apéndice A

## Selección de parámetros

# 1 Tabla de selección de parámetros para cada imagen

<b>Imagen</b>	<b>Radio Min</b>	<b>Radio Max</b>
granos.1	25	35
granos.2	45	50
granos.3	25	36
granos.4	20	35
granos.5	35	40
granos.6	25	40
granos.7	16	30
granos.8	45	55
granos.9	35	40
granos.10	35	40
granos.11	35	40
granos.12	45	50
granos.13	55	60
granos.14	40	50
granos.15	27	35
granos.16	30	40
granos.17	30	40
granos.18	30	35
granos.19	30	35
granos.20	30	35
granos.21	24	30
granos.22	25	35
granos.23	60	70
granos.24	20	40
granos.25	30	35
granos.26	35	40
granos.27	35	40
granos.28	27	36
granos.29	30	35
granos.30	30	35
granos.31	30	40
granos.32	30	35
granos.33	40	45
granos.34	30	35
granos.35	30	40
granos.36	35	40
granos.37	30	40
granos.38	50	60
granos.39	25	45
granos.40	25	45
granos.41	45	50
granos.42	20	40
granos.43	40	50
granos.44	25	40
granos.45	27	45
granos.46	35	45
granos.47	25	35
granos.48	30	40
granos.49	40	45
granos.50	25	40



# Apéndice B

## Resultados de experimentos

# 1 Tabla de resultados para Hough Transform

Imagen	HOUGH			
	Precision	Recall	Exactitud	f1-score
granos_1	0.833	1.000	0.833	0.909
granos_2	0.429	1.000	0.429	0.600
granos_3	0.254	1.000	0.254	0.405
granos_4	0.257	1.000	0.257	0.409
granos_5	0.824	0.933	0.778	0.875
granos_6	0.200	1.000	0.200	0.333
granos_7	0.078	1.000	0.078	0.145
granos_8	0.818	0.900	0.750	0.857
granos_9	0.231	1.000	0.231	0.375
granos_10	0.375	1.000	0.375	0.545
granos_11	0.400	1.000	0.400	0.571
granos_12	0.105	0.667	0.100	0.182
granos_13	0.800	1.000	0.800	0.889
granos_14	0.235	1.000	0.235	0.381
granos_15	0.227	1.000	0.227	0.370
granos_16	0.235	0.889	0.229	0.372
granos_17	0.286	1.000	0.286	0.444
granos_18	0.455	1.000	0.455	0.625
granos_19	0.364	0.800	0.333	0.500
granos_20	0.500	0.667	0.400	0.571
granos_21	0.471	1.000	0.471	0.640
granos_22	0.280	1.000	0.280	0.438
granos_23	0.429	1.000	0.429	0.600
granos_24	0.355	1.000	0.355	0.524
granos_25	0.364	1.000	0.364	0.533
granos_26	0.357	1.000	0.357	0.526
granos_27	0.278	1.000	0.278	0.435
granos_28	0.462	1.000	0.462	0.632
granos_29	0.417	0.833	0.385	0.556
granos_30	0.571	0.889	0.533	0.696
granos_31	0.158	1.000	0.158	0.273
granos_32	0.417	1.000	0.417	0.588
granos_33	0.727	0.800	0.615	0.762
granos_34	0.714	0.714	0.556	0.714
granos_35	0.225	1.000	0.225	0.367
granos_36	0.214	1.000	0.214	0.353
granos_37	0.382	0.813	0.351	0.520
granos_38	0.357	1.000	0.357	0.526
granos_39	0.176	1.000	0.176	0.300
granos_40	0.217	1.000	0.217	0.357
granos_41	0.714	1.000	0.714	0.833
granos_42	0.212	0.917	0.208	0.344
granos_43	0.385	1.000	0.385	0.556
granos_44	0.353	1.000	0.353	0.522
granos_45	0.172	1.000	0.172	0.294
granos_46	0.290	1.000	0.290	0.450
granos_47	0.255	1.000	0.255	0.406
granos_48	0.238	1.000	0.238	0.385
granos_49	0.462	1.000	0.462	0.632
granos_50	0.167	1.000	0.167	0.286

## 2 Tabla de resultados para Fast Radial Symmetry Transform

	FAST			
Imagen	Precision	Recall	Exactitud	f1-score
granos_1	0.800	0.800	0.667	0.800
granos_2	1.000	1.000	1.000	1.000
granos_3	0.917	0.733	0.688	0.815
granos_4	1.000	0.611	0.611	0.759
granos_5	0.917	0.733	0.688	0.815
granos_6	0.833	0.714	0.625	0.769
granos_7	1.000	0.667	0.667	0.800
granos_8	0.667	0.667	0.500	0.667
granos_9	0.600	1.000	0.600	0.750
granos_10	0.750	1.000	0.750	0.857
granos_11	0.286	1.000	0.286	0.444
granos_12	0.667	0.667	0.500	0.667
granos_13	0.667	1.000	0.667	0.800
granos_14	0.778	0.875	0.700	0.824
granos_15	0.571	0.800	0.500	0.667
granos_16	0.857	0.667	0.600	0.750
granos_17	1.000	0.700	0.700	0.824
granos_18	0.714	1.000	0.714	0.833
granos_19	0.600	0.600	0.429	0.600
granos_20	0.500	0.667	0.400	0.571
granos_21	1.000	0.625	0.625	0.769
granos_22	0.857	0.857	0.750	0.857
granos_23	0.400	0.667	0.333	0.500
granos_24	1.000	0.455	0.455	0.625
granos_25	0.444	1.000	0.444	0.615
granos_26	0.417	1.000	0.417	0.588
granos_27	1.000	1.000	1.000	1.000
granos_28	0.500	0.667	0.400	0.571
granos_29	0.833	0.833	0.714	0.833
granos_30	0.529	1.000	0.529	0.692
granos_31	0.750	1.000	0.750	0.857
granos_32	0.714	1.000	0.714	0.833
granos_33	0.727	0.800	0.615	0.762
granos_34	0.238	0.714	0.217	0.357
granos_35	1.000	0.667	0.667	0.800
granos_36	0.500	0.333	0.250	0.400
granos_37	0.727	1.000	0.727	0.842
granos_38	0.625	1.000	0.625	0.769
granos_39	0.909	0.833	0.769	0.870
granos_40	0.857	0.600	0.545	0.706
granos_41	0.833	1.000	0.833	0.909
granos_42	0.909	0.909	0.833	0.909
granos_43	0.750	0.600	0.500	0.667
granos_44	1.000	0.833	0.833	0.909
granos_45	0.750	0.600	0.500	0.667
granos_46	0.875	0.700	0.636	0.778
granos_47	1.000	0.615	0.615	0.762
granos_48	0.750	0.900	0.692	0.818
granos_49	0.400	0.667	0.333	0.500
granos_50	1.000	0.667	0.667	0.800

### 3 Tabla de resultados para P-TRAP

	P-TRAP			
Imagen	Precision	Recall	Exactitud	f1-score
granos_1	0.091	0.100	0.050	0.095
granos_2	0.083	0.667	0.080	0.148
granos_3	0.625	1.000	0.625	0.769
granos_4	0.704	1.000	0.704	0.826
granos_5	0.882	1.000	0.882	0.938
granos_6	0.700	1.000	0.700	0.824
granos_7	0.600	1.000	0.600	0.750
granos_8	0.310	1.000	0.310	0.474
granos_9	0.143	1.000	0.143	0.250
granos_10	0.500	1.000	0.500	0.667
granos_11	0.182	1.000	0.182	0.308
granos_12	0.000	0.000	0.000	0.000
granos_13	0.750	0.750	0.600	0.750
granos_14	0.533	1.000	0.533	0.696
granos_15	0.500	1.000	0.500	0.667
granos_16	0.400	0.889	0.381	0.552
granos_17	0.345	1.000	0.345	0.513
granos_18	0.455	1.000	0.455	0.625
granos_19	0.417	1.000	0.417	0.588
granos_20	0.533	1.000	0.533	0.696
granos_21	0.636	0.875	0.583	0.737
granos_22	0.538	1.000	0.538	0.700
granos_23	0.263	0.833	0.250	0.400
granos_24	0.833	0.909	0.769	0.870
granos_25	0.600	0.750	0.500	0.667
granos_26	0.263	1.000	0.263	0.417
granos_27	0.250	1.000	0.250	0.400
granos_28	0.600	1.000	0.600	0.750
granos_29	0.545	1.000	0.545	0.706
granos_30	0.900	1.000	0.900	0.947
granos_31	0.273	1.000	0.273	0.429
granos_32	0.556	0.833	0.500	0.667
granos_33	0.323	1.000	0.323	0.488
granos_34	0.286	0.571	0.235	0.381
granos_35	0.276	0.889	0.267	0.421
granos_36	0.067	1.000	0.067	0.125
granos_37	0.343	0.750	0.308	0.471
granos_38	0.250	0.900	0.243	0.391
granos_39	0.632	1.000	0.632	0.774
granos_40	0.588	1.000	0.588	0.741
granos_41	0.556	1.000	0.556	0.714
granos_42	0.714	0.909	0.667	0.800
granos_43	0.250	1.000	0.250	0.400
granos_44	0.810	0.944	0.773	0.872
granos_45	0.417	1.000	0.417	0.588
granos_46	0.333	1.000	0.333	0.500
granos_47	0.650	1.000	0.650	0.788
granos_48	0.714	1.000	0.714	0.833
granos_49	0.150	1.000	0.150	0.261
granos_50	0.4	0.666667	0.333333	0.5

#### 4 Tabla de resultados para Hough Transform + Fast Radial Symmetry Transform

	HOUGH + FAST			
Imagen	Precision	Recall	Exactitud	f1-score
granos_1	0.889	0.800	0.727	0.842
granos_2	1.000	1.000	1.000	1.000
granos_3	0.846	0.733	0.647	0.786
granos_4	1.000	0.611	0.611	0.759
granos_5	1.000	0.733	0.733	0.846
granos_6	0.833	0.714	0.625	0.769
granos_7	1.000	0.667	0.667	0.800
granos_8	0.833	0.625	0.556	0.714
granos_9	0.750	1.000	0.750	0.857
granos_10	1.000	1.000	1.000	1.000
granos_11	0.500	1.000	0.500	0.667
granos_12	1.000	0.667	0.667	0.800
granos_13	1.000	1.000	1.000	1.000
granos_14	0.875	0.778	0.700	0.824
granos_15	1.000	0.800	0.800	0.889
granos_16	0.857	0.667	0.600	0.750
granos_17	1.000	0.700	0.700	0.824
granos_18	0.714	1.000	0.714	0.833
granos_19	0.600	0.600	0.429	0.600
granos_20	0.857	0.750	0.667	0.800
granos_21	1.000	0.625	0.625	0.769
granos_22	1.000	0.857	0.857	0.923
granos_23	0.500	0.667	0.400	0.571
granos_24	1.000	0.455	0.455	0.625
granos_25	0.667	1.000	0.667	0.800
granos_26	0.833	1.000	0.833	0.909
granos_27	1.000	1.000	1.000	1.000
granos_28	0.571	0.667	0.444	0.615
granos_29	1.000	0.833	0.833	0.909
granos_30	0.727	0.889	0.667	0.800
granos_31	1.000	1.000	1.000	1.000
granos_32	0.833	1.000	0.833	0.909
granos_33	0.875	0.700	0.636	0.778
granos_34	1.000	0.571	0.571	0.727
granos_35	1.000	0.667	0.667	0.800
granos_36	0.500	0.333	0.250	0.400
granos_37	0.765	0.813	0.650	0.788
granos_38	0.714	1.000	0.714	0.833
granos_39	0.909	0.833	0.769	0.870
granos_40	0.857	0.600	0.545	0.706
granos_41	1.000	1.000	1.000	1.000
granos_42	0.909	0.909	0.833	0.909
granos_43	1.000	0.600	0.600	0.750
granos_44	0.938	0.833	0.789	0.882
granos_45	1.000	0.600	0.600	0.750
granos_46	1.000	0.778	0.778	0.875
granos_47	0.889	0.615	0.571	0.727
granos_48	0.750	0.900	0.692	0.818
granos_49	0.800	0.667	0.571	0.727
granos_50	1.000	0.667	0.667	0.800

# Apéndice C

## Conjunto de datos y código fuente

# 1 Imágenes y código fuente

En este enlace se puede acceder al conjunto de imágenes utilizado para la ejecución de los experimentos así como el código fuente de las implementaciones de los algoritmos *Fast Radial Symmetry Transform* y *Hough Transform*.

<https://drive.google.com/open?id=0B-dFsYianGXZdk1SMXJCR3d3cVk>