

DETERMINACIÓN DE LA DISTRIBUCIÓN DE LOS VALORES DE PÍXEL
PRESENTES EN LAS IMÁGENES ASTRONÓMICAS CAPTURADAS DESDE EL
OBSERVATORIO DE LA UNIVERSIDAD TECNOLÓGICA DE PEREIRA

JUAN SEBASTIÁN ARIAS HERNÁNDEZ

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍA INDUSTRIAL
INGENIERÍA INDUSTRIAL
PEREIRA

2014

DETERMINACIÓN DE LA DISTRIBUCIÓN DE LOS VALORES DE PÍXEL
PRESENTES EN LAS IMÁGENES ASTRONÓMICAS CAPTURADAS DESDE EL
OBSERVATORIO DE LA UNIVERSIDAD TECNOLÓGICA DE PEREIRA

JUAN SEBASTIÁN ARIAS HERNÁNDEZ

Trabajo de Grado para Optar al Título de Ingeniero Industrial

Director:

MSc. EDWIN ANDRÉS QUINTERO SALAZAR

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍA INDUSTRIAL

INGENIERÍA INDUSTRIAL

PEREIRA

2014

Nota de Aceptación:

Ingeniero Orlando Antonio Sabogal
Jurado

MSc. Edwin Andrés Quintero
Director

*“A mi familia y novia; por su cósmica
paciencia e infinito apoyo ”*

AGRADECIMIENTOS

El presente trabajo investigativo no hubiera visto la luz sin los consejos, sugerencias y recomendaciones que miembros de la academia y familiares brindaron durante la realización de este.

Especial mención al MSc. Edwin Andrés Quintero Salazar, por su compromiso y disposición; por compartir las ideas que convergerían en el presente documento.

A los profesores José Adalberto Soto y Elkin López Rendón por aclarar mis dudas cuando surgieron.

Al Ingeniero Orlando Antonio Sabogal por prestarse voluntariamente a la revisión del documento.

A Mis padres; hermanos y hermanita; sobrinos y novia. Por darle brillo y color, a un mundo gris y sombrío.

GLOSARIO

Arcosegundo: unidad de medida angular equivalente a $1/3600$ de grado.

Ascensión recta: Una de las dos coordenadas ecuatoriales absolutas y se define como el ángulo contado sobre el ecuador celeste a partir del punto vernal en dirección este hasta el círculo de declinación del astro.

Astrometría: Es la rama de la astronomía dedicada al estudio de las posiciones, distancias y movimientos de los cuerpos celestes con el fin de obtener sus coordenadas y sus variaciones en el tiempo.

Corriente de oscuridad: Parámetro de especificación de los sensores CCD correspondientes a la cantidad de electrones generados por segundo por pixel.

Corriente oscura ruido: Variación estadística de la corriente de oscuridad.

Cuentas: Valor numérico discreto registrado en las imágenes digitales luego de la conversión analógico digital.

Dark frame o dark: Fotografía tomada con el obturador de la cámara cerrado, registrando de esta manera el ruido producido por la electrónica de la cámara.

Declinación: Es la segunda de las dos coordenadas ecuatoriales absolutas y corresponde al ángulo medido sobre el círculo de declinación del astro y que empieza a contarse a partir del ecuador celeste.

Fondo de cielo: Es el brillo de la esfera celeste en las noches, debido principalmente a fuentes como la contaminación lumínica.

Fotometría: La fotometría es la ciencia que estudia la medida del brillo de la luz en longitudes de onda percibidas por el ojo humano. Esta medida en astronomía ayuda a construir información del cuerpo de estudio tal como: temperatura del objeto, distancia, edad entre otras.

Obturador: Dispositivo de las cámaras fotográficas que permite o no el ingreso de luz.

Pixel: Cada uno de los elementos indivisibles y más pequeños que componen la imagen.

Ruido: Se considera ruido, en astrofotografía con *CCD*, a los electrones generados por el sensor por fuentes diferentes a los fotones que provienen del cuerpo de estudio.

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	1
1. PRELIMINARES.....	3
1.1 DESCRIPCIÓN DEL PROBLEMA	3
1.2 JUSTIFICACIÓN	5
1.3 OBJETIVO GENERAL	6
1.4 OBJETIVOS ESPECÍFICOS	6
1.5 ESTADO DEL ARTE	7
2. MARCO TEÓRICO.....	13
2.1 VARIABLE ALEATORIA	13
2.2 FUNCIÓN DE DISTRIBUCIÓN PARA UNA VARIABLE ALEATORIA	13
2.3 FUNCIÓN DE DENSIDAD PARA UNA VARIABLE ALEATORIA.....	13
2.4 DISTRIBUCIÓN DE POISSON	14
2.5 DISTRIBUCIÓN NORMAL	14
2.6 DISTRIBUCIÓN WEIBULL.....	16
2.7 DISTRIBUCIÓN GAMMA.....	16
2.8 DISTRIBUCIÓN LOG-NORMAL	17
2.9 PRUEBAS DE BONDAD Y AJUSTE.....	18
2.10 PRUEBA KOLMOGOROV-SMIRNOV	18
2.11 PRUEBA <i>CHI-CUADRADO</i>	20
2.12 ANÁLISIS DE CLÚSTER	21
2.13 PROMEDIO ENTRE GRUPOS	21
2.14 MÉTODO DEL CENTROIDE	22
2.15 MÉTODO DEL PROMEDIO GENERALIZADO	23
2.16 MÉTODO DEL CENTROIDE GENERALIZADO	25
2.17 MUESTREO ALEATORIO SIMPLE	27
2.18 MUESTREO SISTEMÁTICO.....	28
2.19 IMÁGENES DIGITALES	29

3. METODOLOGÍA.....	30
3.1 DESARROLLO DE LOS ALGORITMOS.....	30
3.2 RECOLECCIÓN DE LA INFORMACIÓN	30
3.2.1 Dark frames	31
3.2.2 Procedimiento para la obtención de los Darks	31
3.2.3 Recolección de las fotos para las estrellas y el fondo de cielo	32
3.2.4 Procedimiento para la toma de <i>Alnitak</i>	33
3.3 INSTRUMENTACIÓN	33
3.4 CONSIDERACIONES DE LAS PRUEBAS	34
3.4.1 Para los darks.....	34
3.4.2 Para la distribución del fondo de cielo y la estrella	35
3.4.3 Para la distancia entre dos estrellas	36
3.5 METODOLOGÍAS DE LAS PRUEBAS	40
3.5.1 Pruebas realizadas sobre las imágenes darks	41
3.5.1.1 Pruebas de bondad y ajuste a un vector construido con muestras tomadas de cada imagen	41
3.5.1.1.1 Manejo de atípicos	42
3.5.1.2 Bondad y ajuste para valores registrados por un mismo pixel.....	47
3.5.2 Distribución fondo de cielo y de estrella.....	51
3.5.3 Distancia entre objetos para determinar las coordenadas de uno de ellos	56
3.5.3.1 Método de los promedios	57
3.5.3.2 Método del centroide	58
4. RESULTADOS	61
4.1 RESULTADOS DEL POSIBLE AJUSTE A UNA DISTRIBUCIÓN DE LA CORRIENTE DE OSCURIDAD.....	62
4.1.1 Resultados para la distribución de todo el sensor	62
4.1.1.1 Resultados para la primera jornada.....	62
4.1.1.2 Para la segunda jornada.....	63
4.1.1.3 Para la tercera jornada	64
4.1.2 Pruebas para los pixeles particulares	66
4.1.2.1 Resultados de la primera jornada.....	66

4.1.2.2 Para los pixeles en particular segunda jornada	67
4.1.2.3 Para los pixeles en particular tercera jornada.....	68
4.2. RESULTADOS PARA EL FONDO DEL CIELO	70
4.2.1 Distribución para el fondo de cielo. Prueba <i>Chi</i>	70
4.2.2 Resultado Prueba K_S para fondo de cielo	71
4.3 RESULTADOS PARA LA DISTRIBUCIÓN DE LA ESTRELLA.....	71
4.4 CÁLCULO DE LAS COORDENADAS POR AMBOS MÉTODOS.....	72
4.4.1 Resultado dispersión para las declinaciones	74
4.4.2 Resultado dispersión para las Ascensión recta	75
CONCLUSIONES.....	76
RECOMENDACIONES	77
BIBLIOGRAFÍA	78
ANEXOS	80

ÍNDICE DE FIGURAS

	Pág.
Figura 1. Fragmento de un Dark Frame.	3
Figura 2. Fotografía de una estrella adquirida desde OAUTP.	4
Figura 3. Función de densidad de Poisson.	15
Figura 4. Función de Densidad Normal.	15
Figura 5. Función de Densidad Weibull.	16
Figura 6. Función de Densidad Gamma.	17
Figura 7. Función de Densidad Lognormal.	18
Figura 8. Comparación Función teórica y Empírica para el test KS.	19
Figura 9. Metodología del Promedio Entre Grupos.	21
Figura 10. Metodología del Centroide.	22
Figura 11. Fotografía del cúmulo globular Omega Centauri tomada desde el observatorio astronómico de la Universidad Tecnológica de Pereira.	29
Figura 12. Segmento de imagen seleccionada en el que se aprecia la estrella y el fondo de cielo.	35
Figura 13. Comparación de la posición de las estrellas.	36
Figura 14. Cálculo del ángulo α	37
Figura 15. Ángulo α en la imagen tomada desde OAUTP.	38
Figura 16. Diagrama de flujo para la adecuada posición de la imagen.	39
Figura 17. Secuencia de los ajustes realizados a la imagen.	40
Figura 18. Histograma con valores de pixeles menores o iguales a 5000 cuentas.	42
Figura 19. Histograma con valores de pixeles menores a 1600 cuentas.	44
Figura 20. Diagrama de flujo. Prueba de bondad y ajuste para muestras tomadas de cada imagen.	46
Figura 21. Diagrama de flujo para los valores registrados por un mismo pixel. ...	49
Figura 22. Fragmento de imagen utilizada para identificar el posible ajuste del fondo de cielo y de la estrella.	51
Figura 23. Diagrama flujo de los pasos para identificar la distribución del fondo de cielo.	53
Figura 24. Diagrama flujo de los pasos para identificar la distribución de la estrella.	54
Figura 25. Diagrama de flujo método de los promedios.	59
Figura 26. Diagrama de flujo. Método del centroide.	60
Figura 27. Síntesis de las pruebas a realizar.	61
Figura 28. Comparación entre el estadístico calculado y el crítico para la prueba Chi primera jornada.	62

Figura 29. Comparación del estadístico calculado y el crítico para la prueba KS primera jornada.	63
Figura 30. Comparación del estadístico calculado y el crítico para la prueba Chi segunda jornada.....	63
Figura 31. Comparación del estadístico calculado y el crítico de la prueba KS para la segunda jornada.....	64
Figura 32. Comparación del estadístico calculado y el crítico de la prueba Chi para la tercera jornada.	65
Figura 33. Comparación del estadístico calculado y el crítico de la prueba KS para la tercera jornada.	65
Figura 34. Comparación del estadístico calculado y el crítico de la prueba Chi para pixeles particulares en la primera jornada (la Tabla 3 indica la posición de cada píxel).....	66
Figura 35. Comparación del estadístico calculado y el crítico de la prueba KS para pixeles particulares en la primera jornada.....	67
Figura 36. Comparación del estadístico calculado y el crítico de la prueba Chi para pixeles particulares en la segunda jornada.	67
Figura 37. Comparación del estadístico calculado y el crítico de la prueba KS para pixeles particulares en la segunda jornada.	68
Figura 38. Comparación del estadístico calculado y el crítico de la prueba Chi para pixeles particulares en la tercera jornada.....	68
Figura 39. Comparación del estadístico calculado y el crítico de la prueba KS para pixeles particulares en la tercera jornada.....	69
Figura 40. Comparación del estadístico calculado y el crítico de la prueba Chi para el fondo de cielo.	70
Figura 41. Comparación del estadístico calculado y el crítico de la prueba KS para el fondo de cielo.	71
Figura 42. Comparación del estadístico calculado y el crítico de la prueba Chi para la estrella.....	71
Figura 43. Comparación del estadístico calculado y el crítico de la prueba KS para la estrella.	72
Figura 44. Objetos seleccionados para realizar las pruebas.....	73
Figura 45. Coordenadas obtenidas con ambos métodos.	73
Figura 46. Diagrama dispersión para la declinación.....	74
Figura 47. Diagrama dispersión para la ascensión recta.	75

ÍNDICE DE TABLAS

	Pág.
Tabla 1. Método para la identificación de atípicos.....	43
Tabla 2. Frecuencias resultantes al eliminar los atípicos.	45
Tabla 3. Aleatorios.	48
Tabla 4. Errores declinación (grados); método del promedio.	74
Tabla 5. Errores en ascensión recta (horas); método del promedio.....	74
Tabla 6. Errores declinación (grados); método del centroide.	74
Tabla 7. Errores en ascensión recta (horas); método del centroide.	74

ANEXOS

	Pág.
Anexo 1. Código para determinar el comportamiento de valores tomados de diferentes imágenes.	80
Anexo 1.1 Prueba_Funciones.	80
Anexo 1.2 Tama_muestra.	81
Anexo 1.3 aleat_sin_rept.	83
Anexo 1.4 Coor_Pixel.	84
Anexo 1.5 Misma_muestra.	85
Anexo 1.6 prueba_bya.	87
Anexo1.7 Frecuencia_intervalos.	90
Anexo 1.8 K_S_Nor.	92
Anexo 1.9 K_S_Weibull.	93
Anexo 1.10 K_S_Gamma.	94
Anexo 1.11 K_S_Poss.	94
Anexo 1.12 Chi_Cuad_Norm.	95
Anexo 1.13 Chi_Cuad_gamma.	97
Anexo 1.4 Chi_Cuad_weibull.	99
Anexo 1.15 Chi_Cua_Poiss.	100
Anexo 1.16 intervalo_corre.	101
Anexo 2 Código desarrollado para identificar las posibles distribuciones de los pixeles en particular.	103
Anexo 2.1 Val_mismo_pixel.	103
Anexo 2.2 Mismos_pixels.	105
Anexo 2.3 Res_Chi.	106
Anexo 3 Método de los promedios para el cálculo de las distancias entre los clústers.	109
Anexo 3.1 Promedios.	109
Anexo 3.2 posicion_Promedio.	114
Anexo 3.3 Conversor1.	115

Anexo 4. Método del centroide para las coordenadas y el cálculo de las distancias.	118
Anexo 4.1 Centroide.	118
Anexo 4.2 posicion_centro.	121
Anexo 4.3 Conversor.	122
Anexo 5. Código para la estimación de la distribución del fondo de cielo.	124
Anexo 6. Código para identificar el posible ajuste de los pixeles de la estrella. ...	127
Anexo 7. Resultados distribución todo el sensor.	128
Anexo 7.1. Resultado <i>Chi</i> todo el sensor primera jornada.	128
Anexo 7.2. Número de clases de la prueba <i>Chi</i> todo el sensor primera jornada.	128
Anexo 7.3. Resultados K-S todo el sensor. Primera jornada	129
Anexo 7.4. Resultado <i>Chi</i> todo el sensor segunda jornada.	129
Anexo 7.5. Número de clases de la prueba <i>Chi</i> todo el sensor segunda jornada.	130
Anexo 7.6. Resultados K-S todo el sensor. Segunda jornada	130
Anexo 7.7. Resultado <i>Chi</i> todo el sensor tercera jornada.	131
Anexo 7.8. Número de clases de la prueba <i>Chi</i> todo el sensor tercera jornada.	131
Anexo 7.9. Resultados K-S todo el sensor. Tercera jornada	132
Anexo 8. Resultados distribución todo el sensor.	133
Anexo 8.1. Resultado <i>Chi</i> pixeles particulares primera jornada.	133
Anexo 8.2. Número de clases <i>Chi</i> pixeles particulares primera jornada.	133
Anexo 8.3. Resultado <i>K-S</i> pixeles particulares primera jornada.	133
Anexo 8.4. Resultado <i>Chi</i> pixeles particulares para la segunda jornada.	134
Anexo 8.5. Número de clases <i>Chi</i> pixeles particulares para la segunda jornada.	134
Anexo 8.6. Resultados <i>K-S</i> pixeles particulares para la segunda jornada.	135
Anexo 8.7. Resultado <i>Chi</i> pixeles particulares para la tercera jornada.	135
Anexo 8.8. Número de clases <i>Chi</i> pixeles particulares para la tercera jornada.	135
Anexo 8.9. Resultados <i>K-S</i> pixeles particulares para la tercera jornada.	136

Anexo 9. Coordenadas obtenidas mediante las metodologías de análisis de Clúster.....	137
Anexo 9.1. Coordenadas en declinación implementando el método del promedio ponderado.....	137
Anexo 9.2. Coordenadas en ascensión recta implementando el método del promedio ponderado.....	137
Anexo 9.3. Coordenadas en declinación implementando el método del centroide.....	137
Anexo 9.4. Coordenadas en ascensión recta implementando el método del centroide.....	138
Anexo 10. Distribución fondo de cielo.....	138
Anexo 10.1. Valores del estadístico <i>Chi</i> para el fondo de cielo.....	138
Anexo 10.2. Número de clases para los estadísticos <i>Chi</i> del fondo de cielo. ..	138
Anexo 10.3. Valores del estadístico <i>K-S</i> para el fondo de cielo.....	139
Anexo 11. Estadísticos calculados para la distribución de la estrella.....	139
Anexo 11.1. Valores del estadístico <i>Chi</i> para la distribución de la estrella.	139
Anexo 11.2. Valores del estadístico <i>K-S</i> para la distribución de la estrella.....	139

RESUMEN

En los observatorios astronómicos la captura de imágenes permite el registro de diferentes fenómenos proporcionando información de primera mano sobre la que los investigadores realizan diferentes tipos de análisis en diversas áreas como la astrometría, fotometría, espectroscopia, entre otras.

En el observatorio astronómico de la Universidad Tecnológica de Pereira (*OAUTP*) estas imágenes son adquiridas mediante cámaras con tecnología *CCD*¹, tecnología que presenta enormes ventajas como por ejemplo su gran sensibilidad. Sin embargo, el registro de electrones producto de la electrónica del instrumento es una, entre varias, de sus grandes desventajas ya que genera información que no es propia del objeto observado.

En el presente trabajo de investigación se propone, partir de la adquisición de imágenes mediante la cámara *SBIG ST 2000XM* en el *OAUTP*, identificar la función de distribución que mejor describe un ruido conocido como corriente de oscuridad generado por dicho instrumento. Esta señal afecta significativamente la presentación e información contenida en las capturas realizadas. Por tanto, conocer el ajuste de estos datos permitirá emplear metodologías para una óptima extracción de este registro.

Entre tanto, se presenta el inconveniente de identificar la pertenencia de un pixel a una estrella o al fondo de cielo, debido a que los pixeles que forman el astro poco a poco van haciéndose más tenues dificultando la identificación de sus límites. Por consiguiente, si se logra identificar la función de distribución de la estrella y del fondo de cielo que la rodea, se pueden determinar las funciones de densidad de probabilidad. De esta manera, se podría evaluar los valores de pixeles con comportamiento ambiguo en dichas funciones y al observar sus probabilidades dar una idea de la pertenencia a una u otra clase.

Por último, se plantean y analizan metodologías para medir las distancias entre conglomerados de pixeles que forman los cuerpos celestes en las fotografías. La determinación de estas magnitudes serán utilizadas para la determinar las coordenadas ecuatoriales de uno objeto cualquiera en la imagen.

¹ Dispositivo de carga acoplada por sus siglas en ingles (charge-coupled device), es un circuito integrado generalmente de silicio que genera una corriente eléctrica proporcional a la cantidad de fotones que inciden en él y mediante una transformación analógica digital esta señal es convertida en una imagen.

INTRODUCCIÓN

En las diferentes áreas de las ciencias experimentales el manejo de los datos (o de grandes cantidades de los mismos) es de gran importancia, puesto que son ellos los que permiten realizar inferencias o comprobar hipótesis acerca de los fenómenos bajo estudio.

Las imágenes como fuentes de información obtenidas con diversos instrumentos a partir del registro de las diferentes frecuencias del espectro electromagnético y principalmente los rayos X, las ondas de radio, la luz visible, infrarrojo etc. Tienen actualmente gran acogida en campos en los que se destacan: la medicina, la astronomía, obstetricia, inteligencia artificial, biología, ingeniería civil etc. Ya que permiten obtener información de sitios o lugares que el ojo humano no puede observar, bien sea porque existe un cuerpo material que se interpone, por la lejanía del objeto de estudio o porque simplemente el cuerpo emite en una frecuencia diferente a la del espectro visible.

En el caso de la astronomía, el avance vertiginoso de la instrumentación (telescopios ópticos, radiotelescopios, entre otros) ha puesto a disposición de los científicos grandes volúmenes de información que requiere ser analizada, y dado que recursos como el tiempo y la capacidad computacional son limitados (esta última debida a los requerimientos de hardware) ha llevado a la estadística a ocupar un papel determinante en las investigaciones astronómicas.

La astronomía óptica; por otro lado, estudia los objetos astronómicos en longitudes de onda visibles del espectro electromagnético mediante la adquisición de imágenes en observatorios astronómicos. El análisis de estas imágenes permite identificar algunas características de los cuerpos celestes; entre las que se destacan: el brillo, color, forma, paralaje, distancias entre cuerpos en la fotografía, temperatura, composición, etc.

Sin embargo, no todos los datos contenidos en las imágenes astronómicas provienen del objeto registrado ya que la información (la luz percibida) se ve influenciada por diferentes perturbaciones debidas a la atmosfera, la luna, la contaminación lumínica, al ruido instrumental generado por el sensor utilizado para la captura de la fotografía, entre otras. Debido a esto, la señal registrada en los pixeles no es exclusivamente del cuerpo de estudio, sino también de las perturbaciones antes mencionadas a las que se les conoce como ruido.

Ahora bien, los objetos en las imágenes astronómicas están formados por diferentes cantidades de pixeles, estos pixeles a medida que se alejan del “centro”

del objeto fotografiado disminuyen su intensidad hasta el punto de confundirse con el fondo de la imagen haciendo difícil discernir hasta donde llega el cuerpo de estudio y cuáles son los bordes o límites. Por tanto, clasificar que pixeles pueden considerarse como parte de un objeto o del fondo del cielo en la imagen adquiere relevancia en áreas como la fotometría.

Entre tanto, como los objetos fotografiados se componen de diferentes cantidades de pixeles surge el inconveniente de medir las distancias entre dos cuerpos y poder hallar con esto las coordenadas en ascensión recta y declinación de uno de ellos.

Para este último caso y basándose en metodologías del análisis de Clústers, se obtendrán las métricas entre dos o más estrellas de la misma fotografía, para de esta manera determinar la posición de una de ellas.

Con respecto al ruido, inherente a los sensores *CCD*, se intentará determinar si sigue cierta distribución de probabilidad conocida aplicando pruebas de bondad y ajuste, principalmente *Kolmogorov-Smirnov* y *Chi Cuadrado*, a una muestra aleatoria simple obtenida de las imágenes; así como también, a los valores que registra un mismo pixel en diferente tiempo de exposición.

Por último. Se plantea la idea de que si se determina la distribución del fondo del cielo así como la distribución de los pixeles que conforman la estrella, resultaría sencillo evaluar las cuentas de un pixel obteniendo así su probabilidad y poder realizar la clasificación de si pertenece al fondo de cielo o a la estrella siempre y cuando se puedan determinar sus respectivas distribuciones.

Estos aspectos a desarrollar intentan dar solución a ciertos inconvenientes que se presentan en el Grupo de Investigación en Astroingeniería Alfa Orión adscrito a la Facultad de Ciencias Básicas de la Universidad Tecnológica de Pereira.

1. PRELIMINARES

1.1 DESCRIPCIÓN DEL PROBLEMA

El observatorio astronómico de la Universidad Tecnológica de Pereira cuenta, entre muchos otros instrumentos, con una cámara SBIG ST-2000XM la cual es utilizada para la toma de fotografías de diferentes cuerpos celestes. Dicha cámara expone un sensor CCD a la luz, el cual emite electrones proporcionales a la cantidad de fotones que inciden en él. Estos electrones son almacenados en diferentes pozos llamados pixeles, pixeles que posteriormente mediante un conversor de unidades analógico digitales son cuantizados y codificados binariamente para luego ser mostrados como imágenes digitales. Ahora bien, debido a la electrónica de la cámara, son registradas ciertas cantidades de electrones que no son debidas a fotones del cuerpo astronómico sino por el contrario propias del funcionamiento del instrumento, dichos electrones reciben el nombre de corriente de oscuridad, esta señal hace parte de la fotografía y debido a su naturaleza aditiva interviene directamente en la cantidad de cuentas por pixel registradas en la imagen afectando significativamente estimaciones fotométricas y disminuyendo la calidad de la fotografía.

En la Figura 1 se muestra un fragmento de un dark frame, la diferencia en las tonalidades se deben a que los pixeles tienen un registro de cuentas diferentes variando de fotografía en fotografía incluso bajo condiciones similares de exposición y temperatura. Esto es debido a diferencias en la sensibilidad, a procesos de manufactura entre otras.

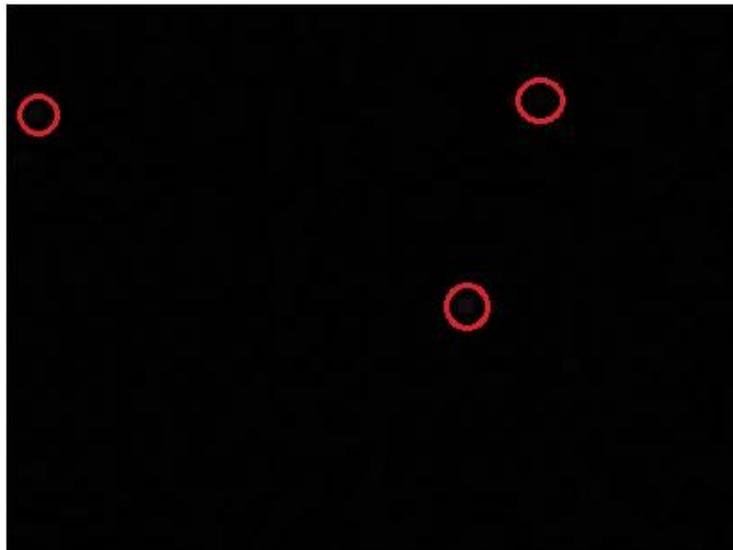


Figura 1. Fragmento de un Dark Frame.

Por otro lado, al determinar las distancias verticales y horizontales de varios elementos a uno en particular de la misma imagen y obtener con ello las coordenadas en ascensión recta y declinación de este último, se presentan dificultades al encontrar una magnitud para dicha medida entre los cuerpos y el error asociado, esto se debe principalmente a que cada uno de los objetos está formado por diferentes cantidades de píxeles lo que dificulta realizar la métrica. La Figura 2 ilustra la situación planteada.



Figura 2. Fotografía de una estrella adquirida desde OAUTP.

Por último, se busca determinar si un píxel hace parte o no de la estrella, pues como se aprecia en la imagen anterior los píxeles a medida que se alejan del “centro” del astro se van haciendo cada vez más tenues hasta confundirse con el fondo del cielo; surgiendo así el inconveniente de la mencionada clasificación.

1.2 JUSTIFICACIÓN

La corriente de oscuridad generada por los sensores *CCD* está implícita en las cuentas de cada pixel de una imagen digital debido a su naturaleza aditiva. Esta señal variable, que no proviene del objeto fotografiado, ha de ser extraída de cada fotografía en la cantidad óptima minimizando la pérdida o el exceso de información producto de separar dicha señal. Por tanto, la estimación estadística del comportamiento de esta perturbación adquiere relevancia ya que permitirá una sustracción adecuada de tal ruido mejorando la calidad de las imágenes y la precisión de la información allí contenida.

Entre tanto, el implementar metodologías para medir las distancias entre conglomerados de pixeles es de suma importancia ya que posibilitará determinar las coordenadas ecuatoriales desconocidas de un objeto en una fotografía a partir de otros de la misma. Primer objetivo y paso de un proceso que tiene como meta determinar orbitas de objetos tales como asteroides o cometas. Además el entregar una incertidumbre en dicha medida permitirá hallar el error sistémico en la determinación de tales orbitas.

Por último, el clasificar que pixel hace parte de la estrella y no del fondo de cielo en una imagen, mediante sus funciones de distribución, ofrecerá una mejor confianza en el valor de la sumatoria de las cuentas de todos los pixeles que forman la estrella y de esta manera no sumar valores que no corresponden; permitiendo con ello una mejor estimación de las magnitudes fotométricas.

Estas dificultades han de superarse exitosamente ya que son inconvenientes generados en investigaciones realizadas o que se están llevando a cabo por el Grupo de Astroingeniería Alfa Orión, a la Facultad de Ciencias Básicas de la Universidad Tecnológica de Pereira.

Superar cabalmente estos impases es fundamenta ya que garantiza que los pasos subsiguientes tengan datos sólidos sobre los que realizar inferencias o cálculos, pues realizar un análisis fotométrico defectuoso, una sustracción incorrecta de la corriente de oscuridad; así como también, un cálculo erróneo en la posición de un cuerpo se evidenciará en gran variedad de predicciones incorrectas, cálculo de parámetros erróneos y conclusiones no muy acertadas que lapidarán la investigación.

1.3 OBJETIVO GENERAL

Establecer sí el ruido aditivo y los valores registrados por un cuerpo celeste en las imágenes astronómicas capturadas desde el Observatorio Astronómico de la Universidad Tecnológica de Pereira se ajustan a una distribución de probabilidad.

1.4 OBJETIVOS ESPECÍFICOS

- Determinar mediante pruebas de bondad y ajuste sí el ruido aditivo presente en las imágenes astronómicas capturadas desde el observatorio de la Universidad Tecnológica de Pereira, se ajustan a una distribución y establecer cuál sería en tal caso.
- Determinar mediante pruebas de bondad y ajuste la distribución de probabilidad de los valores de píxel registrados al observar un cuerpo celeste en las fotografías.
- Determinar en caso de que exista, la función de distribución del fondo del cielo durante una jornada de observación y una región de cielo en particular.
- Realizar la medición de las distancias existentes entre los objetos presentes en la fotografías utilizando técnicas de análisis de Clústeres.
- Estimar el error en la distancia entre dos objetos celestes presentes en la misma fotografía.
- Obtener las coordenadas ecuatoriales de un objeto en particular a partir de otros.

1.5 ESTADO DEL ARTE

Desde la invención de los primeros dispositivos *CCD*, llevado a cabo por William Boyle y George Smith en 1969 en los laboratorios Bell, este dispositivo mostró gran aplicación en diversos campos debido a sus ventajas sobre la fotografía convencional como la digitalización de la imagen que permitió un fácil manejo de esta, su gran sensibilidad y resolución, mostrar la fotografía sin necesidad de revelado, etc.

Por otro lado existe un ruido asociado a cada imagen obtenida por un *CCD* debido principalmente a procesos de manufactura, eficiencia fotoeléctrica, corriente oscura, fluctuaciones aleatorias de la cantidad de fotones que llegan al sensor, el proceso de conversión de unidades analógico digitales, diferencias en la sensibilidad, inicio en el registro de las cuentas, entre otras, lo que ha conducido a la implementación de diversas metodologías para minimizar al máximo los efectos de esta señal. Estas metodologías son empleadas en diversos campos que van desde la visión máquina, biología, radiología hasta la astronomía.

Con el fin de medir y corregir el ruido de los sensores *CCD* se han desarrollado diversas investigaciones desde la invención del dispositivo, ejemplo de ello en 1974 M. Mohsen, Michael F. Tomsett y Carlo H Sequin [1] realizaron un montaje electrónico que mediante el uso de electrodos de policilicio trifásicos sobre el *CCD*, buscaba cuantificar diversos ruidos clasificados en intrínsecos y extrínsecos, siendo los primeros asociados a la operación física inherente a los dispositivos mientras los últimos se asociaron a la señal que ingresaba y salía del dispositivo. En aquel trabajo las conclusiones de las medidas obtenidas mostraron que para las fuentes intrínsecas el nivel del ruido concordaba con lo que se sabía del funcionamiento del dispositivo sin embargo el nivel de ruido de las fuentes extrínsecas resultó mucho mayor, que el esperado a partir de la teoría.

Por otro lado, modelos matemáticos de ruido, funciones y filtros también son desarrollados para disminuir el impacto de esta señal perniciosa en las imágenes digitales siendo así que en 1994 [2] se generaron modelos matemáticos tanto para la cámara *CCD* como materiales reflectantes, cuantificando las variaciones de los píxeles en cámara digitales que son debidos al sensor y la variación de la escena. Este análisis forma la base del algoritmo para caracterizar la cámara, calibrarla y realizar una descripción de la escena. Para ello se examinan las propiedades de los dispositivos de carga acoplada mediante el estudio del *CCD* y con ello identificar y cuantificar varias fuente de ruido, se genera un modelo de la cámara que tiene en cuenta diferentes tipos de señal tal como *dark current* y ruido de disparo. Luego se modela la reflectancia y la iluminación y se estima la varianza

del ruido debido a las fuentes intrínsecas del *CCD* del modelo propuesto de la cámara. Posteriormente se puede entonces remover el ruido mediante un algoritmo de restauración

Más adelante, en 2006 [3], se propone un esquema para remover el ruido y restaurar las imágenes generadas por el *CCD*, el modelo de ruido propuesto es generado como la suma de una señal dependiente y otra señal independiente en términos de ruido. En aquella investigación se desarrollaron dos técnicas adaptativas de restauración que tienen en cuenta la no linealidad del sensor. Una opera en el espacio de luz, donde los valores de la luz incidente y los valores del espacio de luz son lineales (la luz que incide y la que se registra o electrones emitidos por el sensor), mientras la segunda utiliza el modelo del ruido transformado para operar en el espacio de la imagen. Ambas técnicas utilizan filtros adaptativos y unen sus respuestas para dar una imagen restaurada. Para tal fin, y para poder distinguir entre el espacio de la luz y el espacio de la imagen, se crea una *CRF* (función de respuesta de la cámara) la cual relaciona la cantidad de fotones incidentes y la salida de la cámara. Esta relación es no lineal debido al comportamiento del sensor y rutinas en él. Los valores de salida del sensor antes de ser digitalizados es lo que se denomina espacio de luz, mientras que el valor final del pixel es lo que los autores denominan espacio de la imagen. Luego de generar la función *CRF* se desarrollaron los métodos de restauración basados en el modelo del ruido para el espacio de luz y la variación no transformada de este modelo en el espacio de imagen. Los resultados obtenidos sugirieron que en el espacio de luz es más eficiente pues permite diseñar un filtro de aplicación más simple.

Siguiendo esta misma línea de diseños de filtros en 2007 Wu Jingjing, Zhang Zhenchuan y Tian Yong [4] diseñaron un filtro adaptativo para remover el ruido de las imágenes digitales generadas por un *CCD*, para ello un modelo de ruido es propuesto como una mezcla de patrones fijos, ruido multiplicativo gaussiano y una señal independiente de ruido, bajo estas condiciones se construye el modelo matemático que servirá de soporte para el diseño de un algoritmo que restaurará las imágenes del ruido. El primer paso consistió en el cálculo de los parámetros del modelo a través del método “fuera de línea” que permiten formular la función FANS, filtro adaptativo de ruido suave por sus siglas en inglés (Adaptative Noise Smoothing Filter), este es el filtro que se utilizó para remover el ruido luego de realizar una descomposición secuencial de Wavelet sobre la imagen, que permiten una fuerte correlación en todas las dimensiones. Por último para definir el algoritmo se usaron cantidades estadísticas simples como la varianza y el promedio en la imagen transformada. Los resultados obtenidos mediante este

método en imágenes con un ruido simulado fueron superiores comparados con otras formas de filtros.

Poco tiempo después en 2008 Ce Liu, Richard Szeliski, Sing Bing Kang y William T. Freeman [5] y advirtiendo que en la mayoría de los trabajos de eliminación del ruido que existían asumían un ruido aditivo gaussiano blanco (*AWGN*) siendo este removido independientemente de los datos *RGB* de la imagen sin tener en cuenta que el tipo y el nivel de ruido generado por cámaras digitales son desconocidos si la serie y la marca de la cámara así como sus características (Iso, Shutter Speed, Apertura...) son también desconocidas. Los autores proponen pues un marco unificado para dos tareas: estimar y remover el ruido automáticamente mediante la implementación de un algoritmo usando fragmentos suaves de la imagen e introduce la función de nivel de ruido (*NLF*), una función que describe el nivel del ruido en función del brillo de la imagen. Con el fin de sustraer el ruido, se propone estimar un límite superior de la función nivel de ruido (*NLF*) obtenido de las desviaciones estándar de las variaciones de los segmentos tomados de la imagen, para ello la imagen se divide en regiones suaves a trozos, esta segmentación se realizó mediante el método K-means agrupando píxeles en regiones en que la media se estima del brillo y la desviación estándar es una sobre estimación del ruido. La probabilidad de la función (*NLF*) es obtenida mediante la simulación del proceso digital de la cámara y es usada para estimar correctamente la curva donde hay datos faltantes. Los resultados arrojados por el algoritmo mostraron efectividad en la remoción del ruido.

Entretanto formas más “directas” para remover el ruido también han sido propuestas siendo así como en [6] se propone un método para estimar y remover la corriente oscura, el método iterativo construye una aproximación sucesiva de esta señal instrumental de la imagen. El algoritmo para realizar la remoción fue probado en imágenes adquiridas en THERMO-WET, sistema computarizado para medir las propiedades de la superficie de un material a altas temperaturas. El proceso inicia colocando la muestra del material investigado dentro del horno, ello se hace mediante un mecanismo de inserción, la muestra es calentada a una temperatura mucho más alta que el punto de fusión, cuando se convierte en gota la unidad de visión del THERMO-WET adquiere las imágenes de la muestra y procede a una segmentación de la imagen, en el siguiente paso el proceso de medición de los parámetros geométricos de la gota son determinados, es de señalar que la calidad de la imagen es crucial para realizar el proceso de medición. Experimentos realizados mostraron que los niveles de gris en los dark frames pueden ser considerados constantes para una cámara dada. Esto significa que una vez construido el master dark frame se puede sustraer de la imagen. El dark frame es construido de la siguiente manera: se construye una matriz de ceros

con las mismas dimensiones de los dark frames, se leen los dark frames, se promedian los dark frames, el promedio se le suma a la matriz ceros, luego se le resta el promedio a cada uno de los dark frames. Este ciclo se repite hasta que no exista distorsión en los dark frames. Obtenido el master dark frame se calcula el bias² para restar la corriente de oscuridad de las imágenes.

En astronomía la eliminación del ruido también es tratada pues permite mejores mediciones fotométricas y en general métricas sobre la imagen como el cálculo de centroides. No se han encontrado trabajos en los que el ruido en imágenes astronómicas sea tratado mediante algún tipo de filtro o modelo matemático siendo este estimado mediante la construcción de diferentes frames.

Y es así como en [7] se plantea un nuevo enfoque para mejorar la precisión del centroide dado que la presencia de varias fuentes de ruido durante la exposición de un *CCD* puede dar origen a una significativa degradación en la precisión de este. El enfoque presentado aquí utiliza un modelo Spot³ para derivar el límite de la señal que es usado para eliminar el ruido. Este proceso reduce efectivamente el ruido aleatorio y el bias. Con el fin de reducir al máximo el ruido se hicieron sugerencias para usar únicamente 9 píxeles alrededor del pico de la señal. Esta suposición facilita y mejora la velocidad con la que se calcula el centroide sin afectar la precisión de este únicamente si la señal es limitada a esta pequeña región local. Sin embargo se notó que esta suposición afecta la precisión del cálculo, por tanto el número de píxeles tiene que ser cuidadosamente seleccionado para no sacrificar precisión. En el artículo se propone usar un método Spot para determinar que píxeles son utilizados para la estimación del centroide. Es claro que el bias cerca de los bordes de la ventana del centroide aumenta el ruido ya que las coordenadas del objeto aumentan hacia allí también. Debido a que el error aleatorio del centroide (*NEA*, noise equivalent angle) es inversamente proporcional a la *SNR* (señal noise ratio) lo que se busca es incrementar la señal o disminuir el ruido con el fin de disminuir el *NEA*. Para ello se construyen modelos Spot y cada modelo es utilizado para determinar la cantidad de píxeles que están corrompidos por ruido de manera significativa, un centroide es estimado utilizando un algoritmo estándar. Este centroide es usado para determinar qué modelo Spot debe ser utilizado. Luego de esto se estima el nivel de ruido presente en la ventana. Este se estima mediante un promedio y por último un algoritmo de centroide estándar es aplicado.

² Es un ruido generado por el convertidor analógico digital de la cámara y causa que los píxeles no inicien el registro de las cunetas en cero sino en valores superiores no siendo uniformes para todo el sensor.

³ Los modelos *Spot* buscan identificar la respuesta de los píxeles del sensor a una fuente puntual mediante la función de dispersión de punto (*PSF*).

Por otro lado en [8] cuyo principal objetivo es, a partir de la observación de M 13, detectar estrellas RR Lyrae en los cúmulos globulares con telescopios presentes en casas de grandes aficionados o en pequeños observatorios permitiendo el seguimiento durante largos periodos de tiempo, las observaciones fueron realizadas en el observatorio astronómico de Cantabria, España. Que cuenta con un telescopio Meade de 16", un sensor CCD ST-8VE entre otros. Cada noche de observación se realizaron imágenes de calibración. Flats al atardecer y al amanecer, bias y darks al comienzo, medio y final de la noche. El proceso de calibración se realizó promediando los dark, flats y bias. Luego se restó el promedio tanto de dark y bias a la imagen para dividirla más tarde por el flat.

En [9] se tiene como objetivo la operación, calibración y análisis de datos ópticos del instrumento *OMC* (Optical Motoring Camera) a bordo del *INTEGRAL* (International Gamma-ray Astrophysics Laboratory) cuyos principales objetivos fueron: definir un algoritmo de centrado que permita determinar en tiempo real el apuntado de la *OMC* con una precisión comparable al tamaño de su pixel, definir una estrategia de calibración tanto desde el punto de vista de operaciones como de procesamiento de datos, desarrollo y mantenimiento de los paquetes de software necesario para realizar la calibración del instrumento (corriente de oscuridad y polarización, aplanamiento del campo, etc). Varios son los parámetros que permiten la configuración del algoritmo de calibración: las coordenadas de las secciones que se deben transmitir, el número de tomas a realizar, tiempo de integración, factores de binning entre otros. Para determinar la corriente de polarización la cámara permite el registro de lectura del detector repetidamente, de esta forma se obtiene una imagen que incluye únicamente la corriente de polarización y el ruido de lectura. Para la calibración del campo plano, se obtiene una secuencia de imágenes con cada uno de los leds encendidos, leds a bordo de la *OMC*, y con ambos apagados. A todas las imágenes obtenidas se les resta la corriente de polarización, se combinan las imágenes con los leds encendidos y si se dispone de imágenes con el led apagado (fondo del cielo) son también combinadas entre sí, para restar el resultado a la imagen previamente combinada del led encendido. Posteriormente se deriva un patrón a gran escala ajustando una superficie a la imagen resultante mediante el uso de una interpolación de Splines en dos dimensiones (las secciones corregidas durante este proceso son normalizadas por su moda y unidas para crear una imagen completa del detector, esta imagen es utilizada para determinar los pixeles malos, calientes o fríos. En este caso los pixeles que se alejan por encima o por debajo de la media un 20%). Finalmente se divide la imagen combinada, con el cielo ya restado, por el patrón de iluminación para obtener la imagen corregida.

El conocer las distribuciones que siguen los diferentes tipos de ruido es de gran importancia pues permitirá establecer cuál es la mejor metodología para tratar esta señal. Siendo entonces que en 2013 [10] se implementa y desarrolla un algoritmo para la estimación automática de la varianza y discriminación de ruido presente en imágenes digitales, la varianza del ruido se estima basado en el cálculo de la moda obtenida de las varianzas a partir de ventanas locales (segmentos de la imagen) además se transforma una distribución de ruido específica en una distribución Gaussiana con varianza unitaria utilizando transformaciones de estabilización de la varianza, el algoritmo implementado en Matlab estima la varianza y discrimina entre las siguientes distribuciones de ruido: Gussiano, Poisson, Gamma y Rayleigt.

2. MARCO TEÓRICO

En las páginas siguientes referentes al marco teórico se explicarán los conceptos que soportan el desarrollo metodológico.

Con el fin de identificar la distribución de probabilidad seguida por los diferentes valores que toman los píxeles en las fotografías de las estrellas, de la corriente de oscuridad y del fondo del cielo; se definirán conceptos como variable aleatoria, seguido por las distribuciones que serán contrastadas en las hipótesis nulas de las pruebas de bondad y ajuste. Posteriormente, se ilustrarán los test *Kolmogorov-Smirnov* y *Chi cuadrado*. Para luego pasar a las metodologías del análisis de *Clústers*; Además de lo anterior, se explica el tipo de muestreo que se aplicó a las fotografías. Por último, la idea de imagen digital es definida.

2.1 VARIABLE ALEATORIA

Es una función que asigna un valor numérico a cada uno de los posibles eventos de un espacio muestral, los valores de dicha variable que constituyen un conjunto finito de números o que se pueden listar en una secuencia infinita de estos recibe el nombre de variable aleatoria discreta [11].

2.2 FUNCIÓN DE DISTRIBUCIÓN PARA UNA VARIABLE ALEATORIA

Dada una variable aleatoria X , la función de distribución se define como [12]:

$$F(x) = P(X \leq x) \quad \forall x \in \mathbb{R} \quad (1)$$

Es decir, la función de distribución asigna la probabilidad que la variable aleatoria tome valores menores o iguales x .

Propiedades:

- $0 \leq F(x) \leq 1 \quad \forall x \in \mathbb{R}$
- $\forall a \leq b \quad P(a \leq \xi \leq b) = F(b) - F(a)$

2.3 FUNCIÓN DE DENSIDAD PARA UNA VARIABLE ALEATORIA

Sea X una variable discreta que puede tomar los valores x_1, x_2, \dots, x_n ; se denomina función de densidad $f(x)$ a la función que asigna a cada valor de la variable, la probabilidad de que ocurra [12].

$$f(x_i) = P(X = x_i) \quad (2)$$

Propiedades:

- $0 \leq f(x_i) \leq 1 \quad \forall x_i$
- $\sum_{i=1}^n f(x_i) = 1$

La relación existente entre la función de densidad y la función de distribución se puede expresar de la siguiente manera para variables discretas.

Dada $f(x_i)$ se puede obtener $F(x)$ como:

$$F(x_j) = \sum_{i=1}^j f(x_i) \quad (3)$$

De la misma manera también se puede obtener $f(x_j)$ mediante:

$$f(x_j) = F(x_j) - F(x_{j-1}) \quad (4)$$

2.4 DISTRIBUCIÓN DE POISSON

Una variable aleatoria discreta X sigue una distribución de Poisson de parámetro $\lambda > 0$ y se denota como $X \sim P(\lambda)$, si su función de densidad, la cual es mostrada en la Figura 3 para valores diferentes λ , viene dada por [12]:

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda} \quad k \in \{0, 1, 2, \dots\} \quad (5)$$

2.5 DISTRIBUCIÓN NORMAL

Sea x una variable aleatoria, se dice que sigue una distribución normal denotada como $x \sim N(\mu, \sigma)$; como bien puede apreciarse en la Figura 4, donde μ y σ representan la media y la desviación respectivamente, si su función de densidad viene dada por [13].

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (6)$$

Y su función de distribución se puede expresar como:

$$F(x) = P(X \leq x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-(v-\mu)^2/2\sigma^2} dv \quad (7)$$

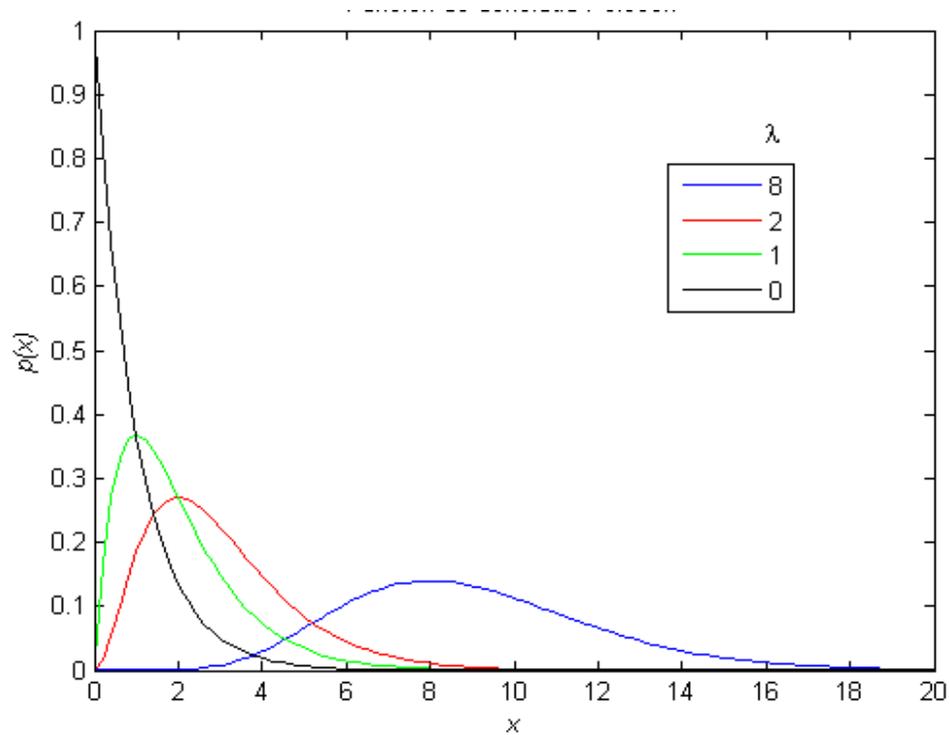


Figura 3. Función de densidad de Poisson.

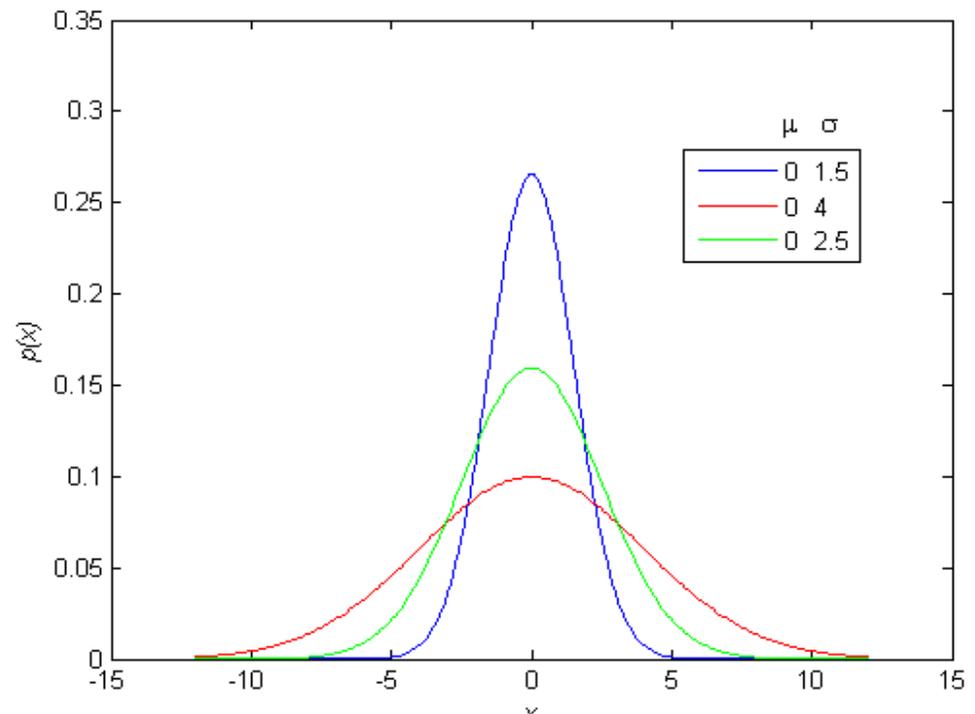


Figura 4. Función de Densidad Normal.

2.6 DISTRIBUCIÓN WEIBULL

Una variable aleatoria X sigue una distribución Weibull de parámetros α y β denotada $X \sim Wiubull(\alpha, \beta)$ si su función de densidad está definida por:

$$f(x) = \alpha\beta(\alpha x)^{\beta-1}e^{-(\alpha x)^\beta} \quad (8)$$

Donde α y β son mayores a cero, la Figura 5 muestra dicha función para valores diferentes de sus parámetros, y representan la escala y forma de la distribución respectivamente. Su función de distribución está definida como [14]:

$$f(x) = 1 - e^{-(\alpha x)^\beta} \quad (9)$$

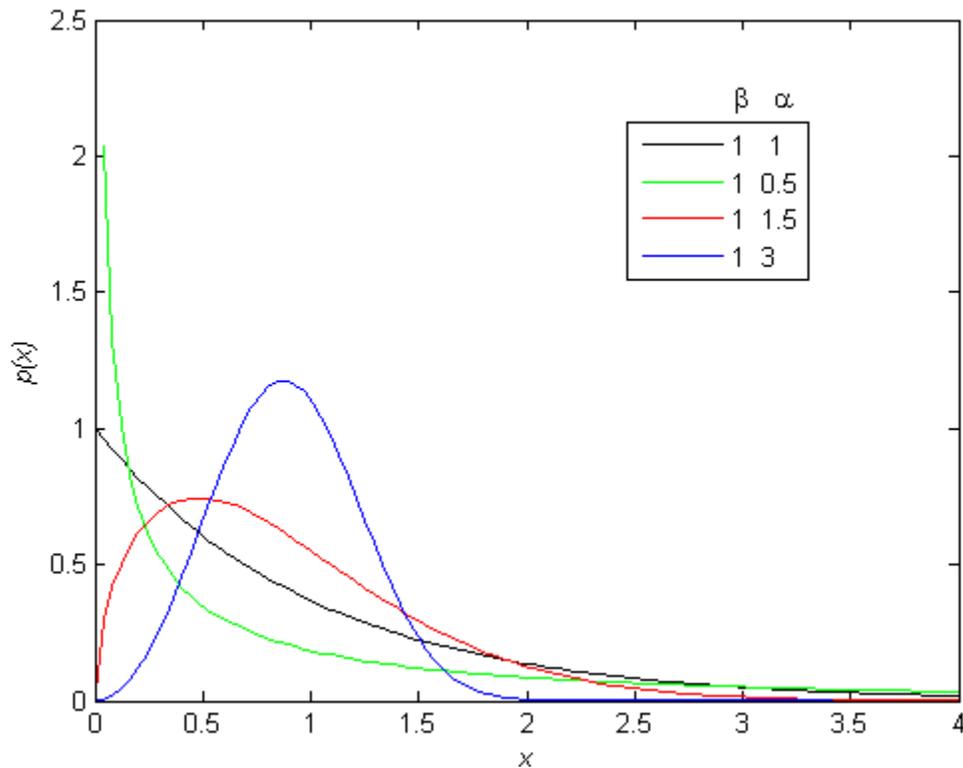


Figura 5. Función de Densidad Weibull.

2.7 DISTRIBUCIÓN GAMMA

Una variable aleatoria X sigue una distribución gamma, tal y como lo muestra la Figura 6, si su función de densidad viene dada por [12]:

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-x\lambda} \quad (10)$$

Donde

$\lambda \geq 0$, $\alpha \geq 0$ y $\Gamma(\lambda)$ es la función gamma definida por:

$$\Gamma(\lambda) = \int_0^{\infty} t^{\lambda-1} e^{-t} dt \quad (11)$$

α Parámetro de forma.

λ Parámetro de escala.

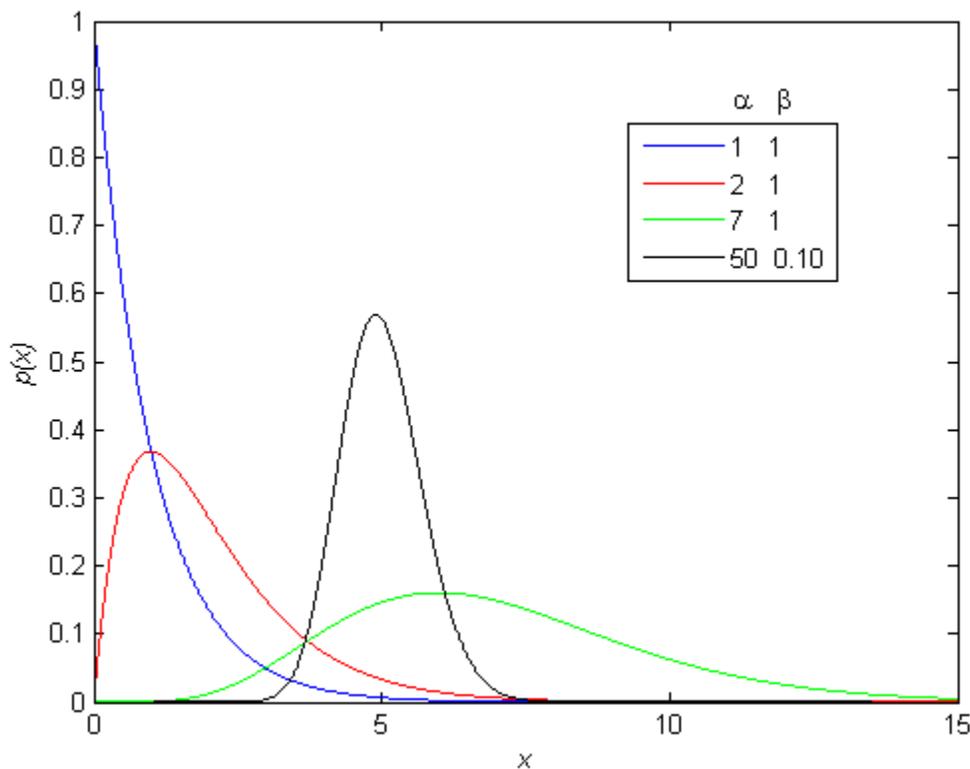


Figura 6. Función de Densidad Gamma.

2.8 DISTRIBUCIÓN LOG-NORMAL

Sea x una variable aleatoria y si se define $v = \ln(x)$ y se demuestra que v sigue una distribución normal con media μ y desviación σ entonces se dice que x sigue una distribución lognormal con parámetros μ, σ denotándose como $x \sim \text{LogN}(\mu, \sigma)$, tal y como se ilustra en la Figura 7, si su función de densidad viene dada por [15]:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}} \quad (12)$$

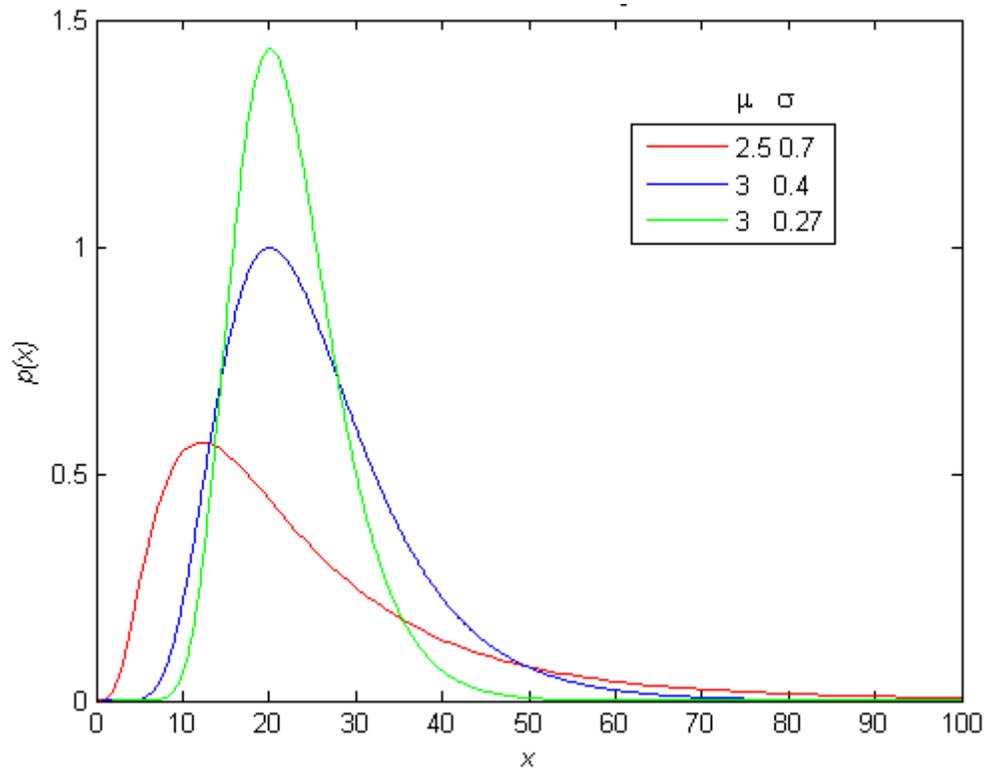


Figura 7. Función de Densidad Lognormal.

2.9 PRUEBAS DE BONDAD Y AJUSTE

Permiten, mediante una prueba de hipótesis, establecer si un conjunto de observaciones tienen un comportamiento similar a una función de distribución teórica que será el contraste a realizar.

2.10 PRUEBA KOLMOGOROV-SMIRNOV

Esta prueba no paramétrica⁴ permite establecer la distribución que pueden llegar a seguir una muestra de datos x_1, x_2, \dots, x_n comparando las frecuencias acumuladas empíricas con la distribución teórica propuesta en la hipótesis nula [16], hipótesis que se define como:

$$H_0: F_n(x) = F_0(x) \quad (13)$$

Donde:

$F_n(x)$ Representa la función de distribución empírica.

⁴ Este tipo de estadística se caracteriza porque no supone ningún parámetro ni distribución para la variable aleatoria; a diferencia de la paramétrica que si lo hace.

$F_0(x)$ Representa la función de distribución teórica propuesta en la hipótesis nula.

El estadístico D que permite aceptar o rechazar la hipótesis nula se define como:

$$D = \text{Max}|F_n(x) - F_0(x)| \quad 0 \leq x \leq 1 \quad (14)$$

Y puede ser estimado mediante:

$$D = \text{Max} \left\{ \left| F_0(x_i) - \frac{i}{n} \right|, \left| F_0(x_i) - \frac{(i-1)}{n} \right| \right\} \quad i = 1, 2, 3, \dots, n \quad (15)$$

La Figura 8 muestra lo que se pretende con dicho estadístico y es hallar la mayor diferencia entre la función teórica propuesta en la hipótesis nula y la función teórica obtenida de los datos.

Este valor D es comparado con el estadístico Kolmogorov-Smirnov de acuerdo a la cantidad de datos y al nivel de significancia deseado, una tabla para la distribución de dicho estadístico se puede encontrar en [17] página 311, si D es mayor que dicho valor se rechaza H_0 mas si es menor se acepta H_0 .

$$D < D_{K-S} \quad (16)$$

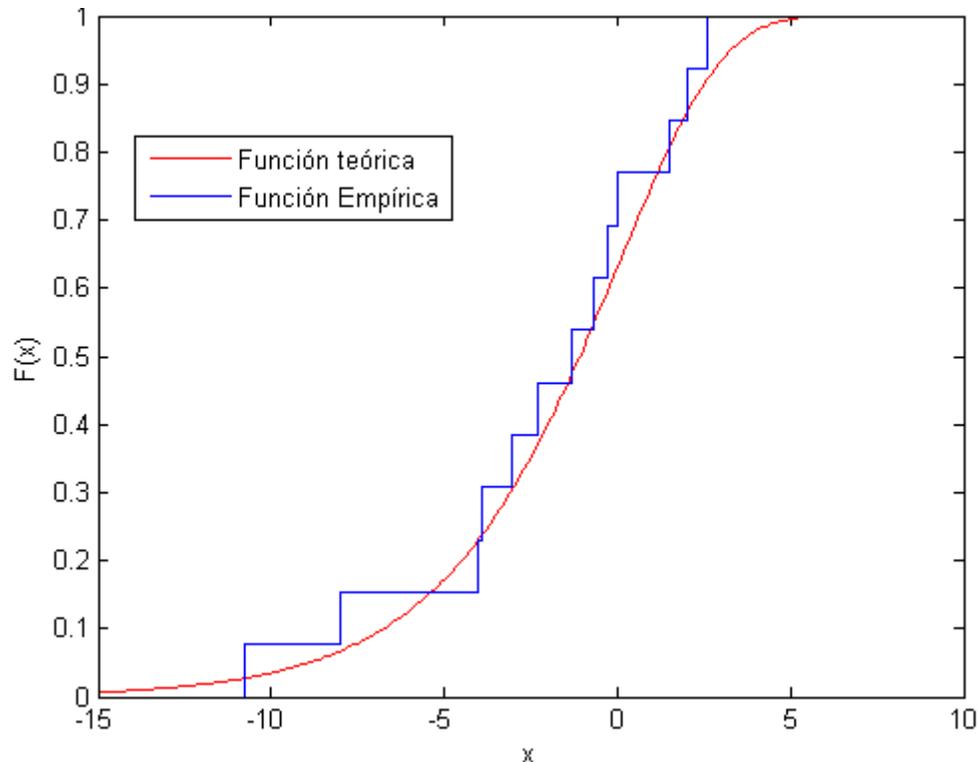


Figura 8. Comparación Función teórica y Empírica para el test KS.

2.11 PRUEBA CHI-CUADRADO

Si una muestra de una variable aleatoria x_1, x_2, \dots, x_n es agrupada en k clases, el estadístico χ^2 es obtenido mediante:

$$\chi^2 = \sum_{i=1}^k \frac{(Fo_i - Fe_i)^2}{Fe_i} \quad (17)$$

Donde:

Fo_i Representa la frecuencia observada en la i – esima clase.

Fe_i Representa la frecuencia esperada en la i – esima clase.

Si las frecuencias esperadas son al menos iguales a cinco la distribución muestral de χ^2 es aproximadamente Chi cuadrado [12].

Y en este trabajo en particular servirá para aceptar o rechazar la hipótesis nula.

$$H_0: F_n(x) = F_0(x) \quad (18)$$

Es decir los datos empíricos siguen cierta distribución teórica.

Si el valor χ^2 de la ecuación (17) es menor que el valor χ^2_{cr} crítico, la hipótesis nula es aceptada al nivel de significancia y grados de libertad dados, de lo contrario se infiere que hay evidencia para rechazarla. Una tabla para la distribución de este estadístico se encuentra en [17] página 382.

Los grados de libertad son hallados mediante $v = k - 1 - m$, donde k y m representan el número de clases y la cantidad de parámetros estimados de la muestra respectivamente.

La frecuencia esperada en cada clase debe ser mayor o igual a cinco de lo contrario se pueden incurrir en errores de estimación. Por otro lado si se aplican distribuciones continuas a datos discretos se ha de aplicar la corrección de Yates dada por:

$$\chi^2 = \sum_{i=1}^k \frac{(|Fo_i - Fe_i| - 0.5)^2}{Fe_i} \quad (19)$$

2.12 ANÁLISIS DE CLÚSTER

El análisis de Clúster permite agrupar datos que presentan algún grado de semejanza en clases o categorías y con ello descubrir asociaciones que no son evidentes a simple vista.

En este trabajo de grado los Clúster están ya formados siendo estos los pixeles que conforman cada una de las estrellas. Parte de la metodología que se emplea en el análisis de Clúster es aquella que tiene que ver principalmente con las distancias entre estos pues es esta la que permite los agrupamientos. Estas formas de medir las distancias son las que se estudiarán para los fines que en últimas conciernen esta investigación.

Existen varios métodos para obtener estas distancias en el análisis de Clúster, los escogidos fueron: promedio entre grupos y el centroide.

2.13 PROMEDIO ENTRE GRUPOS

Se define como el promedio de todas las posibles distancias resultado de las combinaciones que puedan existir entre dos conjuntos con cantidades diferentes de elementos [17] y se determina mediante:

$$D_{(u,v)} = \frac{\sum_{i,k} d_{i,k}}{n_u n_v} \quad (20)$$

Donde

$d_{i,k}$ Distancia del objeto i del grupo u al objeto k del grupo v .

n_u Número de elementos del grupo u .

n_v Número de elementos del grupo v .

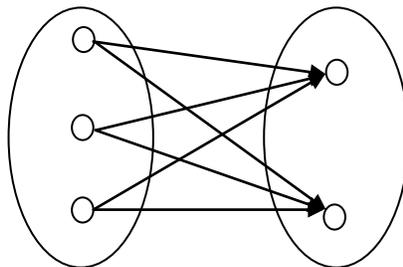


Figura 9. Metodología del Promedio Entre Grupos.

La Figura 9 permite una mejor comprensión de dicha metodología, en esta se observa que las distancias son medidas de cada uno de los elementos de un conglomerado a cada uno de los elementos de otro conglomerado para luego ser promediados.

2.14 MÉTODO DEL CENTROIDE

La distancia entre dos grupos se hace igual a la distancia euclídea entre sus centros, aquellos centros corresponden a los vectores de medias de los grupos [18].

$$d(A, B) = d(\bar{x}_A, \bar{x}_B) \quad (21)$$

Donde

$d(A, B)$ Distancia entre el grupo A y B .

$d(\bar{x}_A, \bar{x}_B)$ Distancias entre los vectores de las medias de los grupos.

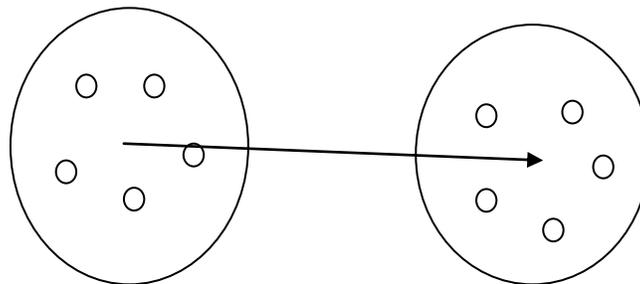


Figura 10. Metodología del Centroide.

La Figura 10 muestra lo anteriormente señalado, el método del centroide busca un centro común a cada uno de los elementos de ambos grupos a partir del cual se realiza la métrica.

Ahora bien, el método del promedio y del centroide pueden ser generalizados de manera que en ambos se calculen las $n - \text{simas}$ distancias con respecto a un primer clúster C_1 , seleccionado a priori, con respecto a un $C_n - \text{enesimo}$ cluster cualquiera. De la siguiente manera.

Sea $I(x, y)$ una imagen digital con n estrellas y cada una de ellas con m cantidad de pixeles, cada uno de ellos con un único par de coordenadas (x, y) , asumiendo cada una de estas estrellas como Clúster C_n a los que vienen asociados un par de coordenadas en ascensión recata y declinación. Podemos hallar la distancia de los C_n Clúster a C_1 , que representa el objeto cuyas coordenadas desconocemos.

2.15 MÉTODO DEL PROMEDIO GENERALIZADO

La distancia promedio en x y en y de cada C_n a un objeto C_1 viene dada por:

$$d(x)_k = \frac{1}{m_1 m_n} \sum_{j=1}^{m_n} \sum_{i=1}^{m_1} d_x(C_{1i}, C_{nj}) \quad k = 1, 2, \dots, n - 1 \quad (22)$$

Donde:

$d(x)_k$ Representa k -ésima distancia ponderada en x del primer al n -ésimo Clúster.

n n -ésimo Clúster, $n = 2, 3, \dots$

m_n Cantidad de pixeles del n -ésimo Clúster.

m_1 Cantidad de pixeles en el primer Clúster.

i Representa el i -ésimo pixel de primer Clúster $1 \leq i \leq m_1$.

j Representa el j -ésimo pixel del n -ésimo Clúster $1 \leq j \leq m_n$.

$d_x(C_{1i}, C_{nj})$ Es la distancia en el eje x del i -ésimo pixel del primer Clúster al j -ésimo elemento del Clúster n -ésimo dada por:

$$d_x = |C_{1i_x} - C_{nj_x}| \quad (23)$$

C_{1i_x} Representa la coordenada en x del i -ésimo pixel de primer Clúster.

C_{nj_x} Representa la coordenada en x del j -ésimo pixel de n -ésimo Clúster.

Análogamente la distancia en y puede ser determinada cómo:

$$d(y)_k = \frac{1}{m_1 m_n} \sum_{j=1}^{m_n} \sum_{i=1}^{m_1} d_y(C_{1i}, C_{nj}) \quad k = 1, 2, \dots, n - 1 \quad (24)$$

Donde:

$d(y)_k$ Representa k-ésima distancia ponderada en y .

n Enésimo cluster $n = 2, 3, \dots$

m_n Cantidad de pixeles del enésimo Clúster.

m_1 Cantidad de pixeles en el primer Clúster.

i Representa el i-ésimo pixel de primer Clúster $1 \leq i \leq m_i$.

j Representa el j-ésimo pixel del enésimo Clúster $1 \leq j \leq m_n$.

$d_y(C_{1i}, C_{nj})$ Es la distancia en el eje y del i-ésimo pixel del primer Clúster al i-ésimo elemento del Clúster enésimo dada por:

$$d_y = |C_{1i_y} - C_{nj_y}| \quad (25)$$

C_{1i_y} Representa la coordenada en y del i-ésimo pixel de primer Clúster.

C_{nj_y} Representa la coordenada en y del j-ésimo pixel de enésimo Clúster.

La incertidumbre en la medición para los Clúster se expresa como la dispersión de las medidas realizadas para cada el par de Clúster C_{1i}, C_{nj} .

Para y se tiene:

$$d_{Yk} \{ |C_{1i_y} - C_{nj_y}| \} \quad k = 1, 2, \dots, n - 1 \quad (26)$$

Se determina su desviación como:

$$\sigma_{kY} = \sqrt{\frac{1}{w - 1} \sum_{t=1}^w (d_{Yt} - \mu_{dY})} \quad (27)$$

Donde:

d_{Yk} Conjunto k-ésimo de medidas realizadas para el par de Clúster C_{1i}, C_{nj} .

C_{1i_y} Representa la coordenada en y del i-ésimo pixel de primer Clúster.

C_{nj_y} Representa la coordenada en y del j-ésimo pixel de enésimo Clúster.

σ_{kY} Desviación estándar del conjunto k-ésimo.

w Representa la cantidad de elementos del conjunto d_{Yk} .

d_{Yt} Representa el elemento t-ésimo del conjunto d_Y .

μ_{dY} Corresponde al valor de la media del conjunto d_Y .

De la misma manera tenemos que para x :

$$d_{Xk}\{[C_{1i_x} - C_{nj_x}]\} \quad (28)$$

$$\sigma_{d_X} = \sqrt{\frac{1}{w-1} \sum_{t=1}^w (d_{Xt} - \mu_{d_X})} \quad (29)$$

Donde:

d_{Xk} Conjunto k-ésimo de medidas realizadas para el par de Clúster C_{1i}, C_{nj} .

C_{1i_x} Representa la coordenada en x del i-ésimo pixel de primer Clúster.

C_{nj_x} Representa la coordenada en x del j-ésimo pixel del enésimo Clúster.

w Representa la cantidad de elementos del conjunto d_{Xk} .

d_{Xt} Representa el elemento t-ésimo del conjunto d_X .

μ_{d_X} Corresponde al valor de la media del conjunto d_X .

2.16 MÉTODO DEL CENTROIDE GENERALIZADO

En este método se calculan las distancias entre los centroides de cada uno de los Clúster. El centroide del enésimo Clúster c_n viene dado por el vector:

$$c_n = (c_{n_x}, c_{n_y}) \quad (30)$$

Cuyas componentes están definidas como:

$$c_{n_x} = \frac{1}{m_n} \sum_i^{m_n} C_{ni_x} \quad (31)$$

$$c_{n_y} = \frac{1}{m_n} \sum_i^{m_n} C_{ni_y} \quad (32)$$

Donde:

n Numero de Clusters $n = 1,2,3, \dots$

m_n Cantidad de pixeles del enésimo Clúster.

C_{ni_x} Coordenada en x del i -ésimo pixel del enésimo Clúster.

C_{ni_y} Coordenada en y del i -ésimo pixel del enésimo Clúster.

Con los centroides ya calculados se determina entonces la distancia del enésimo Clúster al C_1 .

$$d_n = |C_1 - C_n| \quad n = 2,3, \dots, n - 1 \quad (33)$$

La dispersión de las coordenadas del centroide se calcula como las desviaciones estándar de cada componente de los pixeles que conforman el Clúster. Así para x se tiene.

$$\sigma_{c_x} = \sqrt{\frac{1}{w-1} \sum_{t=1}^w (c_{Xt} - \mu_{c_x})^2} \quad (34)$$

Donde:

σ_{c_x} La desviación estándar del Clúster en x .

w Representa la cantidad de pixeles del Clúster c .

c_{Xt} Representa la coordenada en x del pixel t -ésimo del Clúster c .

μ_{c_x} Corresponde al valor de la media del Clúster c en x .

Para y es similar.

$$\sigma_{cY} = \sqrt{\frac{1}{w-1} \sum_{t=1}^w (c_{Yt} - \mu_{cY})^2} \quad (35)$$

Donde:

σ_{cY} La desviación estándar del Clúster en y .

w Representa la cantidad de pixeles del Clúster c .

c_{Yt} Representa la coordenada en y del pixel t -ésimo del Clúster c .

μ_{cY} Corresponde al valor de la media del Clúster c en y .

El objetivo del muestreo estadístico es realizar inferencias de la población a partir de la muestra seleccionada. Para ello una de las principales decisiones a tomar corresponde al tamaño de la muestra pues si esta es muy grande se desperdiciará gran cantidad de recursos, por otro lado si es muy pequeña arrojará estimadores imprecisos.

Para el cálculo del tamaño de la muestra se deben de tener en cuenta:

El error de muestreo: es la diferencia que puede existir entre el valor poblacional y la estimación que se haga a partir de la muestra. Este se puede expresar como un porcentaje no mayor del 10% aplicado a la media aritmética a partir de un estudio previo.

La varianza: con esta se busca que los valores que pueda tomar la variable queden representados.

La confianza: esta es asignada por el investigador de acuerdo al grado de precisión que desee.

Tamaño de la población: este es utilizado cuando se trata de poblaciones finitas.

2.17 MUESTREO ALEATORIO SIMPLE

En el muestreo aleatorio simple todos los elementos que conforman la población tienen la misma posibilidad de ser seleccionados [19] y el tamaño de muestra viene dado por:

$$n = \frac{n_0}{1 + \frac{n_0}{N}} \quad (36)$$

Donde

$$n_0 = \frac{Z^2 S^2}{E^2} \quad (37)$$

A menudo se considera que la varianza obtenida mediante un estudio preliminar no es representativa de la población y por tanto se realiza la siguiente corrección.

$$n_0 = \frac{Z^2 S^2}{E^2} \left[1 + \frac{2}{n_1} \right] \quad (38)$$

Siendo:

n El tamaño de la muestra.

n_0 Tamaño de muestra inicial estimado.

Z Nivel de confianza establecido.

S^2 Varianza inicial estimada.

N Tamaño de la población.

E Error aceptado.

n_1 Tamaño de la muestra preliminar para estimar S^2 y E .

Como en la mayoría de los casos se desconoce la varianza por tanto esta es estimada a partir de un porcentaje arbitrario de la población.

2.18 MUESTREO SISTEMÁTICO

Este tipo de muestreo consiste en seleccionar los elementos de la población en intervalos regulares.

Si definimos n como el tamaño de la muestra y a N como el tamaño de la población el tamaño del intervalo de selección se define como:

$$I = \frac{N}{n} \quad (39)$$

En caso de que el número no sea entero se redondea al superior. Posteriormente se selecciona un número aleatorio entre 1 e I a partir del cual se empieza a acumular el valor del intervalo tantas veces como tamaño de la muestra.

2.19 IMÁGENES DIGITALES

Una imagen digital es una función de intensidad de luz $f(x,y)$ donde x e y representan las coordenadas espaciales y el valor de f en un punto cualquiera (x,y) indica el nivel de gris [20], en caso de imágenes monocromáticas. Esta función discreta tanto para los niveles de gris como para las coordenadas permite conocer el valor del pixel en una posición dada (x,y) tal como se muestra en la Figura 11.

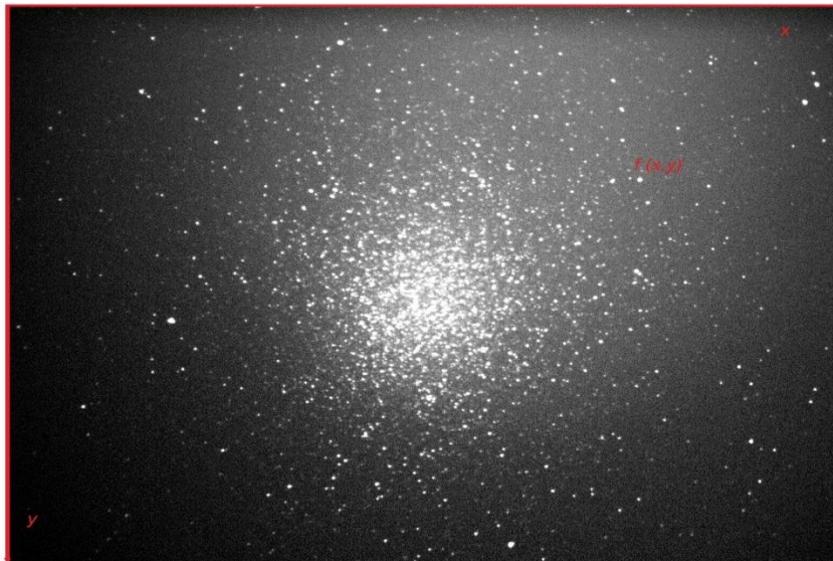


Figura 11. Fotografía del cúmulo globular Omega Centauri tomada desde el observatorio astronómico de la Universidad Tecnológica de Pereira.

3. METODOLOGÍA

En el presente capítulo se expone la metodología para la recolección de información, la instrumentación utilizada y el desarrollo de los algoritmos que implementan las estrategias de análisis que se explicaron en el marco teórico.

Para el desarrollo de este trabajo de grado los datos fueron adquiridos en el OAUTP; estos claro está, corresponden a los diferentes tipos de imágenes: de cuerpos celestes (estrellas), fondo de cielo y de *darks*.

3.1 DESARROLLO DE LOS ALGORITMOS

Inicialmente los algoritmos son escritos en *Matlab* de manera que realicen las diferentes pruebas tanto de bondad y ajuste como del análisis de *clústers*. *Matlab* es escogido debido a su gran versatilidad en campos ingenieriles, investigativos y científicos, y particularmente al permitir un manejo amigable de las imágenes digitales. Luego de que los scripts son construidos estos son probados en las fotografías adquiridas.

Como se mencionó anteriormente, el marco teórico soporta algunos de los conceptos claves en el desarrollo de los programas; tales como:

- Prueba Kolmogorov-Smirnov.
- Prueba Chi cuadrado.
- Método del promedio.
- Método del centroide.
- Muestreo aleatorio simple.

Una descripción de las ideas propuestas, desarrolladas y evaluadas; así como también, una breve explicación del código creado se muestra a continuación.

3.2 RECOLECCIÓN DE LA INFORMACIÓN

Con el objetivo de obtener los datos sobre los cuales realizar los respectivos estudios, se tomaron los siguientes tipos y cantidades de imágenes:

- 180 fotografías de *Darks*.

- 10 fotos de *Alnitak*⁵.

3.2.1 Dark frames

Para determinar la distribución que siguen las cuentas registradas con el obturador de la cámara cerrado, impidiendo que ingrese luz al sensor, se adquirieron 180 fotografías en tres jornadas diferentes (tres días arbitrarios); en cada una de estas jornadas se tomaron 60 imágenes a fin de apreciar el comportamiento del sensor.

La información para esta investigación fue adquirida con una cámara SBIG ST-2000XM. Cámara cuyo rango de obturación puede ser variado de 0.001 a 3,600 segundos; generando 3'600,000 exposiciones diferentes. Debido a esto, se acordó que el tiempo de exposición fuese de 1 a 600 segundos; además, estos últimos son los periodos que mayoritariamente se manejan durante la toma de fotografías en el OAUTP.

Una vez seleccionado el intervalo, 1 a 600 segundos, se decide que solo se tomaran 60 fotografías; es decir, el 10% del total de los tiempos establecidos. Ello debido a que no es usual tomar fotografías durante más de dos días seguidos en jornada continua, la idea detrás es adquirir las imágenes sin apagar la cámara, simulando una jornada de observación.

3.2.2 Procedimiento para la obtención de los Darks

Para realizar la recolección, la cámara es encendida y conectada a un computador de escritorio. Se ajustan algunos parámetros de refrigeración y se espera a que la temperatura de la cámara se estabilice. Luego de ello se realizan algunas pruebas para verificar que no exista ingreso de luz al sensor, posteriormente se procede con la toma de fotografías. Durante este proceso la temperatura es constantemente monitoreada ya que el instrumento ha de estar a diez grados por debajo de la temperatura ambiente. Como la temperatura varia a medida que la adquisición de fotografías avanza, se decide que este ajuste solo se realizará cada 0.5°C del valor inicial registrado.

Para la primera jornada, y ya establecido que se realizará un muestro sistemático con 60 intervalos, se selecciona un aleatorio de 1 a 10, siendo este igual a 3, y se construye una serie de números dados por la siguiente ecuación que corresponden con las exposiciones a realizar.

⁵ Alnitak es un sistema triple de estrellas en la constelación de orión que junto con Alnilam y Mintaka forman el cinturón del gigante.

$$t_{exp}(n) = 10n - 7 \quad n = 1, 2, \dots, 60 \quad (40)$$

Donde

t_{exp} Corresponde al tiempo de exposición.

El orden en que la adquisición de estas imágenes es llevada a cabo también es dado en la expresión anterior.

La temperatura durante la toma de las fotografías para esta primera jornada varió entre 20.5°C a 23.2°C y como se mencionó anteriormente, esta temperatura es monitoreada para realizar los ajustes necesarios a la cámara y evitar con ello un mal funcionamiento de la misma.

Para la segunda jornada se tomaron las mismas cantidades de fotografías; es decir 60, bajo los mismos tiempos de exposición y orden que en la primera. Con el fin de determinar si los resultados podrían ser repetibles. La temperatura en esta ocasión fluctuó entre los 22.5°C y 22.9°C.

Para la tercera y última jornada, al igual que en las anteriores se toman 60 imágenes, pero esta vez se toman tiempos diferentes de exposición. Para ello, nuevamente se genera un aleatorio entre 1 y 10, siendo este igual a 7, y se procede a la captura de las imágenes mediante la siguiente expresión:

$$t_{exp}(n) = 10n - 3 \quad n \sim U(1, 60) \quad (41)$$

Donde n sigue una distribución uniforme entre 1 y 60. Esto garantiza el tiempo de las exposiciones a realizar así como también el orden en que se va a llevar a cabo cada una de ellas. Nuevamente la temperatura es monitoreada para llevar un control del dispositivo registrando valores entre 20.4°C y 21.3°C.

3.2.3 Recolección de las fotos para las estrellas y el fondo de cielo

Para estas pruebas se decidió fotografiar *Alnitak*, la estrella más al sur de la terna que forma el cinturón de la constelación de orión, tomándose diez fotografías.

Esta se escogió debido a su posición en el cielo al momento de realizar la captura, ubicándose cerca al zenit; además, las estrellas que la rodean son de fácil identificación simplificando las pruebas a realizar.

3.2.4 Procedimiento para la toma de *Alnitak*

Para realizar estas adquisiciones la cúpula del OAUTP se abre con el fin de que el telescopio adquiera la temperatura ambiente.

Posteriormente se procede a orientar el instrumento de manera que realice el seguimiento de la estrella de forma correcta; ello se realiza mediante el método de alineación con dos estrellas. Inicialmente el telescopio es ubicado en *Home position* (ascensión recta 0 y declinación 90). Se escogen dos estrellas que en este caso fueron *Capella* y *Canopus* y se le da la orden al telescopio que se dirija a la primera de ellas; se enfoca dicha estrella en el ocular y se ejecuta sincronizar; se realizan los mismos pasos con la segunda estrella. El telescopio está ya listo para la captura de las imágenes.

Una vez la alineación es culminada con éxito se procede a la adquisición de las imágenes. Se enfoca *Alnitak* en el ocular paso seguido se instala la cámara con el reductor focal.

Luego que la cámara está instalada el objeto es enfocado. Por último la cámara es orientada de manera que la parte superior de la fotografía corresponda con el norte mientras que su izquierda corresponda con el este.

Listos los pasos anteriores se fotografía la estrella.

3.3 INSTRUMENTACIÓN

Los instrumentos principalmente usados durante la adquisición de la información del presente trabajo fueron:

- Telescopio Meade LX200 de 406.4mm de diámetro que utiliza tecnología Schmidt-Cassegrain; con una longitud focal de 4064 mm y una resolución de 0.28 segundos de arco.
- Cámara SBIG ST2000XM que emplea un sensor CCD KAI 2000M de dimensiones 11.8 x 9.0 mm; la cantidad de pixeles es de 1600x1200 y cuyos tamaños son del orden de 7.4x7.4 μ .
- Un Reductor focal Celestron f/6.3.
- Pantalla táctil de la estación meteorológica *PCE-FWS 20* para medir la temperatura.

3.4 CONSIDERACIONES DE LAS PRUEBAS

3.4.1 Para los darks

Para realizar el muestreo y con el fin de tener los elementos listados y numerados para su posterior selección, los pixeles en la imagen son “etiquetados” de izquierda a derecha y de arriba a abajo de 1 a 1'920,000; número que corresponde con la dimensión de la imagen 1,200 X 1,600 pixeles y por tanto con la población.

De esta manera a cada aleatorio generado en dicho intervalo corresponderá un único pixel. Además de ello la confianza utilizada para el muestreo aleatorio simple es del 97% y el error igual al 1% del promedio estimado en un estudio previo que realiza el algoritmo en el que se generan 3000 aleatorios y se seleccionan los pixeles correspondientes. Luego se calcula la varianza y el promedio, parámetros que son desconocidos, y que son necesarios para el cálculo del tamaño de la muestra.

Las hipótesis a contrastar son las siguientes:

1. $H_0: x \sim N(\mu, \sigma)$

$H_1: x \neq N(\mu, \sigma)$

2. $H_0: x \sim \sim P(\lambda)$

$H_1: x \neq \sim P(\lambda)$

3. $H_0: x \sim Wiubull(\alpha, \beta)$

$H_1: x \neq Wiubull(\alpha, \beta)$

4. $H_0: x \sim Gamma(\alpha, \beta)$

$H_1: x \neq Gamma(\alpha, \beta)$

5. $H_0: x \sim LogN(\mu, \sigma)$

$H_1: x \neq LogN(\mu, \sigma)$

Donde la variable aleatoria x representa las cuentas registradas por los pixeles.

Estas hipótesis son propuestas debido a la semejanza de los datos observando un comportamiento similar a ellas y por tanto se pueden ajustar a alguna de estas.

Estas hipótesis nulas serán contrastadas tanto con el estadístico *Chi-cuadrado* como con el *Kolmogorov-Smirnov* que servirán para ratificar o no los resultados. El nivel de significancia asumido fue del 95%.

3.4.2 Para la distribución del fondo de cielo y la estrella

Para determinar estas distribuciones se selecciona una porción de imagen que contenga tanto a la estrella como pixeles con valores de fondo de cielo tal y como se muestra en la Figura 12.



Figura 12. Segmento de imagen seleccionada en el que se aprecia la estrella y el fondo de cielo.

Al igual que como se hizo con los *darks*, los pixeles en esta imagen son “etiquetados” izquierda a derecha y de arriba a abajo para su selección. La confianza para el muestreo aleatorio simple nuevamente es del 97% y el error igual al 1% un promedio estimado.

Nuevamente las hipótesis a contrastar tanto para la estrella como para el fondo de cielo son:

1. $H_0: x \sim N(\mu, \sigma)$

$$H_1: x \neq N(\mu, \sigma)$$

2. $H_0: x \sim \sim P(\lambda)$

$$H_1: x \neq \sim P(\lambda)$$

3. $H_0: x \sim Wiubull(\alpha, \beta)$

$$H_1: x \neq Wiubull(\alpha, \beta)$$

4. $H_0: x \sim Gamma(\alpha, \beta)$

$$H_1: x \neq \text{Gamma}(\alpha, \beta)$$

5. $H_0: x \sim \text{LogN}(\mu, \sigma)$

$$H_1: x \neq \text{LogN}(\mu, \sigma)$$

Donde la variable aleatoria x representa las cuentas registradas por los pixeles.

Y se contrastan con los estadísticos que se han venido trabajando a lo largo de este documento y bajo el mismo nivel de significancia.

3.4.3 Para la distancia entre dos estrellas

Antes de ejecutar los algoritmos para realizar las medidas de distancia entre dos o más cuerpos y calcular así las coordenadas ecuatoriales del primero con respecto a los demás; las imágenes son sometidas a un procedimiento previo que busca una adecuada disposición de estas.

Como se ha mencionado las capturas deben de estar alineadas; es decir, la parte superior e izquierda de la imagen han de coincidir lo más exactamente posible con el norte y el este geográfico respectivamente.

Para garantizar dicha configuración, las tomas se equiparan con las posiciones que deben de tener los astros fotografiados en dicha región. Esto se hace con la ayuda del *Stellarium*⁶, en dicho software se identifica *Alnitak* (la estrella más grande en ambas imágenes) y sus vecinas tal y como se muestra en la Figura 13.

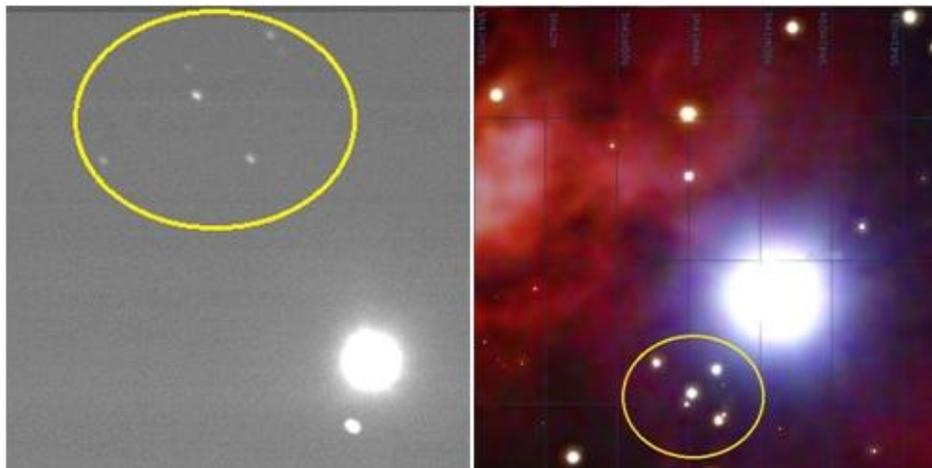


Figura 13. **Comparación de la posición de las estrellas.**

⁶ Stellarium es un software libre que simula un planetario virtual; pudiéndose visualizar gran cantidad de objetos como estrellas, planetas, nebulosas, galaxias; a demás de brindar información de estos, sus coordenadas, distancias, paralaje entre otras.

La comparación es realizada mostrando que las estrellas de la fotografía al lado izquierdo dentro del círculo amarillo se encuentran en una posición diferente de las estrellas de la imagen a la derecha; se puede apreciar claramente como se encuentran en la parte superior izquierda de la imagen. Esto se debe a que el sistema óptico de telescopio genera una imagen invertida del objeto. Por tal razón; la imagen es girada 180° .

Luego de que la captura es girada. Se determina el ángulo que hay entre el norte y cualquiera de las estrellas de la imagen obtenida del *Stellarium*. Dicho ángulo se hará coincidir con la toma realizada desde el OAUTP.

Para esto se procede a calcular el ángulo α entre el eje norte-sur y los objetos 1 y 2 como se muestra en la Figura 14.

Cabe aclarar que la línea norte-sur y la recta que forman el ángulo α pasan por el centro de la estrella.

El ángulo es entonces determinado al construir un triángulo rectángulo en la Figura 14 y cuyos lados: adyacente y opuesto pueden ser medidos por la cantidad de píxeles que contienen para luego hallar dicha incógnita con una función trigonométrica.

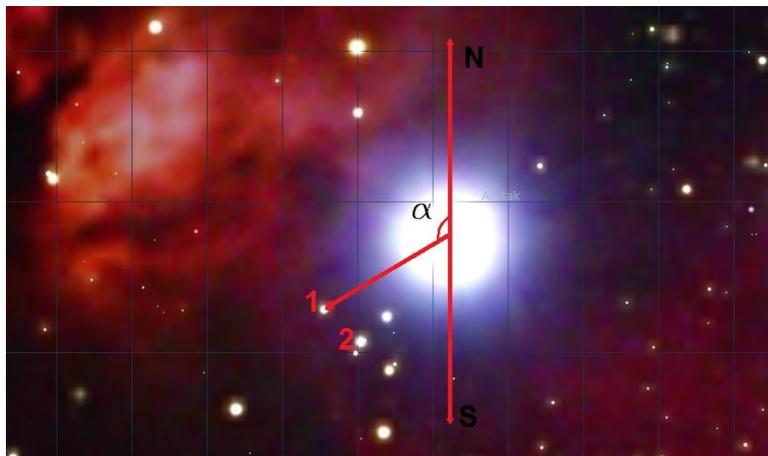


Figura 14. Cálculo del ángulo α .

Análogamente se hace lo mismo con la fotografía tomada desde el OAUTP como se muestra en la Figura 15.

Estos ángulos, α y α' , son determinados ya que dan una idea de donde se encuentran las estrellas y donde deberían estar; así que su diferencia indica cuantos grados ha de ser girada la imagen de la misma forma que el sentido del giro.

A continuación se muestran los valores obtenidos al realizar los cálculos:

$$\alpha_1 = 120.329^\circ$$

$$\alpha_2 = 126.748^\circ$$

$$\alpha'_1 = 140.135^\circ$$

$$\alpha'_2 = 146.619^\circ$$

Donde:

α_1 Es el ángulo medido del eje N-S al objeto 1 de la Figura 14.

α_2 Es el ángulo medido del eje N-S al objeto 2 de la Figura 14.

α'_1 Es el ángulo medido del eje N-S al objeto 1 de la Figura 15.

α'_2 Es el ángulo medido del eje N-S al objeto 1 de la Figura 15.

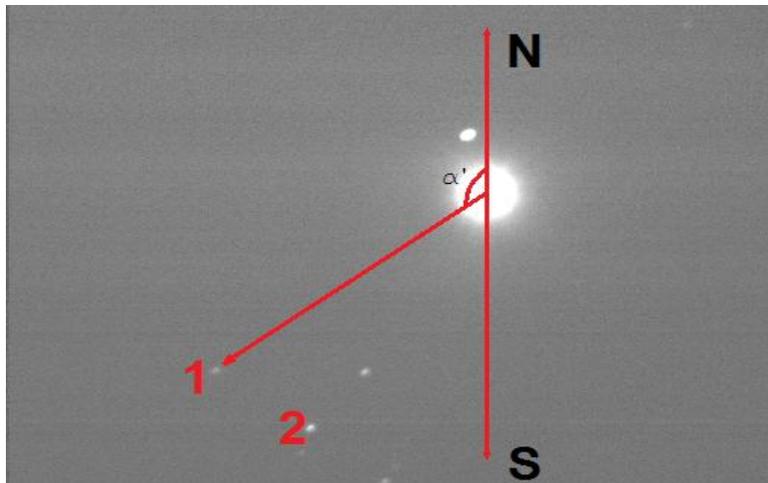


Figura 15. Ángulo α en la imagen tomada desde OAUTP.

Siendo entonces que el ángulo de rotación, $\angle\rho$, será igual a la diferencia entre α y α' para los objetos 1 y 2.

$$\angle\rho = -6.419^\circ$$

$$\angle\rho' = -6.484^\circ$$

Con estos valores se determina un promedio aritmético que indicará el valor del ángulo que se rotará en la imagen.

$$\angle\rho_p = -6.4515^\circ$$

La Figura 16 muestra los procesos a realizar para una adecuada disposición de la imagen.

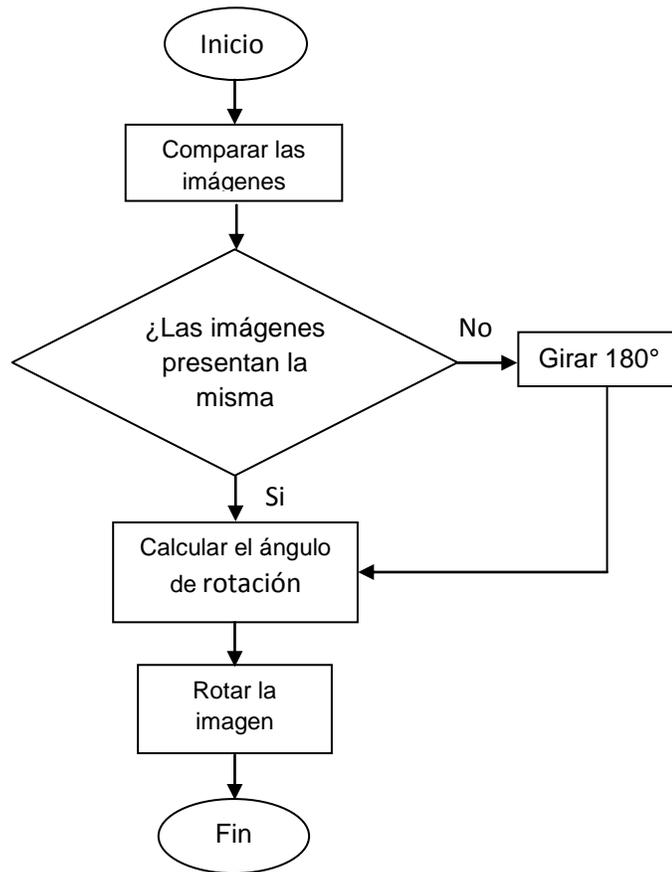


Figura 16. Diagrama de flujo para la adecuada posición de la imagen.

En la Figura 17 se aprecia las transformaciones que se le realizan a la fotografía. En el recuadro superior izquierdo se muestra la imagen original; es decir, la adquirida en el OAUTP. Luego esta imagen es girada 180° tal y como se puede apreciar en el recuadro superior derecho. Por último se rota -6.4515 ° como enseña en el recuadro inferior.

Finalmente un parámetro que se ha de tener en cuenta para la exitosa realización de la prueba es el determinar los segundos de arco por pixel. Esto define la cantidad angular abarcada por cada elemento y viene dada por la siguiente expresión:

$$\frac{sarc}{px} = \frac{206.265 \times Tp}{df} \quad (42)$$

Donde:

$sarc/px$ Segundos de arco por pixel.

Tp Tamaño del pixel en micras.

df Distancia focal del telescopio en milímetros.

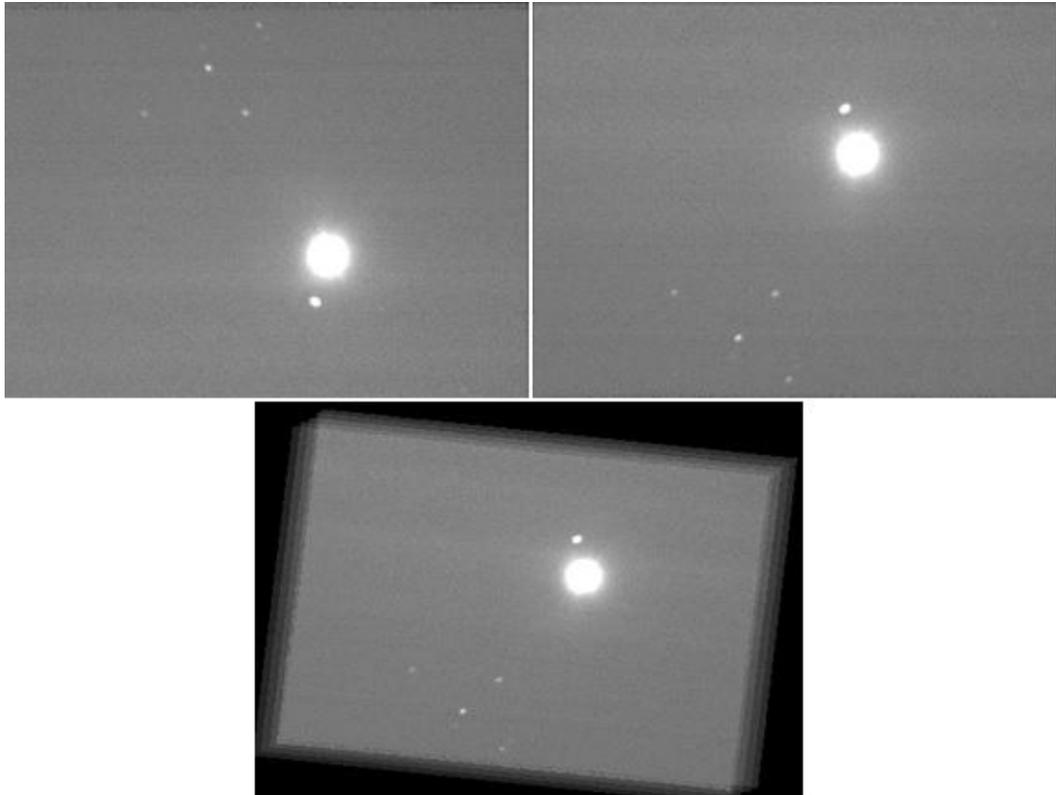


Figura 17. Secuencia de los ajustes realizados a la imagen.

Recordando que la distancia focal para el Meade LX 200 es de 4064mm, el tamaño de pixel de la cámara es de 7.4 micras y teniendo en cuenta que se utilizó un reductor focal f/6.3. Se realiza el cálculo obteniendo el siguiente resultado.

$$sarc/px = 0.59616$$

3.5 METODOLOGÍAS DE LAS PRUEBAS

A continuación se describen las metodologías que se llevaron a cabo para el cumplimiento de los objetivos planteados en el trabajo investigativo.

3.5.1 Pruebas realizadas sobre las imágenes darks

Para este caso, se realizaron dos pruebas para identificar el tipo de distribución que puedan llegar a seguir las cuentas de los pixeles:

- Pruebas de bondad y ajuste a un vector construido con muestras tomadas de cada imagen.
- Bondad y ajuste para valores registrados por un mismo pixel.

3.5.1.1 Pruebas de bondad y ajuste a un vector construido con muestras tomadas de cada imagen

La primera idea a desarrollar consistió en tomar una muestra aleatoria de cada una de las sesenta fotografías y observar el comportamiento en conjunto de estos datos.

Para ello, se corre el código Prueba_Funciones (ver Anexo 1.1) el cual calcula los estadísticos de prueba Chi cuadrado y Komogorov-Smirnov para cada una de las funciones de distribución propuestas en las hipótesis nulas.

En dicho código el primer paso consiste en calcular el tamaño de la muestra para cada imagen mediante la función Tama_muestra (ver Anexo 1.2). Como se desconocen la varianza y el promedio poblacionales estos son estimados generando 3000 (0.15% de la población) números aleatorios del 1 al 1'920,000. Todos ellos diferentes para cada imagen mediante la función aleat_sin_rept (Anexo 1.3), luego se buscan las coordenadas de estos aleatorios en la imagen, identificando así el pixel asociado mediante la función Coor_Pixel (Anexo 1.4).

Teniendo en cuenta que solo serán seleccionados los pixeles cuyo número de cuentas sea menor o igual a 5000, ya que para cuentas superiores se consideran datos que no pertenecen al fenómeno de estudio y deben ser originados por causas diferentes. Para los pixeles que exceden el valor supuesto se crea un nuevo aleatorio diferente a los ya generados y se verifica la condición; repitiendo este paso hasta que tal condición se cumpla.

Una vez seleccionados los 3000 pixeles para cada imagen se procede a estimar los promedios y las varianzas, esta estimación es realizada debido a que se desconoce la distribución que sigue la población y a que no se dispone de un estudio previo con valores para los parámetros mencionados.

El error será igual al 1% del promedio y se maneja una confianza para el muestreo del 97%. Con la varianza, el error y la confianza se procede a calcular un tamaño

de muestra inicial dado en la ecuación (38) para luego estimar el tamaño de muestra final de la ecuación (36) para cada una de las imágenes.

Con los tamaños de muestra calculados se procede a crear la matriz M_{val_Pixel} que contendrá los valores de los pixeles seleccionados de cada imagen.

3.5.1.1.1 Manejo de atípicos

Antes de crear dicha matriz se presenta la dificultad de identificar qué cantidad de cuentas han de ser consideradas para este estudio, aunque es claro que cuentas superiores a 5000 no corresponden con el fenómeno no es sencillo concluir lo mismo para cuentas entre 1900 y 3000.

Para superar este impase se grafica un histograma con pixeles con cuentas menores o iguales a 5000 tomando los tamaños de las muestras calculadas en el paso anterior tal como se muestra en la Figura 17.

El histograma muestra el comportamiento de los datos. Se puede apreciar como estas cifras están agrupadas principalmente entre las 900 y 1400 cuentas aproximadamente, mostrando pocos valores para las cuentas superiores.

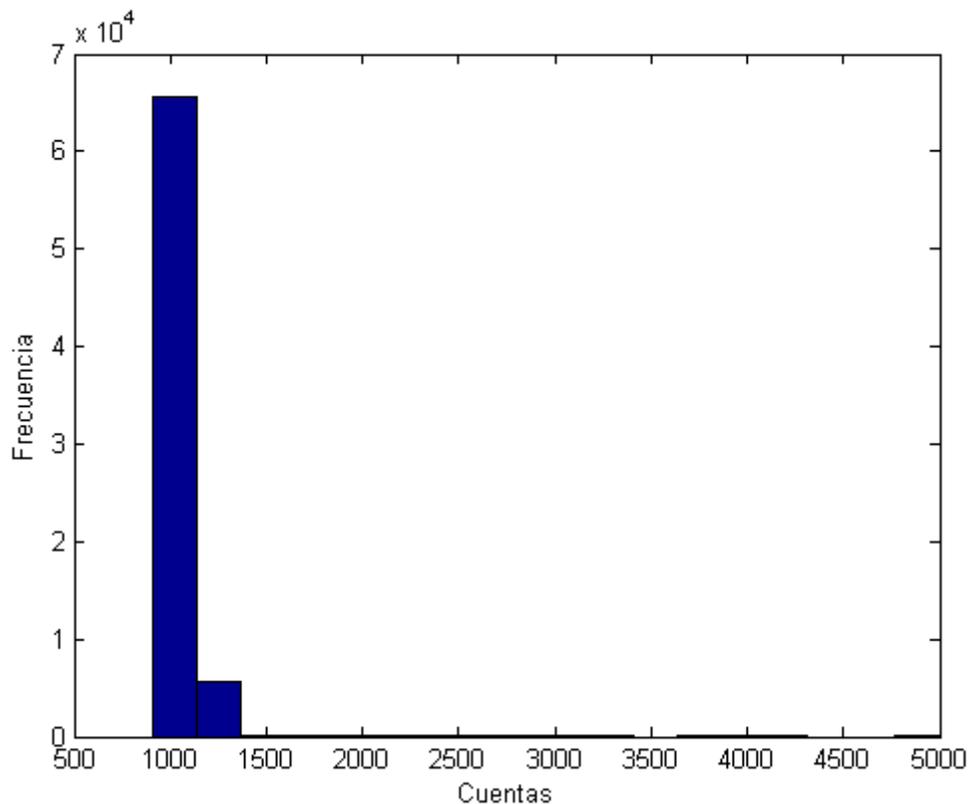


Figura 18. Histograma con valores de pixeles menores o iguales a 5000 cuentas.

La Tabla 1 muestra las clases, los límites, las frecuencias y las frecuencias relativas del histograma anterior. Ahora bien, lo que se plantea es: intervalos con una frecuencia relativa baja (menos del 0.03%) serán considerados valores que tiene causas diferentes a la estudiada; para frecuencias relativas superiores se consideraran valores atípicos no muy graves y que pueden deberse a la distribución de los datos.

De esta manera, empírica, se decidió que solo se tendrán en cuenta pixeles con cuentas menores o iguales a 1600.

<i>Clase</i>	<i>Lim Inf</i>	<i>Lim Sup</i>	<i>Frecuencia</i>	<i>Frecuencia Relativa</i>	<i>Frecuencia Acumulada</i>
1	914.00	1140.72	65640	0.9186	0.91863
2	1140.72	1367.44	5723	0.0801	0.99873
3	1367.44	1594.17	25	0.0003	0.99903
4	1594.17	1820.89	17	0.0002	0.99927
5	1820.89	2047.61	21	0.0002	0.99961
6	2047.61	2274.33	9	0.0001	0.99973
7	2274.33	2501.06	2	0.0000	0.99976
8	2501.06	2727.78	3	0.0000	0.99980
9	2727.78	2954.50	4	0.0001	0.99986
10	2954.50	3181.22	3	0.0000	0.99990
11	3181.22	3407.94	1	0.0000	0.99992
12	3407.94	3634.67	0	0.0000	0.99992
13	3634.67	3861.39	2	0.0000	0.99994
14	3861.39	4088.11	1	0.0000	0.99996
15	4088.11	4314.83	2	0.0000	0.99999
16	4314.83	4541.56	0	0.0000	0.99999
17	4541.56	4768.28	0	0.0000	0.99999
18	4768.28	4995.00	1	0.0000	1.00000
N° Datos			71454		

Tabla 1. Método para la identificación de atípicos.

Conocidos los tamaños de muestra para cada imagen y las cuentas que serán medidas, el código crea la matriz: M_val_Pixel mediante la función Misma_muestra (ver anexo 1.5); esta función genera una cantidad de aleatorios igual al tamaño de la muestra calculada para cada una de las imágenes, y por tanto selecciona una cantidad diferente de pixeles para cada una de ellas, creando la matriz antes mencionada y considerando que solo serán seleccionados los pixeles con cuentas menores o iguales a 1600.

Ahora bien, una vez construida la matriz M_val_Pixcel . Se procede a realizar las pruebas *Kolmogorov-Smirnov* y *Chi-cuadrado* con los datos contenidos en dicho arreglo, la idea con las dos pruebas es contrastar los resultados de ambas, mediante al función *prueba_bya* (ver Anexo 1.6), así mismo dicha función grafica el histograma de frecuencias tal y como se muestra en la Figura 18. En ella se puede apreciar y dar una idea de cómo están distribuidos los datos notando una cola liviana a la derecha del grafico. Los valores correspondientes a dicho grafico se aprecian en la Tabla 2. Cabe hacer la aclaración que por cada vez que se ejecute el algoritmo tanto las frecuencias, los tamaños de muestra, las varianzas y los promedios presentarán valores diferentes debido a la aleatoriedad del proceso y por tanto los histogramas obtenidos pueden variar un poco.

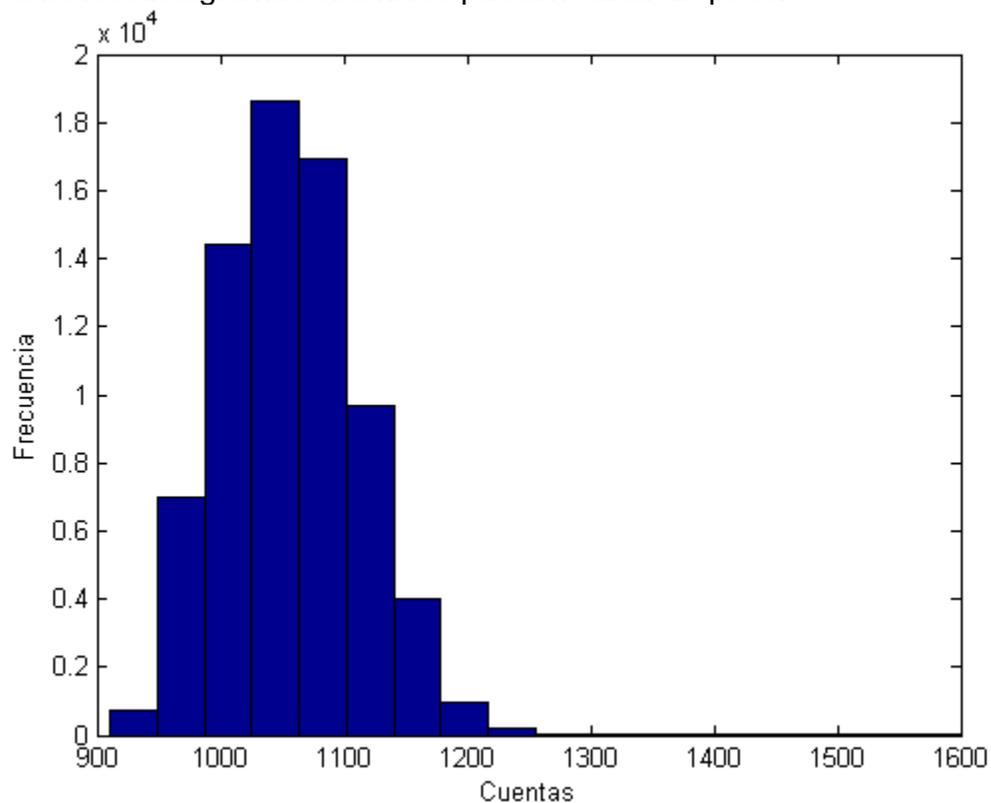


Figura 19. Histograma con valores de píxeles menores a 1600 cuentas.

Para crear el histograma se calculan las clases, los límites de estas, así como también las frecuencias correspondientes a cada una de ellas; mediante la función *Frecuencia_intervalos* (Anexo 1.7), el número de clases es obtenido mediante la regla de Sturges, además esta información servirá para realizar las prueba *Chi-Cuadrado*.

Las funciones: K_S_Nor , $K_S_Weibull$, K_S_Gamma , $K_S_Poisson$ calculan el estadístico de la prueba *Kolmogorv-Smirnov* para las distribuciones normal, *weibull*, *gamma* y *poisson* respectivamente (Anexos 1.8, 1.9, 1.10 y 1.11), para determinar el estadístico para la distribución *lognormal* se calcula el logaritmo natural de la matriz M_val_Pixcel y a estos valores se les realiza la prueba de normalidad. En estas pruebas los datos son ordenados ascendentemente para

luego calcular la máxima diferencia entre las frecuencias empíricas y las frecuencias teóricas como bien se plantea en la ecuación (15). Los resultados son mostrados en la matriz *Matriz_D* en la que las filas indicaran el valor del estadístico para la *normal*, *gamma*, *weibull*, *lognormal* y *poisson*.

Una vez terminada la prueba *K-S* se procede con la prueba Chi Cuadrado. Para esta prueba y conocidas la cantidad de clases, los límites de estas y sus respectivas frecuencias; se estima el estadístico mediante las funciones *Chi_Cuadra_Normal*, *Chi_Cuadra_Gamma*, *Chi_Cuad_weibull*, *Chi_Cuadra_Poisson* (Anexos 1.12, 1.13, 1.14 y 1.15) calculándolo mediante la ecuación (19) únicamente en aquellos casos en que se aplique una distribución continúa a datos discretos; es decir mediante la corrección de Yates.

Clase	Lim Inf	Lim Sup	Frecuencia
1	911.00	949.22	713.00
2	949.22	987.44	6973.00
3	987.44	1025.67	14414.00
4	1025.67	1063.89	18618.00
5	1063.89	1102.11	16931.00
6	1102.11	1140.33	9670.00
7	1140.33	1178.56	3995.00
8	1178.56	1216.78	955.00
9	1216.78	1255.00	200.00
10	1255.00	1293.22	49.00
11	1293.22	1331.44	28.00
12	1331.44	1369.67	12.00
13	1369.67	1407.89	7.00
14	1407.89	1446.11	9.00
15	1446.11	1484.33	12.00
16	1484.33	1522.56	2.00
17	1522.56	1560.78	7.00
18	1560.78	1599.00	4.00

Tabla 2. Frecuencias resultantes al eliminar los atípicos.

Ahora bien cada una de estas funciones agrupa aquellas clases en caso de que su frecuencia esperada sea menor a 5 mediante la función *intervalo_corre* (Anexo 1.16), de esta forma se calcula el estadístico *Chi*; así como también, el número de clases resultante de cada una de las pruebas. El diagrama de flujo de la Figura 19 ilustra de manera general el proceso.

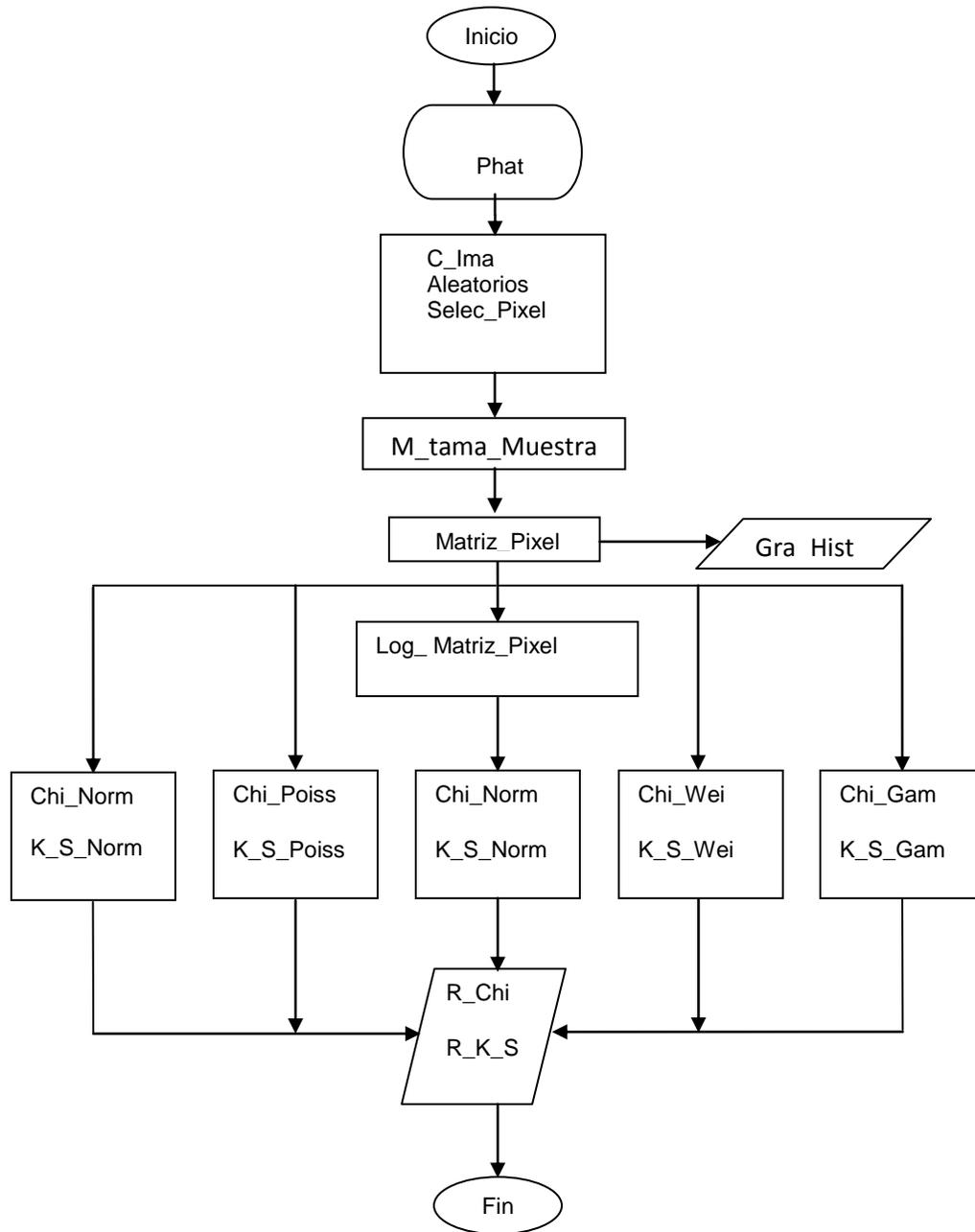


Figura 20. Diagrama de flujo. Prueba de bondad y ajuste para muestras tomadas de cada imagen.

Donde:

Phat: Dirección de la carpeta donde se encuentran las imágenes.

C_Ima: Cargar las imágenes.

Aleatorios: Numeros aleatorios entre el 1 y el tamaño de la población y tantos como tamaño de muestra.

Selec_Pixel: Se seleccionan los pixeles que correspondan a los aleatorios generados.

Matriz_Pixel: Se construye una matriz con los valores de dichos pixeles.

Log_Matriz_Pixel: Se le toma el logaritmo natural a la matriz de pixeles.

K_S_Norm: Realiza una prueba Kolmogorov-Smirnov para normalidad.

K_S_Gam: Realiza una prueba Kolmogorov-Smirnov para distribución gamma.

K_S_Wei: Realiza una prueba Kolmogorov-Smirnov para distribución weibull.

K_S_Poiss: Realiza una prueba Kolmogorov-Smirnov para una distribución poisson.

Chi_Norm: Realiza una prueba chi-cuadrado para normalidad.

Chi_Gam: Realiza una prueba chi-cuadrado para distribución gamma.

Chi_Wei: Realiza una prueba chi-cuadrado para distribución weibull.

Chi_Poiss: Realiza una prueba chi-cuadrado para distribución de poisson.

R_Chi: Presenta los resultados en una matriz para la prueba chi cuadrado.

R_K_S: Presenta los resultados en una matriz para la prueba Kolmogorov-Smirnof.

Gra_Hist: Grafica el hitograma.

M_tama_Muestra: Matriz con los tamaños de la muestra para cada una de las imagenes.

3.5.1.2 Bondad y ajuste para valores registrados por un mismo pixel

Por otro lado, se plantea medir los registros de un mismo pixel en cada una de las 60 exposiciones de forma individual e intentar identificar su posible distribución. Para ello se escogen 10 pixeles de forma aleatoria y se realizan las medidas de cada uno de estos pixeles en cada una de las sesenta exposiciones creando una matriz de 60X10. A cada una de estas columnas de dicha matriz se les realizarán las pruebas de bondad y ajuste.

Los 10 aleatorios generados que representan 10 pixeles del sensor se muestran en la Tabla 3.

N° Aleatorio	Aleatorio
1	8898
2	162117
3	204774
4	498952
5	767583
6	1487829
7	1536132
8	1569223
9	1667894
10	1846845

Tabla 3. Aleatorios.

Estos 10 pixeles son los mismos para la información obtenida en cada una de las tres jornadas de la toma de *darks*. La metodología es similar a la que se ha venido trabajando y se describe a continuación:

El código desarrollado para este objetivo inicia con el script *Val_mismo_pixel* (Anexo 2.1). Inicialmente, mediante la función *Mismos_pixels* (Anexo 2.2) carga las imágenes y selecciona un número que puede ser modificado de pixeles, para este caso los pixeles que se muestran en la Tabla 3, estos elementos son ubicados en las imágenes y medidos para cada una de las 60 fotografías y luego calcular los estadísticos *K-S* y *Chi* de la misma manera que la metodología anterior, es decir bajo las mismas consideraciones y con la misma función *prueba_bya* (Anexo 1.6).

Una vez estimados estos estadísticos y creadas sus respectivas matrices de resultados mediante la función *Res_Chi* (Anexo 2.3), se procede a comparar los valores calculados para cada una de las pruebas con el respectivo Chi crítico, teniendo en cuenta sus respectivos grados de libertad y nivel de confianza (aunque el nivel de confianza es el mismo los grados de libertad puede variar), si todas las hipótesis nulas son aceptadas se escoge aquella con el menor Chi calculado entregando las siguiente respuestas: cantidad de pixeles que se ajusta mejor a una distribución normal, a una poisson, a una gamma, a una weibull o a una lognormal; además, de un vector fila con valores de uno a seis en el que el uno indica que es normal, el dos gamma, el tres weibull, el cuatro lognormal, el cinco poisson y el seis no se ajusta a ninguna de las anteriores, y dado que el vector es de una fila n columnas, el número de la columna identificará al pixel con la distribución mencionada.

Para la prueba *K-S* la metodología es similar. Luego de estimar el estadístico se compara con el valor crítico y se cuentan cuantos pixeles presenta tal o cual comportamiento. Igualmente se genera el vector fila que indica la distribución que mejor se ajusta numéricamente hablando a los datos. La Figura 20 ilustra dicho proceso.

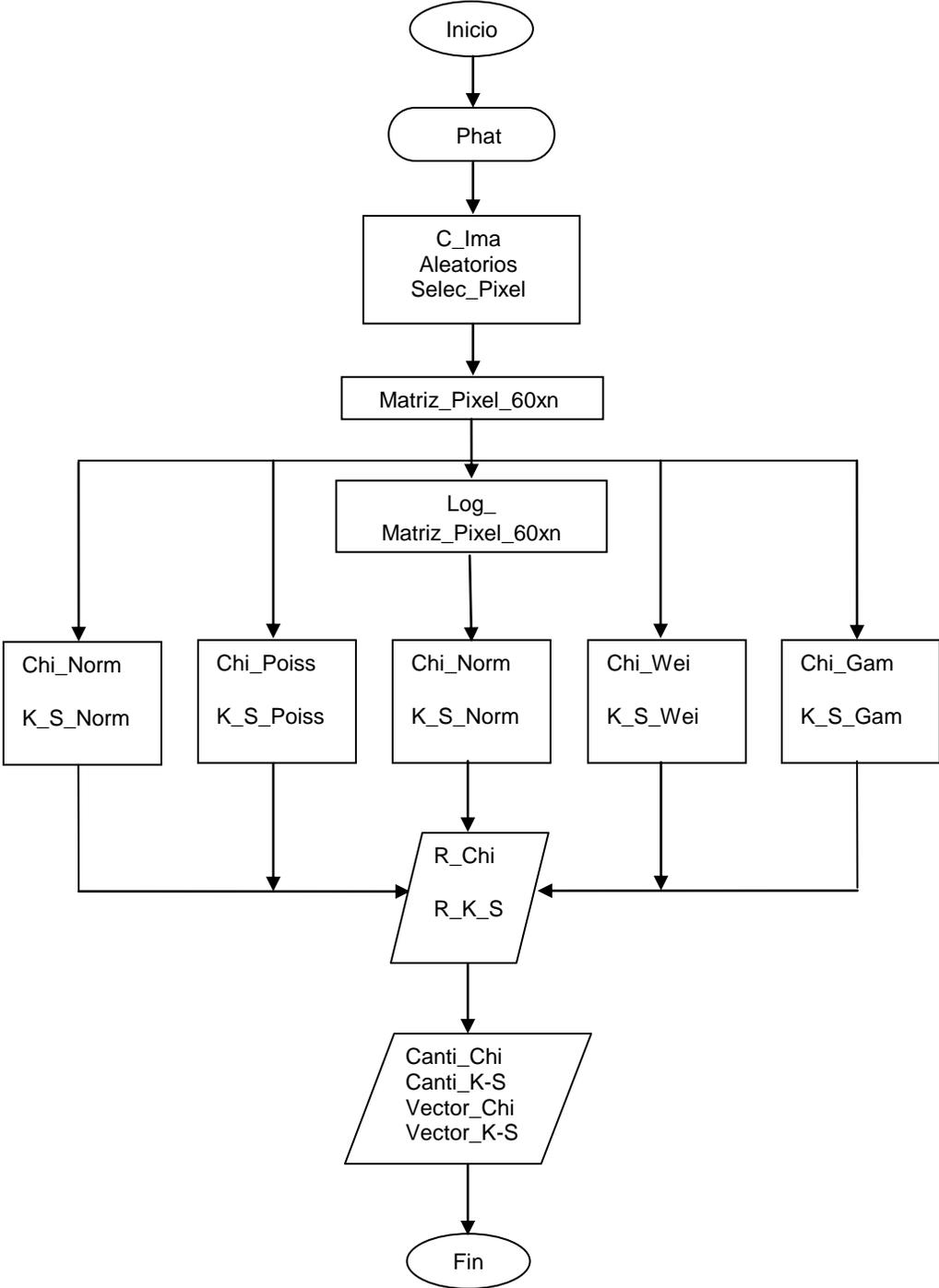


Figura 21. Diagrama de flujo para los valores registrados por un mismo pixel.

Donde:

Phat: Dirección de la carpeta donde se encuentran las imágenes.

C_Ima: Cargar las imágenes.

Aleatorios: Numeros aleatorios entre el 1 y el tamaño de la población y tantos como tamaño de muestra.

Selec_Pixel: Se seleccionan los pixeles que correspondan a los aleatorios generados.

Matriz_Pixel_60xn: Se crea la matriz 60xn.

Log_Matriz_Pixel_60xn: se le toma el logaritmo natural a: Matriz_Pixel_60xn.

K_S_Norm: Realiza una prueba Kolmogorov-Smirnov para normalidad.

K_S_Gam: Realiza una prueba Kolmogorov-Smirnov para distribución gamma.

K_S_Wei: Realiza una prueba Kolmogorov-Smirnov para distribución weibull.

K_S_Poiss: Realiza una prueba Kolmogorov-Smirnov para una distribución poisson.

Chi_Norm: Realiza una prueba chi-cuadrado para normalidad.

Chi_Gam: Realiza una prueba chi-cuadrado para distribución gamma.

Chi_Wei: Realiza una prueba chi-cuadrado para distribución weibull.

Chi_Poiss: Realiza una prueba chi-cuadrado para distribución de poisson.

R_Chi: Presenta los resultados en una matriz para la prueba chi cuadrado.

R_K_S: Presenta los resultados en una matriz para la prueba Kolmogorov-Smirnof.

Canti_Chi: Cantidad de pixeles que presentan una determinada distribución luego de realizar la prueba chi cuadrado.

Canti_K-S: Cantidad de pixeles que presentan una determinada distribución luego de realizar la prueba kolmogorov-Smirnov.

Vector_Chi: vector que indica la distribución que se ajusta a un pixel cualquiera según la prueba chi cuadrado.

Vector_K-S: vector que indica la distribución que se ajusta a un pixel cualquiera según la prueba K-S.

3.5.2 Distribución fondo de cielo y de estrella

Por último, para discernir que pixel hace parte de la estrella y cual no; se propone identificar la distribución que siguen los pixeles que forman tanto el fondo de cielo como los que forman el cuerpo celeste. Si se lograra identificar dichas distribuciones se plantea entonces que a un pixel el cual no se sabe de su pertenencia, se podría clasificar evaluando las cuentas registradas por él en las dos distribuciones y aquella con la probabilidad más alta dará una idea de su pertenencia.

Inicialmente se plantea identificar la posible distribución que pueda llegar a seguir el fondo de cielo, para ello se ejecuta el código *distri_fondo_cie* (Anexo 5) el cual toma una sección de la imagen tal y como se muestra en la Figura 22.



Figura 22. Fragmento de imagen utilizada para identificar el posible ajuste del fondo de cielo y de la estrella.

Luego genera 3000 pixeles de manera aleatoria para calcular el tamaño de la muestra; solo se consideran pixeles con cuentas menores o iguales a 450 ya que para estos valores se asume que solo son generados por el fondo de cielo mientras que para cuentas mayores se juzga que son afectadas por la estrella. Esto se realizó observando el comportamiento de los pixeles en las regiones más

alejadas de la estrella en donde se notó que las cuentas en aquellas regiones difícilmente superaban las 400 cuentas.

Una vez calculado el tamaño de la muestra se generan aleatorios y se ubican estos pixeles en la imagen para luego ser medidos y construir una matriz con dichos valores. A dicha matriz se le aplicará las pruebas de bondad y ajuste.

Los valores de salida del algoritmo son una matriz de 1x5 con los valores del estadístico *Chi* y *K-S* para la normal, gamma, weibull, lognormal y poisson respectivamente; además, una matriz con el numero de clases de la prueba chi para cada hipótesis es mostrada. La Figura 23 ilustra dicho proceso, en ella se define:

Phat: Dirección de la carpeta donde se encuentran las imágenes.

C_Ima: Cargar las imágenes.

Rc_Estl_FonC: Genera una sección rectangular de la imagen en la que se puede apreciar el fondo de cielo y la estrella.

M_pre_meyvar: realiza un muestreo previo para estimar la varianza y el promedio.

M_tama_Muestra: Calcula el tamaño de la muestra.

Aleatorios: Numeros aleatorios entre el 1 y el tamaño de la población y tantos como tamaño de muestra.

Selec_Pixel: Se seleccionan los pixeles que correspondan a los aleatorios generados.

Matriz_Pixel: Se construye una matriz con los valores de dichos pixeles.

Gra_Hist: Grafica el hitograma.

Log_Matriz_Pixel: Se le toma el logaritmo natural a la matriz de pixeles.

K_S_Norm: Realiza una prueba Kolmogorov-Smirnov para normalidad.

K_S_Gam: Realiza una prueba Kolmogorov-Smirnov para distribución gamma.

K_S_Wei: Realiza una prueba Kolmogorov-Smirnov para distribución weibull.

K_S_Poiss: Realiza una prueba Kolmogorov-Smirnov para una distribución poisson.

Chi_Norm: Realiza una prueba chi-cuadrado para normalidad.

Chi_Gam: Realiza una prueba chi-cuadrado para distribución gamma.

Chi_Wei: Realiza una prueba chi-cuadrado para distribución weibull.

Chi_Poiss: Realiza una prueba chi-cuadrado para distribución de poissn.

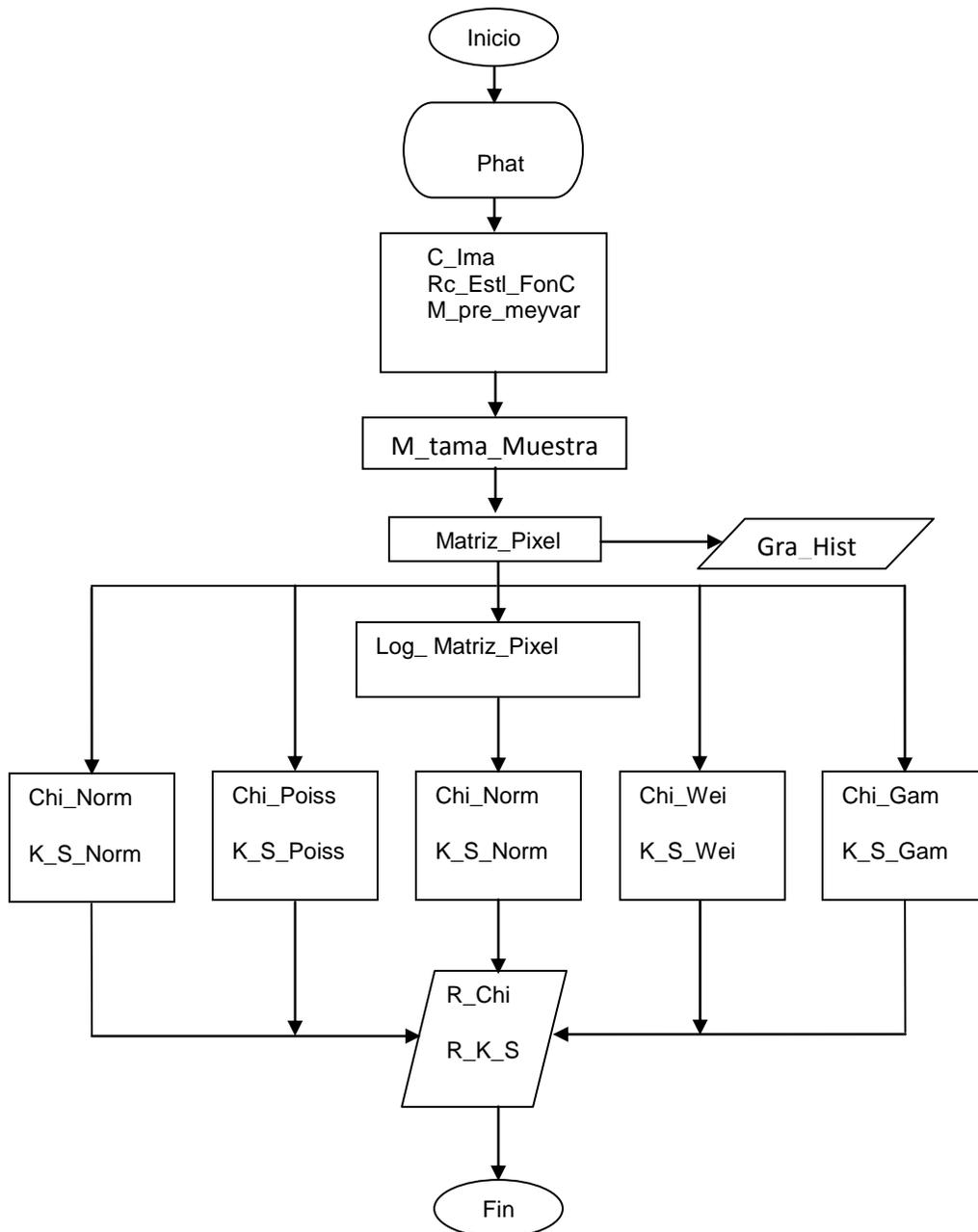


Figura 23. Diagrama flujo de los pasos para identificar la distribución del fondo de cielo.

R_Chi: Presenta los resultados en una matriz para la prueba chi cuadrado.

R_K_S: Presenta los resultados en una matriz para la prueba Kolmogorov-Smirnof.

El procedimiento para realizar estas pruebas a la estrella es similar. Para ello se corre el código *distri_estrella* (Anexo 6) que carga la imagen y crea un rectángulo con la estrella inscrita en el. Luego se seleccionan los pixeles con cuentas mayores a 10000, es decir pixeles con el 16% del brillo del mayor pixel. Y a dichos pixeles se les realiza las pruebas de bondad y ajuste de la misma manera que se ha venido planteando. La Figura 24 esclarece lo propuesto.

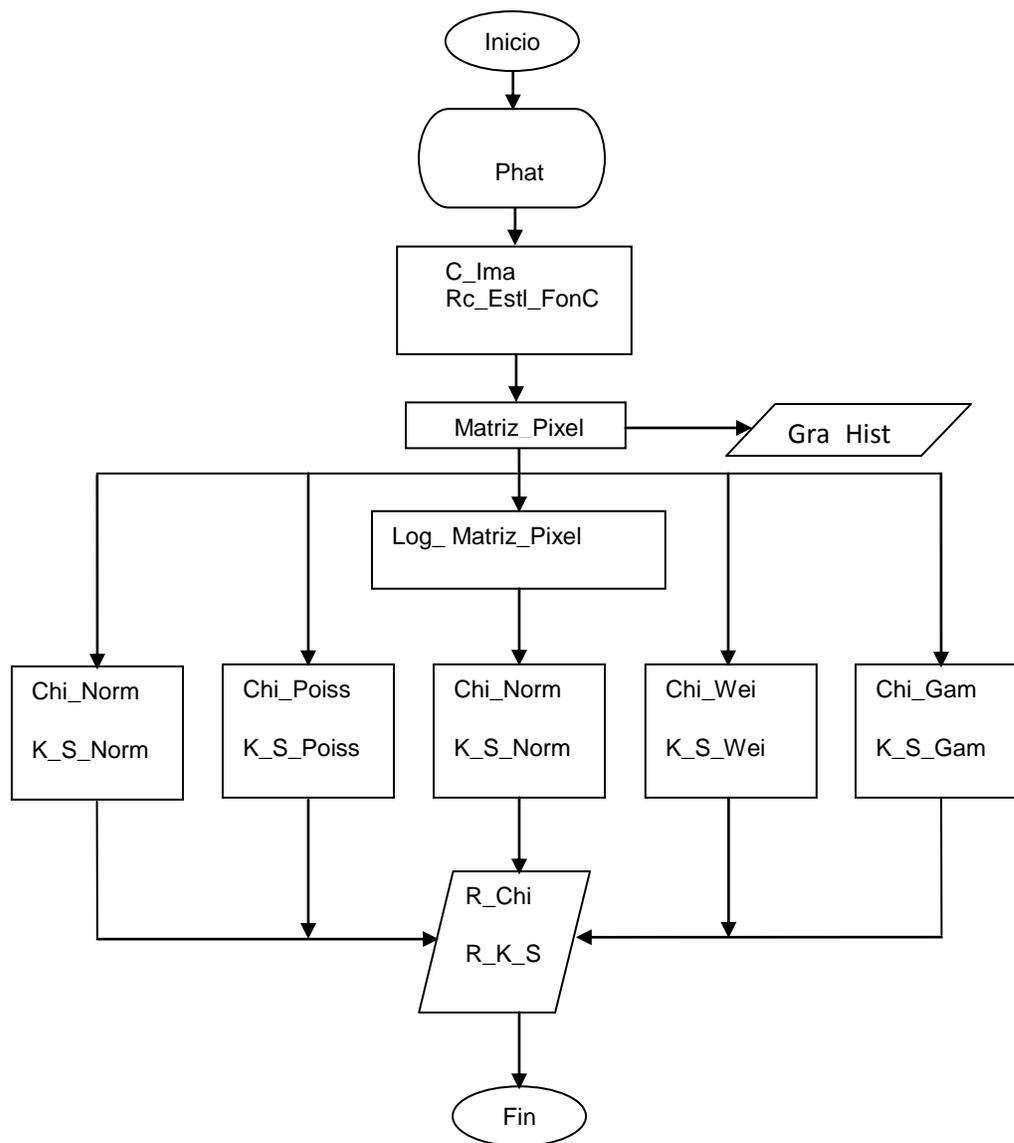


Figura 24. Diagrama flujo de los pasos para identificar la distribución de la estrella.

Donde:

Phat: Dirección de la carpeta donde se encuentran las imágenes.

C_Ima: Carga la imagen.

Rc_Estl_FonC: Genera una sección rectangular de la imagen en la que se puede apreciar el fondo de cielo y la estrella.

Matriz_Pixel: Se construye una matriz con pixeles que superan las 10,000 cuentas.

Gra_Hist: Grafica el histograma.

Log_Matriz_Pixel: Se le toma el logaritmo natural a la matriz de pixeles.

K_S_Norm: Realiza una prueba Kolmogorov-Smirnov para normalidad.

K_S_Gam: Realiza una prueba Kolmogorov-Smirnov para distribución gamma.

K_S_Wei: Realiza una prueba Kolmogorov-Smirnov para distribución weibull.

K_S_Poiss: Realiza una prueba Kolmogorov-Smirnov para una distribución poisson.

Chi_Norm: Realiza una prueba chi-cuadrado para normalidad.

Chi_Gam: Realiza una prueba chi-cuadrado para distribución gamma.

Chi_Wei: Realiza una prueba chi-cuadrado para distribución weibull.

Chi_Poiss: Realiza una prueba chi-cuadrado para distribución de poisson.

R_Chi: Presenta los resultados en una matriz para la prueba chi cuadrado.

R_K_S: Presenta los resultados en una matriz para la prueba Kolmogorov-Smirnof.

Tanto la prueba para el fondo de cielo como para la estrella se realiza con valores de pixel que se considera que solo pueden pertenecer al fondo o a la estrella; Pixeles con valores ambiguos no son tenidos en cuenta.

3.5.3 Distancia entre objetos para determinar las coordenadas de uno de ellos

Ahora bien, el proceso para determinar la distancia entre dos objetos en la misma imagen y con esto hallar las coordenadas de uno de ellos, se describe a continuación.

Para este objetivo se emplearon estrategias de análisis de clústers más no referentes al agrupamiento ya que los pixeles están agrupados y son aquellos que forman las estrellas en la imagen. Lo desarrollado principalmente tiene que ver con el análisis de las distancias entre grupos de conglomerados. En este caso los que hacen parte de las estrellas.

La idea detrás de este objetivo es presentada a continuación:

Al conocer el tamaño en segundos de arco, determinada por la ecuación (42) que puede llegar a tener los pixeles de un sensor *CCD* cualquiera, se puede determinar la cantidad de arco segundos que existen entre un par diferentes de pixeles en la imagen, tanto a la derecha e izquierda como arriba o abajo; basta contar la cantidad de pixeles que hay entre ellos.

En astronomía se emplean, entre otras, un tipo de coordenadas llamadas ecuatoriales que usan dos medidas para encontrar la posición de un cuerpo cualquiera; estas son: Ascensión recta y declinación. La ascensión recta es medida en horas mientras que la declinación en grados.

Es así como si se conoce la coordenada en declinación de un pixel se pueden encontrar las coordenadas de un segundo contando la cantidad de pixeles que los separan y sumando o restando esta distancia en arco segundos al primero; similarmente se puede hacer para la ascensión recta.

Con la anterior idea en mente se propone entonces encontrar las distancias, no entre dos únicos pixeles, sino entre varios objetos de la imagen formados por cantidades diferentes de estos elementos y así determinar las coordenadas de uno de ellos.

Para esto las fotografías son dispuestas de tal forma que a la derecha de estas se encuentre el este y por tanto el lado opuesto corresponderá al oeste, mientras que perpendicular a estos dos puntos se encuentra el eje que va de norte a sur correspondientes al lado superior e inferior respectivamente; tal y como se explicó en las consideraciones para la distancia.

Ya con los puntos cardinales ubicados, se puede notar que al desplazarse a la derecha o a la izquierda de la imagen se realizará un movimiento en ascensión recta; mientras si se hace de arriba a abajo o viceversa, el movimiento se está ejecutando en declinación.

Como se ha mencionado anteriormente, al conocer posición de un objeto se puede determinar la de un segundo únicamente contando la cantidad de pixeles de derecha a izquierda (Ascensión recta) y de norte a sur (Declinación) que los separa, esta cantidad de pixeles es lleva a arco segundos para posteriormente sumárselos o restárselos al clúster que se le conoce la posición. Cabe resaltar que estos arco segundos pueden ser llevados a segundos (temporales) mediante una regla de tres simple.

Con esta idea desarrollada se emplean entonces metodologías de análisis de *Clúster* para determinar la distancia no entre dos pixeles únicamente sino entre un conjunto de ellos. Para ello se emplearon dos métodos: el promedio y el centroide. Ambos programados en Matlab.

3.5.3.1 Método de los promedios

En el método del promedio se programaron las ecuaciones (22), (24), (27) y (29) que determinan las distancias existentes entre varios conglomerados que forman diferentes estrellas a un conglomerado en particular de manera que con dichas distancias se puedan hallar las coordenadas de este último. Para ello se corre el código promedios (Anexo 3.1) que luego de cargar la imagen, ingresar la cantidad de objetos a seleccionar, las coordenadas en ascensión recta, en declinación y los segundos de arco por pixel. Procede a orientar la imagen dejándola en condiciones de realizar la prueba. El algoritmo es ejecutado teniendo en cuenta que el primer objeto a elegir a de ser aquel al que se le desconocen sus coordenadas.

Al escoger los objetos una matriz con las coordenadas de los pixeles de estos es creada, pero solo aquellos pixeles con cuentas superiores al 70% del valor máximo de dicha selección son tenidos en cuenta.

Posteriormente las coordenadas ingresadas en ascensión recta y declinación, son transformadas a arco segundos con el fin de manejar las mismas unidades; paso seguido, se calculan todas las distancias tanto horizontales como verticales. Como se pueden seleccionar varios objetos, solo se permite que uno de ellos tenga coordenadas desconocidas, así que el algoritmo halla siempre las distancias entre el primer objeto (de coordenadas desconocidas) y un segundo, luego entre el primero y el tercero, primero y cuarto; así sucesivamente hasta que se abarquen

todos los objetos escogidos; las distancias son estimadas desde cada uno de los píxeles del primer objeto a cada uno de los píxeles del enésimo objeto.

Luego estas distancias son promediadas y se les calcula la desviación para cada par. Paso seguido, estas magnitudes son entonces sumadas o restadas de cada uno de los objetos de referencia teniendo en cuenta de que si tal objeto ocupa una posición más al norte de la fotografía que el objeto desconocido la distancia es restada, similarmente se realiza para la ascensión recta mediante la función *posicion_Promedio* (Anexo 3.2), cabe aclarar que esta comparación solo se hará con el píxel más al norte y más al este de cada grupo.

Una vez terminado este proceso la función *Conversor1* (Anexo 3.3) transforma los arcos segundos obtenidos para la declinación y la ascensión recta en grados y horas respectivamente. De esta forma se hallan las coordenadas del objeto desconocido. Estos resultados serán mostrados en una matriz con tantas coordenadas como objetos seleccionados menos uno; así mismo, una matriz con las desviaciones de estas medidas es mostrada. El diagrama de flujo es mostrado en la Figura 25.

3.5.3.2 Método del centroide

Para el método del centroide las distancias no son calculadas píxel a píxel, por el contrario, las coordenadas tanto en *x* como en *y* de cada uno de los píxeles que forman cada uno de los cuerpos son promediados obteniendo así un único par de coordenadas como lo muestran las ecuaciones (31) y (32), a partir de estos pares es desde donde se determinan las distancias. El código de este método es similar en gran parte al anterior.

En las líneas llamadas *Centroide* (Anexo 4.1), se ingresan el número de objetos, las matrices con las coordenadas en ascensión recta y declinación, la cantidad de segundos de arco por píxel. Paso seguido se carga y muestra la imagen en la posición correcta. Posteriormente los objetos son seleccionados, teniendo en cuenta que el primer objeto a seleccionar a de ser el que se le desconocen las coordenadas. Luego, una matriz con las coordenadas de dichos objetos es creada y de esta manera determinar las coordenadas promedio de cada uno de los cuerpos escogidos; calculando igualmente, su desviación.

Ya con las coordenadas de estos centroides se procede a calcular las distancias mediante la función *posicion_centro* (Anexo 4.2). Estas distancias son sumadas o restadas tal y como se ha venido planteando. Por último, las coordenadas en arco segundos son entonces llevadas nuevamente a grados para la declinación y a horas para la ascensión recta mediante la función *Conversor* (Anexo 4.3). La

Figura 26 muestra este proceso. Para dar un estimativo de la incertidumbre en la medición, las desviaciones de cada par de centroides son sumadas.

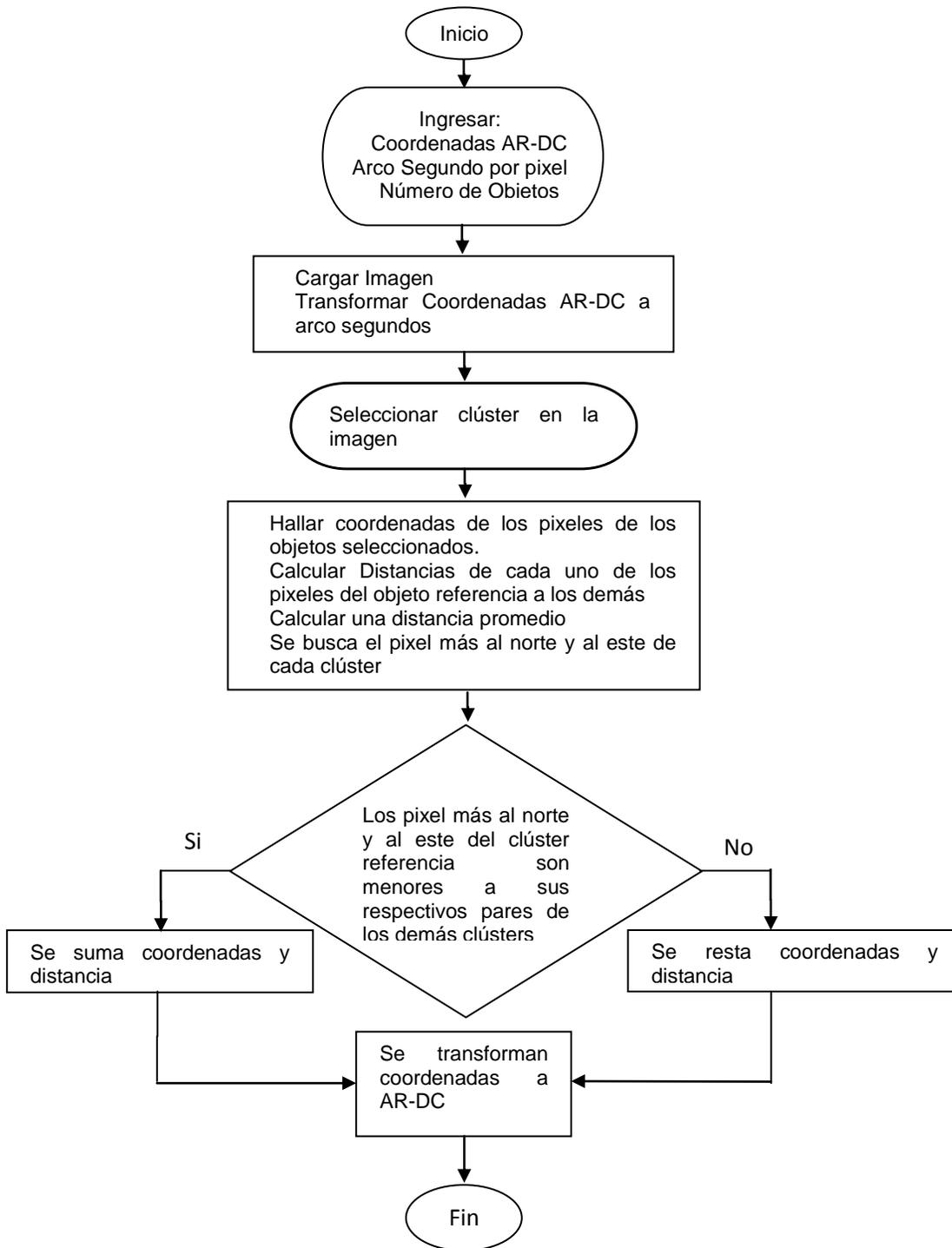


Figura 25. Diagrama de flujo método de los promedios.

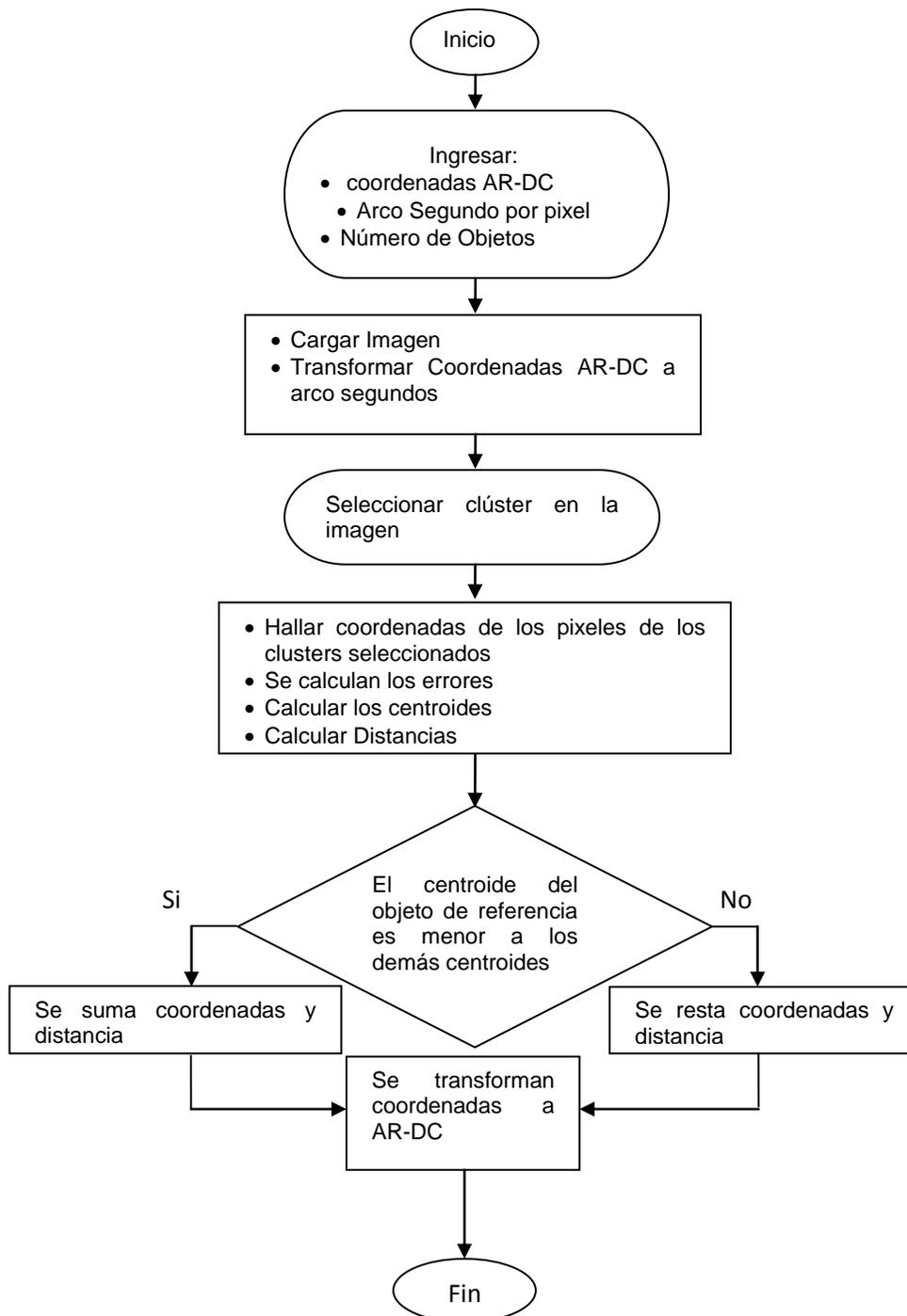


Figura 26. Diagrama de flujo. Método del centroide.

4. RESULTADOS

El esquema de la Figura 27 sintetiza los métodos a llevar a cabo para el cumplimiento de los objetivos planteados.

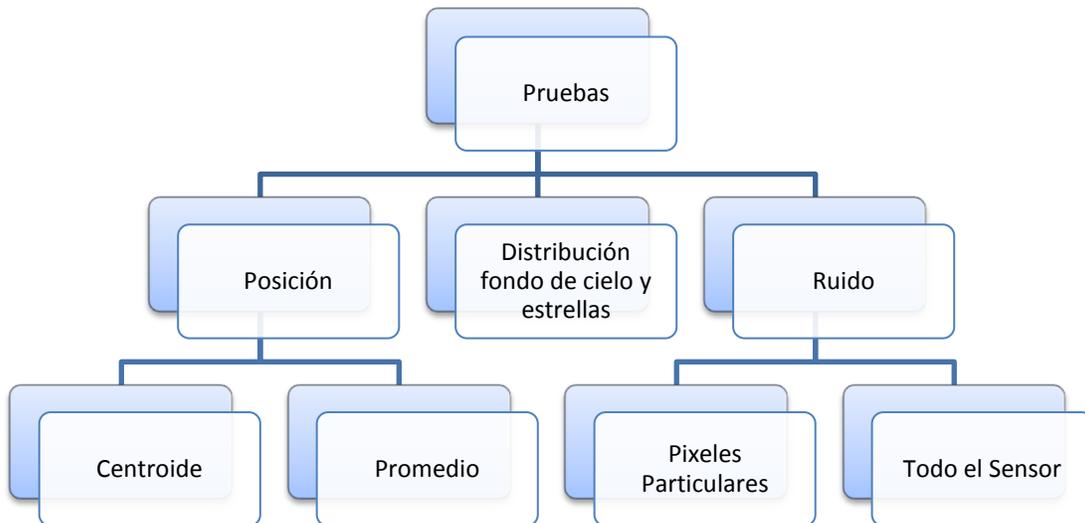


Figura 27. Síntesis de las pruebas a realizar.

Como bien muestra el esquema anterior y para el caso de la corriente de oscuridad se emplea dos metodologías. La primera de ellas busca mediante un muestreo aleatorio simple a cada una de las 60 imágenes, de cada una de las tres jornadas, construir un vector de datos. Ha dicho vector se le realizará las pruebas de bondad y ajuste.

Por otro lado, se le realizan 60 mediciones a ciertos píxeles seleccionados aleatoriamente, es decir, medir las cuentas que registran algunos píxeles en cada una de las 60 exposiciones para cada jornada y determinar si se ajustan o no a alguna de las distribuciones anteriormente expuestas.

Entre tanto y para determinar las coordenadas en ascensión recta y declinación de un objeto cualquiera en una imagen a partir de otros; se desarrollaron los métodos del centroide y del promedio ponderado.

Por último se realizan pruebas a los píxeles que se tiene una certeza de su pertenencia tanto a la estrella como al fondo de cielo e intentar identificar así una posible distribución subyacente a estas cifras.

4.1 RESULTADOS DEL POSIBLE AJUSTE A UNA DISTRIBUCIÓN DE LA CORRIENTE DE OSCURIDAD

Teniendo en cuenta que fueron tres jornadas de captura de *dark*s, se mostrarán las graficas obtenidas al procesar dicha información, correspondientes al cálculo de los estadísticos *Chi-cuadrado* y *Kolmogorov-Smirnov*.

En los esquemas siguientes los estadísticos críticos son mostrados como líneas o como asteriscos; estos han de ser mayores a los estadísticos calculados, mostrados generalmente como barras, para que las hipótesis sean aceptadas a un nivel de significancia del 95%.

4.1.1 Resultados para la distribución de todo el sensor

4.1.1.1 Resultados para la primera jornada

En la Figura 28 se muestra como los valores para los estadísticos calculados, representados por las barras, son mayores al valor crítico para el test Chi cuadrado, representado por la flecha roja.

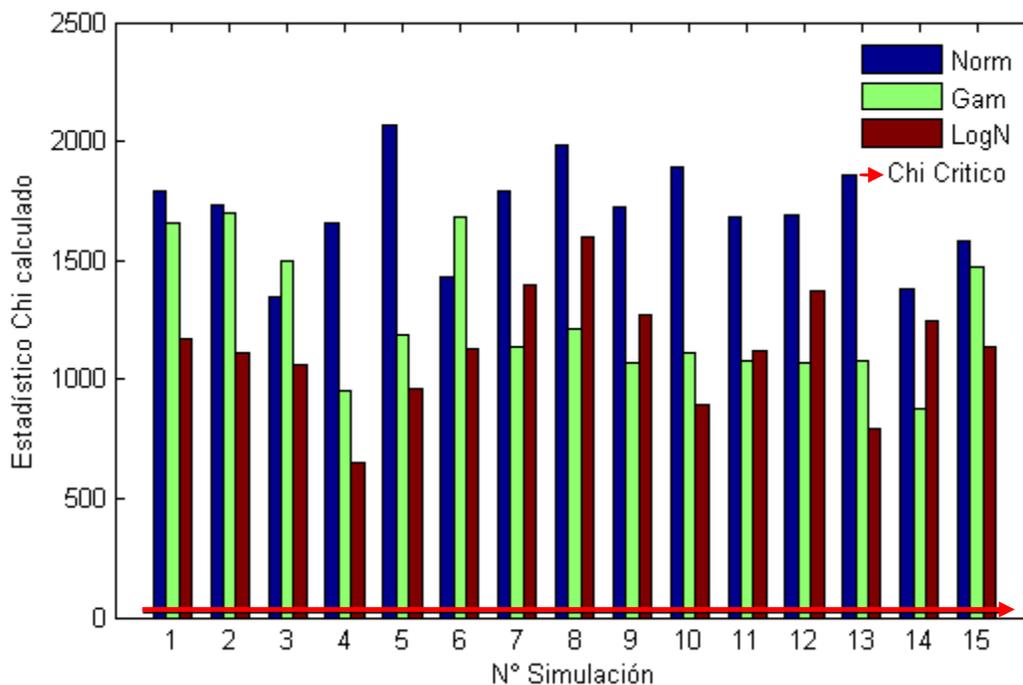


Figura 28. Comparación entre el estadístico calculado y el crítico para la prueba Chi primera jornada.

En Figura 29 se aprecian los resultados para la prueba Kolmogorov-Smirnov, esta respalda los resultados obtenidos mediante el test Chi cuadrado, ya que puede notarse como los estadísticos calculados son superlativamente mayores al crítico.

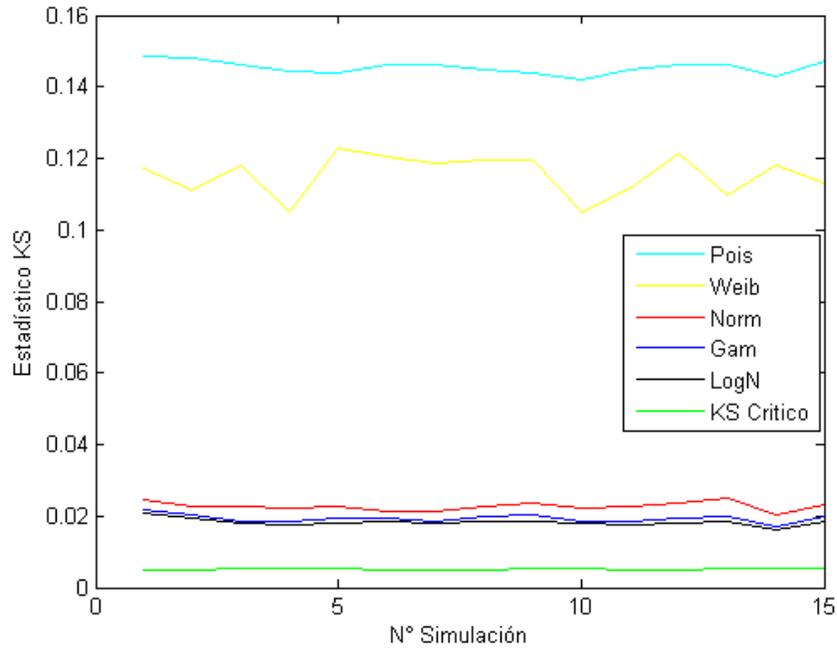


Figura 29. Comparación del estadístico calculado y el crítico para la prueba KS primera jornada.

4.1.1.2 Para la segunda jornada

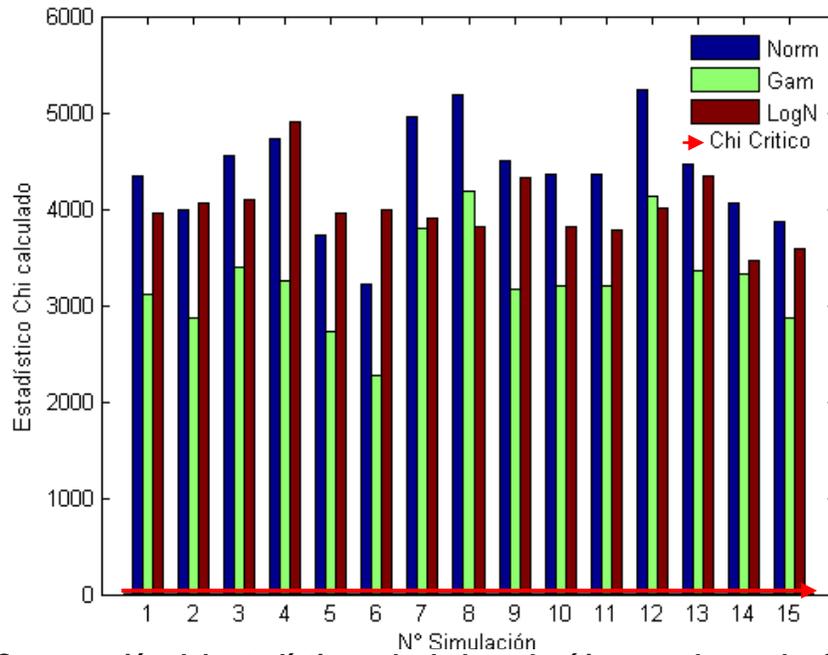


Figura 30. Comparación del estadístico calculado y el crítico para la prueba Chi segunda jornada.

Los resultados de las pruebas Chi y K-S para la segunda jornada no son muy diferentes de los obtenidos en la primera; en ambos test el estadístico crítico es significativamente menor a los estimados, como bien puede ser apreciado en las Figuras 30 y 31.

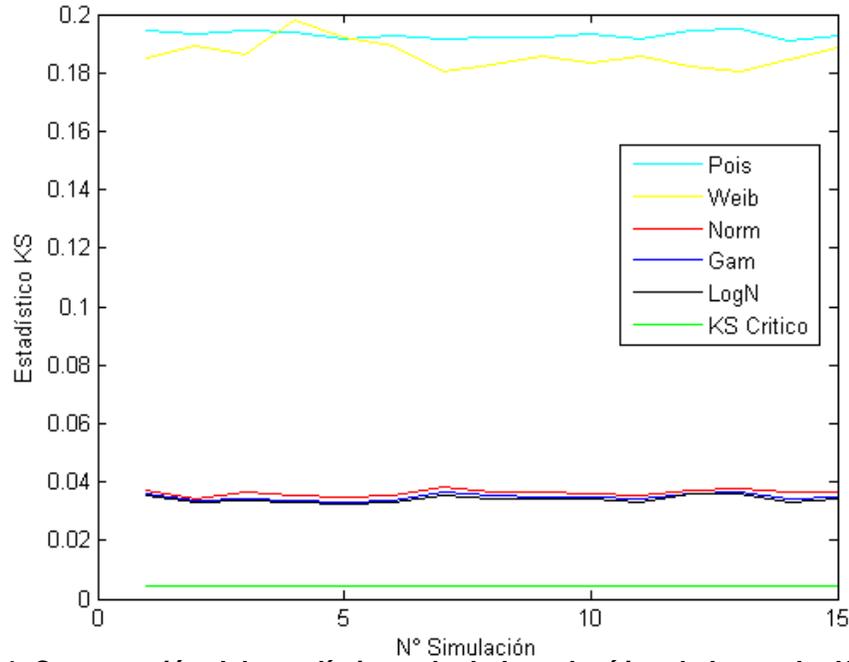


Figura 31. Comparación del estadístico calculado y el crítico de la prueba KS para la segunda jornada.

También puede notarse como los valores de los estadísticos calculados son mayores en la segunda jornada que en la primera.

4.1.1.3 Para la tercera jornada

Para la tercera jornada, y como bien lo ilustran las Figuras 32 y 33, los resultados son semejantes a las jornadas anteriores; los estadísticos críticos son menores a los estimados para ambos contrastes.

En las Figuras 28, 30 y 32 no se grafican los valores calculados para las distribuciones Poisson y Weibull, debido a que presentan valores tan grandes que no facilitan la comparación a escala con los valores obtenidos para la Gamma, la LogNormal y la Normal; como bien puede apreciarse en los Anexos 7.1, 7.4 y 7.7.

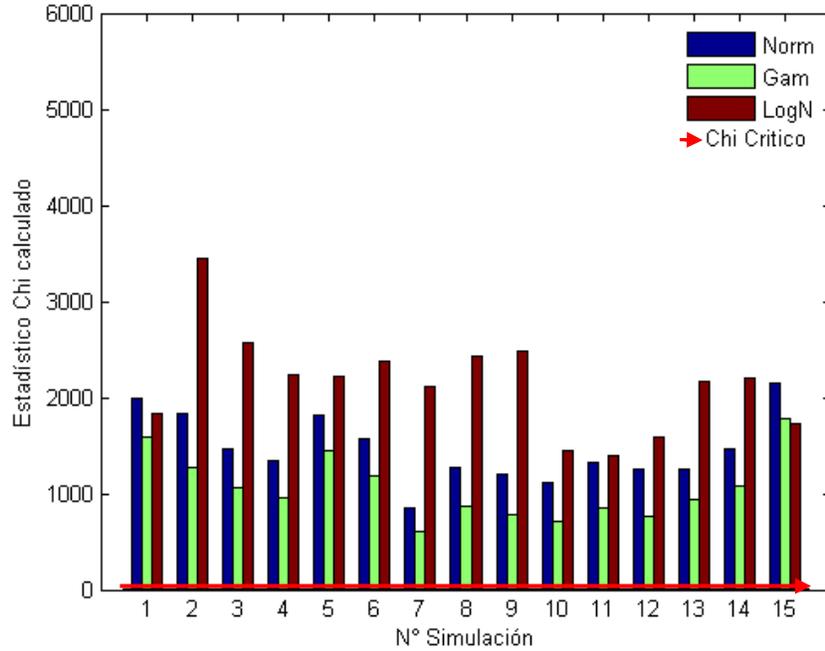


Figura 32. Comparación del estadístico calculado y el crítico de la prueba Chi para la tercera jornada.

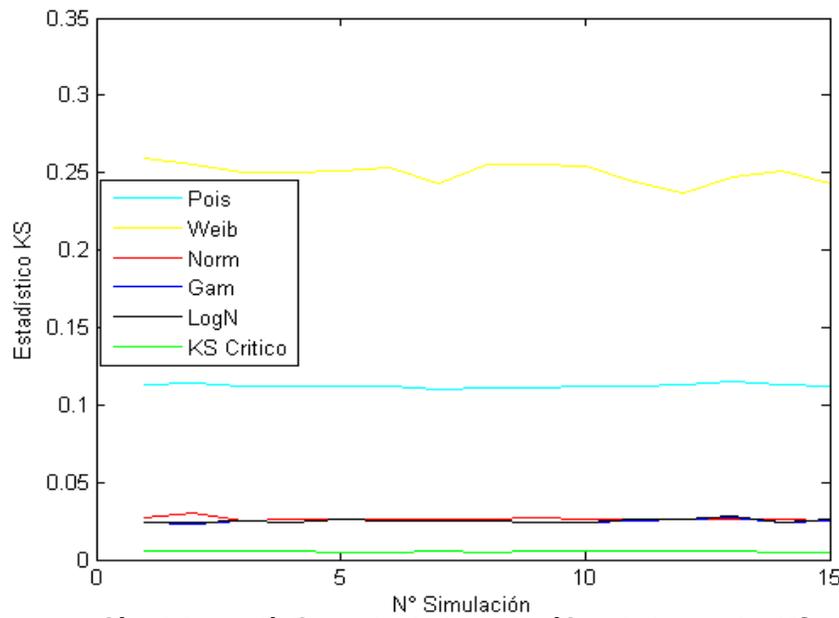


Figura 33. Comparación del estadístico calculado y el crítico de la prueba KS para la tercera jornada.

Se puede apreciar como los mayores valores son generados para las funciones Weibull y Poisson; mientras que la gamma, la lognormal y la normal presentan comportamientos similares en el test *KS* (para las tres jornadas).

4.1.2 Pruebas para los pixeles particulares

Por otro lado se propone realizar las pruebas a un número determinado de pixeles, en este caso 10, escogidos aleatoriamente como bien lo muestra la Tabla 3, en la pagina 48, Posteriormente se tomaron los valores registrados por cada píxel en las 60 fotografías, para finalmente estudiar el comportamiento de estos datos.

Esta prueba fue realizada en vista de que el test presentado en el numeral anterior registra valores de Chi Crítico y KS considerablemente inferiores a los estadísticos estimados de las distribuciones, lo cual imposibilita la determinación de la distribución de probabilidad que mejor se ajusta a los datos tomados.

Para el siguiente análisis cabe aclarar que los asteriscos y la línea continua representan el *Chi* y *K-S* críticos respectivamente, mientras que las barras representan los estadísticos calculados.

4.1.2.1 Resultados de la primera jornada

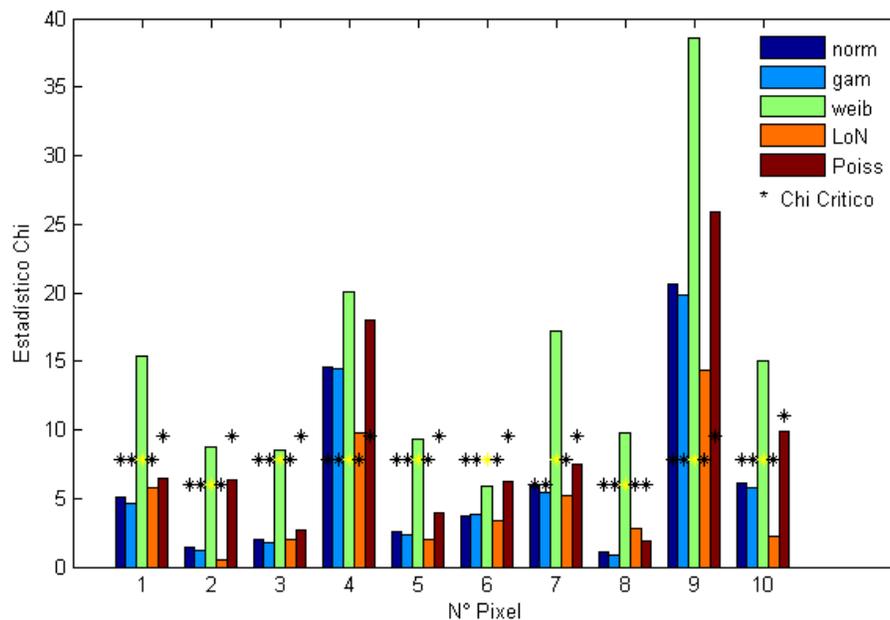


Figura 34. Comparación del estadístico calculado y el crítico de la prueba Chi para pixeles particulares en la primera jornada (la Tabla 3 indica la posición de cada píxel).

En las Figuras 34 y 35 se pone de manifiesto los resultados obtenidos para la primera jornada. En la primera de ellas, Figura 33, los estadísticos calculados son en su mayoría menores al crítico con excepción del estimado para la distribución weibull que presenta un valor mayor en todos los pixeles.

Al analizar la Figura 35 se observa que la gran mayoría de los estadísticos hallados son inferiores al crítico para la prueba *K-S*.

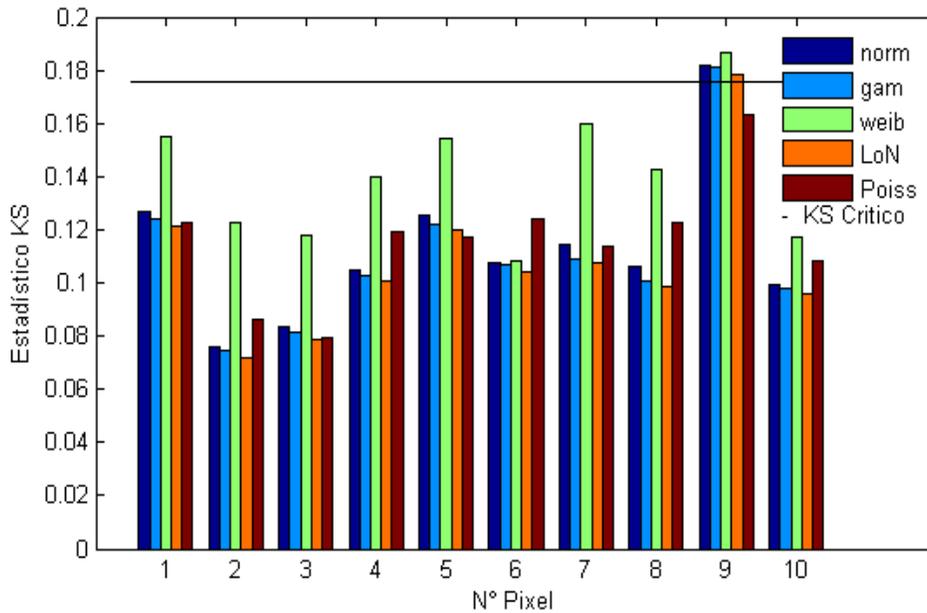


Figura 35. Comparación del estadístico calculado y el crítico de la prueba *K-S* para pixeles particulares en la primera jornada.

4.1.2.2 Para los pixeles en particular segunda jornada

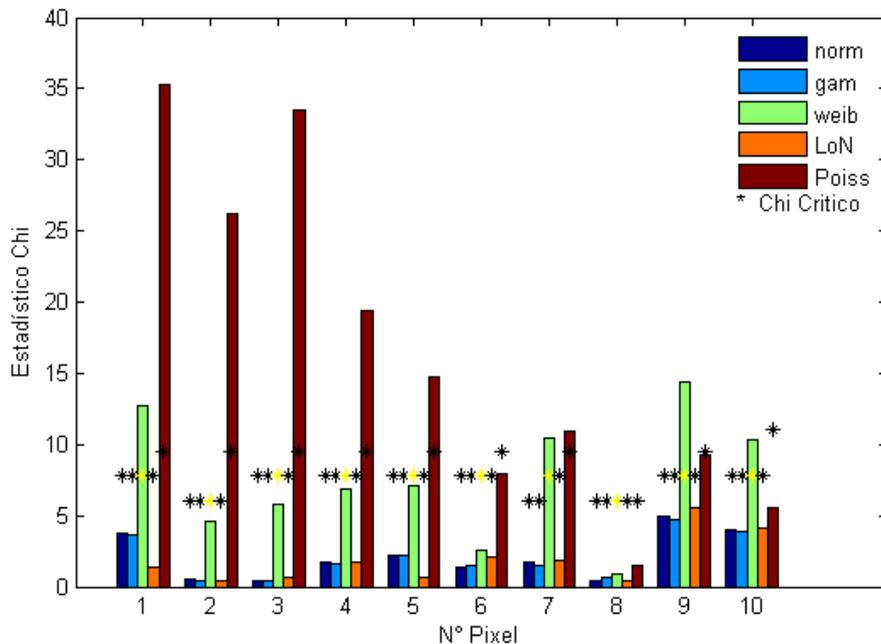


Figura 36. Comparación del estadístico calculado y el crítico de la prueba *Chi* para pixeles particulares en la segunda jornada.

Para la segunda jornada y al observar las Figuras 36 y 37, el patrón de comportamiento es similar al de primera, notándose como los estadísticos estimados para las distribuciones gamma, normal y lognormal son, en su mayoría, menores al crítico.

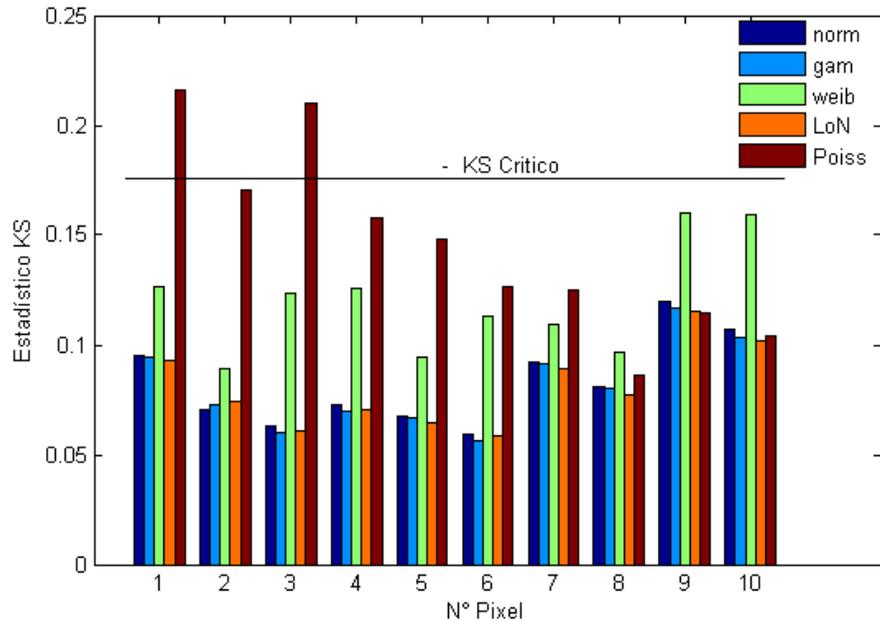


Figura 37. Comparación del estadístico calculado y el crítico de la prueba KS para pixeles particulares en la segunda jornada.

4.1.2.3 Para los pixeles en particular tercera jornada

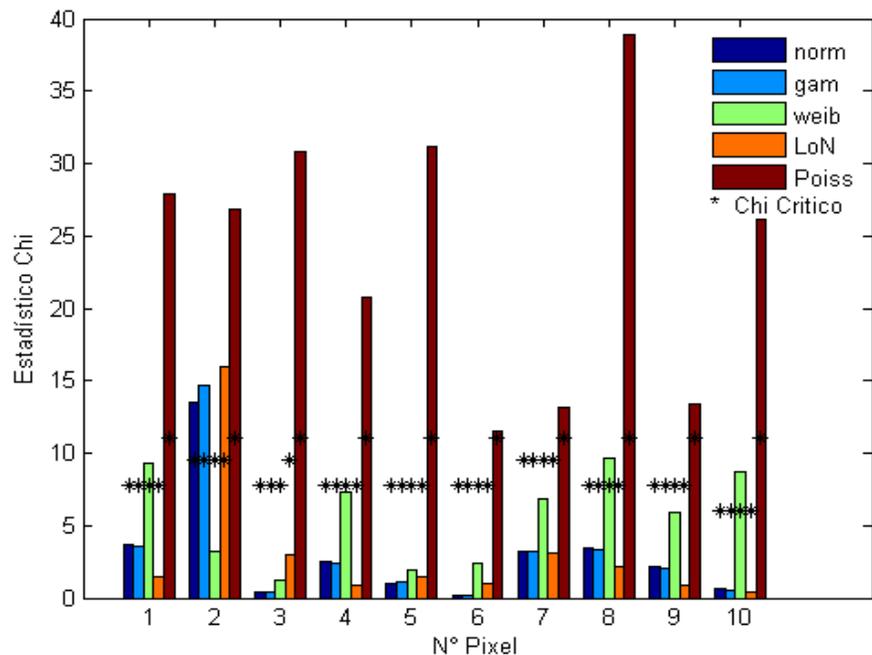


Figura 38. Comparación del estadístico calculado y el crítico de la prueba Chi para pixeles particulares en la tercera jornada.

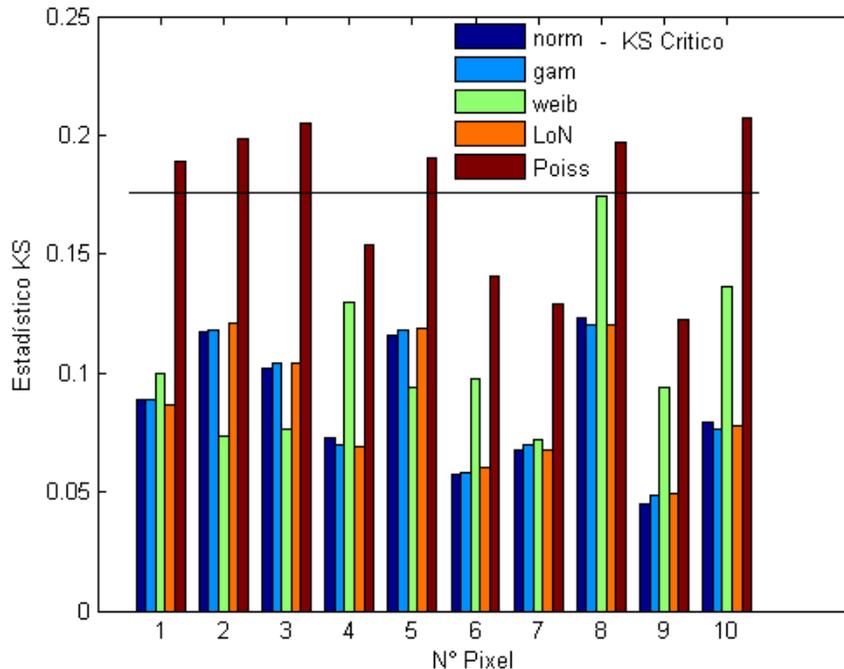


Figura 39. Comparación del estadístico calculado y el crítico de la prueba KS para pixeles particulares en la tercera jornada.

En las Figuras 38 y 39, correspondientes a la tercera y última jornada, se nota nuevamente como los estadísticos estimados son menores a los críticos; con algunas excepciones como el estadístico determinado para la distribución de poisson.

Se identifica así mismo, como el test *Chi-Cuadrado* tiende a rechazar más hipótesis nulas que el test *Kolmogorov-Smirnov* y aunque ambos no convergen a la misma respuesta para un pixel cualquiera. Un ejemplo de ello se presenta en el primer pixel de las figuras 34 y 35, donde la hipótesis que se acepta en ambos casos no es la misma; ya que para la primera se escogería la gamma mientras que para la segunda sería la lognormal. Ambos test tienden a presentar valores menores para la gamma, la lognormal y normal.

Por otro lado, al analizar detalladamente las figuras 34 a 39, se observa que los pixeles no se ajustan a la misma función de distribución para las tres jornadas. Por el contrario, tanto para la prueba *KS* como la *Chi* se pueden encontrar valores diferentes.

Los datos bajo los cuales se construyeron las anteriores figuras se muestran en los anexos 8.1 al 8.9.

4.2. RESULTADOS PARA EL FONDO DEL CIELO

4.2.1 Distribución para el fondo de cielo. Prueba *Chi*

Para la distribución del fondo de cielo se emplearon las mismas estrategias que se han venido planteando.

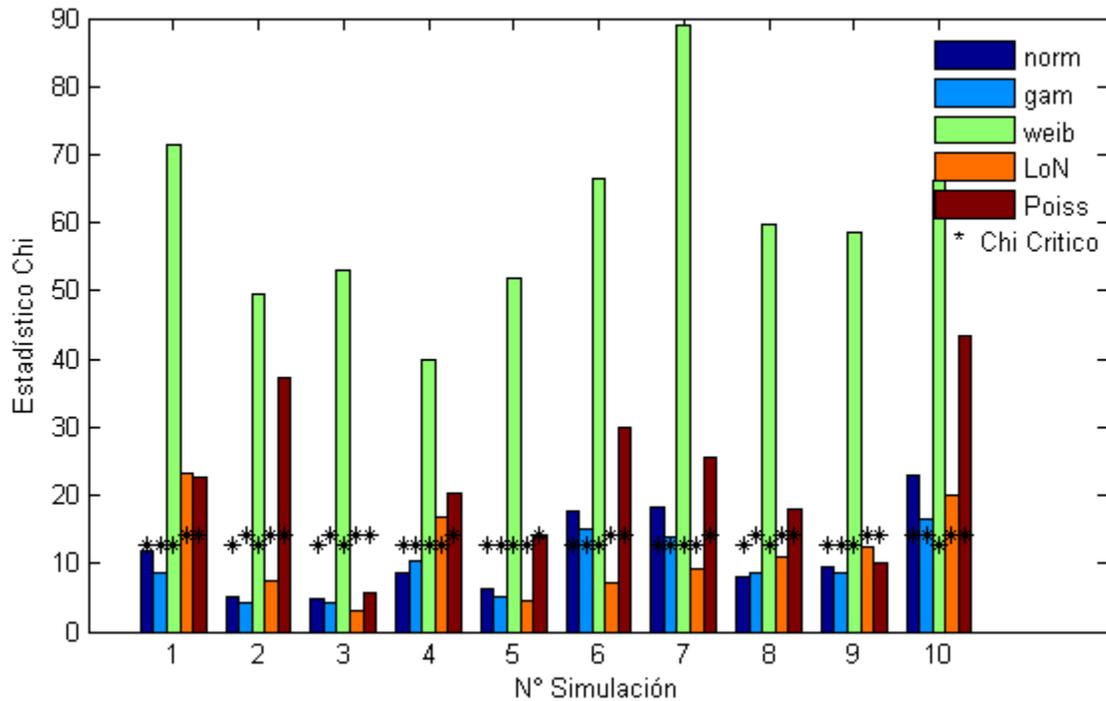


Figura 40. Comparación del estadístico calculado y el crítico de la prueba Chi para el fondo de cielo.

Para identificar el mejor ajuste se realizaron diez simulaciones. Los datos obtenidos son mostrados en el Anexo 10.1.

Como bien se aprecia en las Figuras 40 y 41, el estadístico calculado para la distribución weibull es mayor que el crítico para ambos test.

Además también cabe notar que cada vez que se hace una simulación los valores calculados son diferentes para los dos estadísticos, aunque en ambas se aprecia que la lognormal y la gamma son las que presentan datos menores.

4.2.2 Resultado Prueba K_S para fondo de cielo

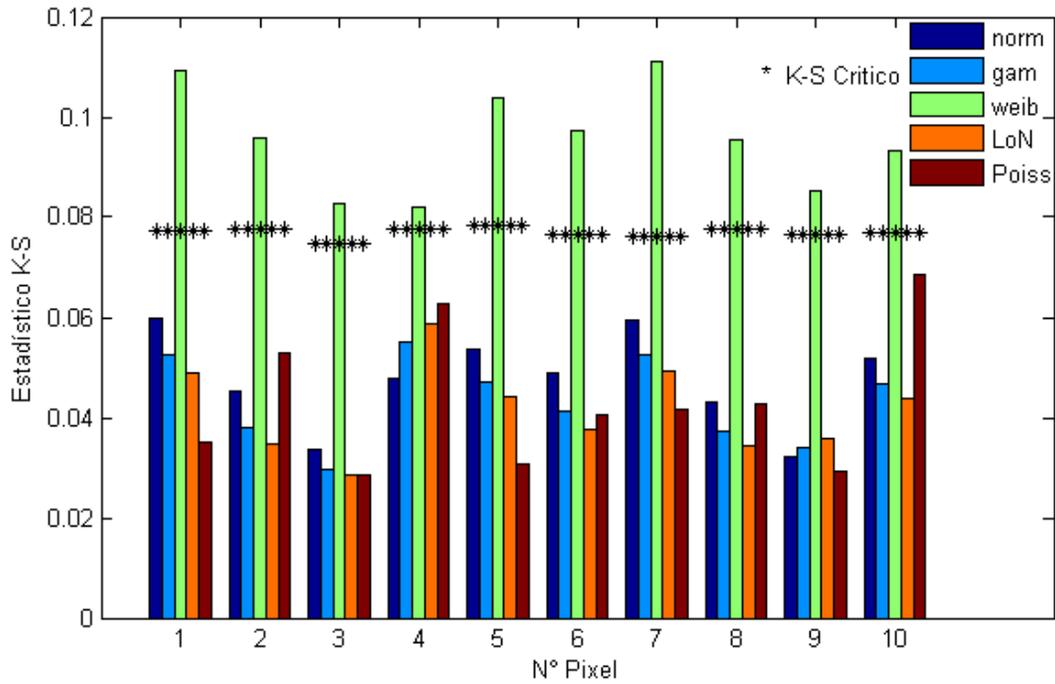


Figura 41. Comparación del estadístico calculado y el crítico de la prueba KS para el fondo de cielo.

4.3 RESULTADOS PARA LA DISTRIBUCIÓN DE LA ESTRELLA

Como bien se aprecia en las Figuras 42 y 43 los estadísticos calculados son mucho mayores que los críticos para ambas pruebas.

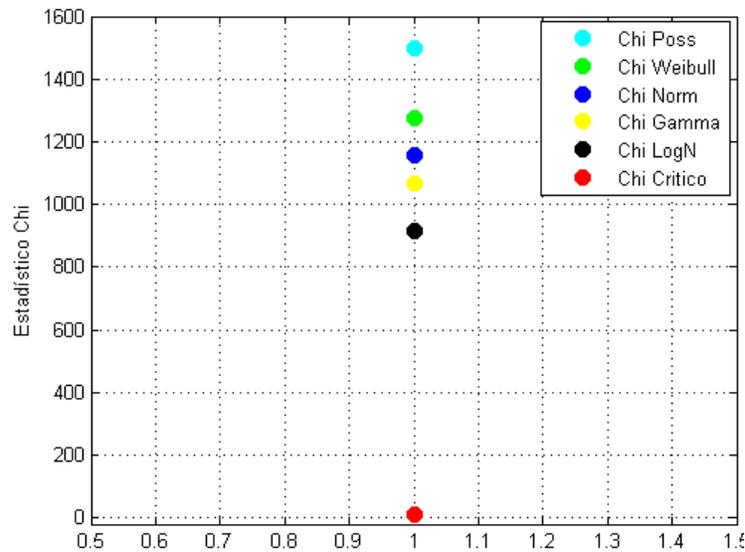


Figura 42. Comparación del estadístico calculado y el crítico de la prueba Chi para la estrella.

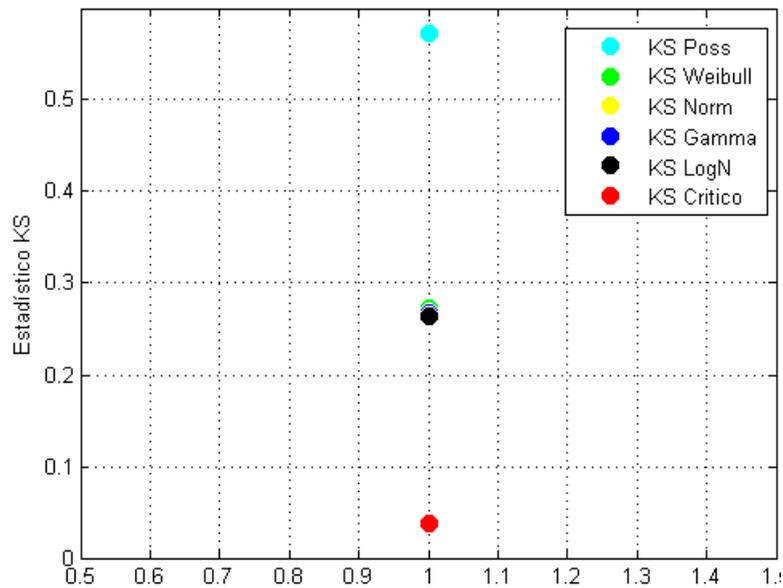


Figura 43. Comparación del estadístico calculado y el crítico de la prueba KS para la estrella.

4.4 CÁLCULO DE LAS COORDENADAS POR AMBOS MÉTODOS

Para esta prueba y con el fin de evaluar los dos métodos, los resultados son comparados respecto a un cuerpo de coordenadas conocidas las cuales fueron tomadas de *SIMBAD*⁷.

En la Figura 44 se observan las 5 estrellas escogidas para los estudios. Las coordenadas del objeto 1 son determinadas a partir de las coordenadas de los objetos 2, 3, 4 y 5.

La Figura 45 muestra puntos con colores diferentes. El punto negro representa las coordenadas reales del objeto de estudio (estrella 1). Los puntos rojos identifican las coordenadas halladas del punto negro a partir de las estrellas de la Figura 40 mediante el método del promedio, y están identificados por la letra *P* seguida por un número que indica la estrella a partir de la cual se determinó la posición. Los puntos azules señalan las coordenadas del punto negro estimadas a partir de las mismas cuatro estrellas anteriores, pero ahora por el procedimiento del centroide y son identificadas análogamente con la letra *C* y un número para identificar la estrella a partir de la cual se obtuvo la ubicación.

⁷ SIMBAD es una base de datos astronómica que brinda información de diferentes objetos celestes, medidas y datos básicos de cuerpos más allá del sistema solar.

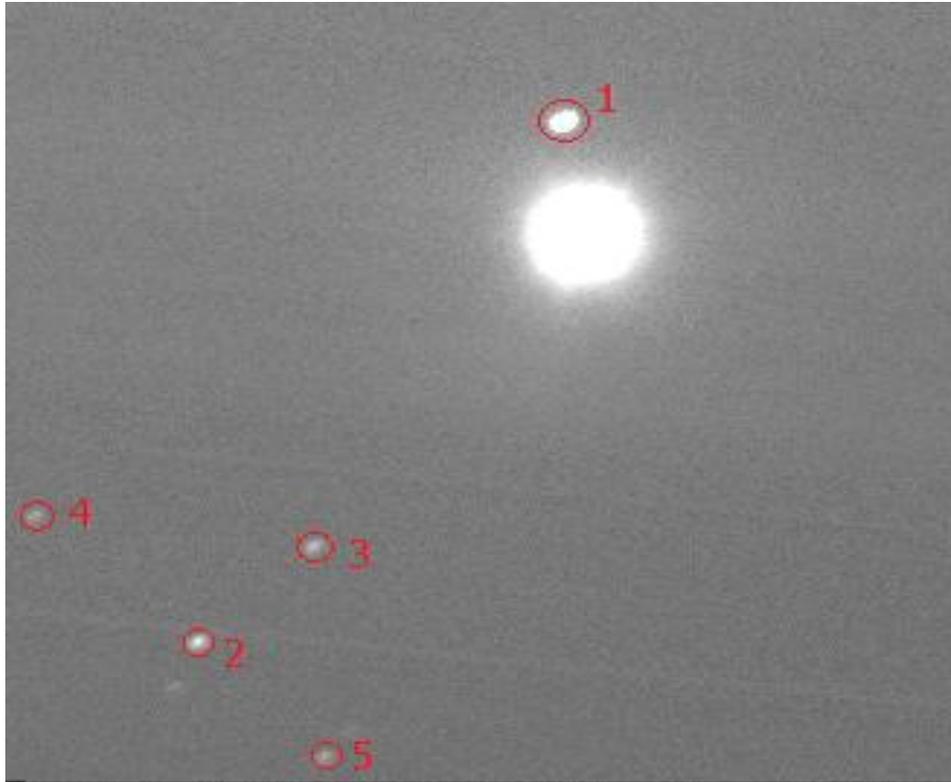


Figura 44. Objetos seleccionados para realizar las pruebas.

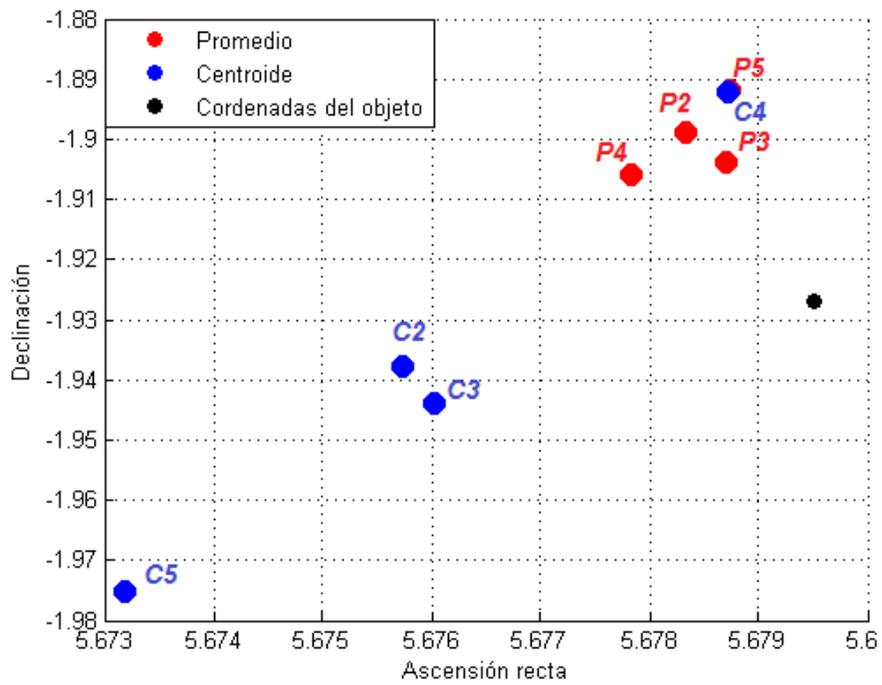


Figura 45. Coordenadas obtenidas con ambos métodos.

De las Tablas 4 a la 7 se aprecian los datos correspondientes a los errores en la medición tanto en la declinación como en la ascensión recata; por ambos métodos.

	Estrella 2	Estrella 3	Estrella 4	Estrella 5
Estrella 1	0.02792389	0.02311221	0.02114147	0.03449751

Tabla 4. Errores declinación (grados); método del promedio.

	Estrella 2	Estrella 3	Estrella 4	Estrella 5
Estrella 1	0.00117343	0.00079323	0.00166772	0.00078469

Tabla 5. Errores en ascensión recta (horas); método del promedio.

	Estrella 2	Estrella 3	Estrella 4	Estrella 5
Estrella 1	0.01090612	0.01710613	0.03453791	0.04850427

Tabla 6. Errores declinación (grados); método del centroide.

	Estrella 2	Estrella 3	Estrella 4	Estrella 5
Estrella 1	0.00376394	0.00348029	0.00077678	0.00631777

Tabla 7. Errores en ascensión recta (horas); método del centroide.

4.4.1 Resultado dispersión para las declinaciones

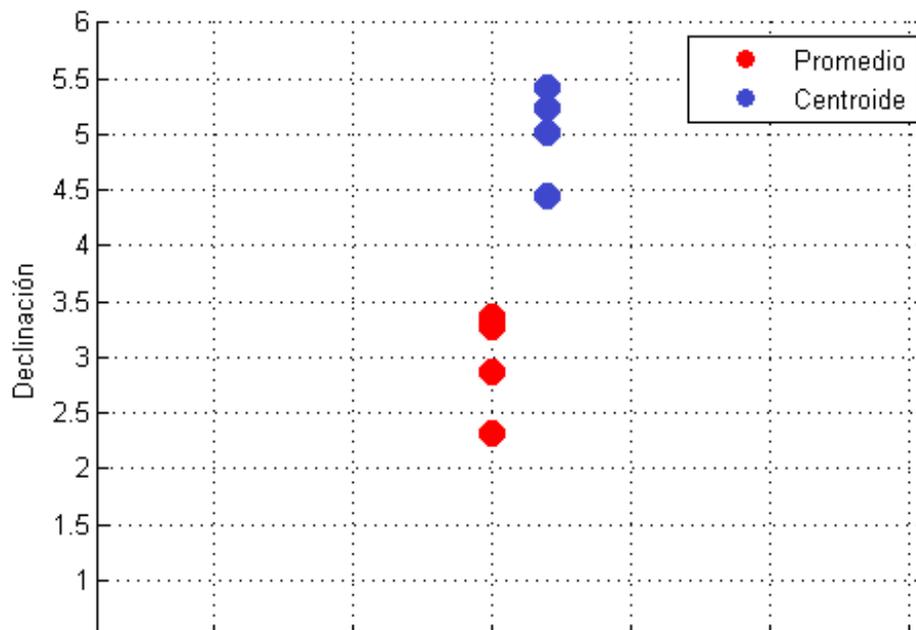


Figura 46. Diagrama dispersión para la declinación.

Como bien se aprecia en las Figuras 46 y 47, la dispersión para la ascensión recta y para la declinación presenta valores menores para la metodología del promedio

ponderado. Cabe aclarar que cada uno de los puntos es la desviación estimada entre el objeto al que se le desconocen las coordenadas y las estrellas conocidas.

4.4.2 Resultado dispersión para las Ascensión recta

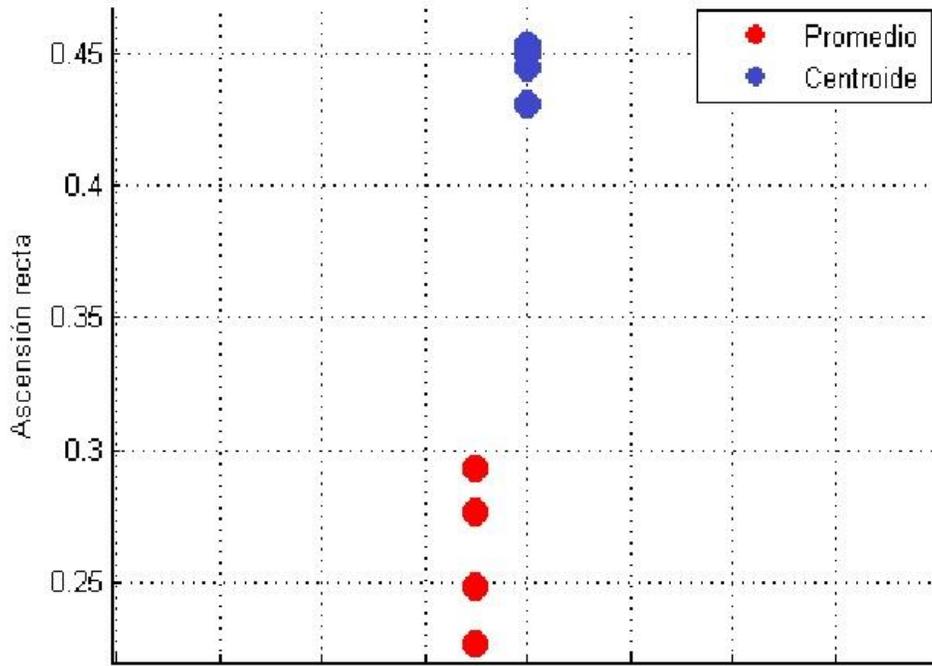


Figura 47. Diagrama dispersión para la ascensión recta.

CONCLUSIONES

Al tomar muestras diferentes en cada una de las 60 imágenes para las tres jornadas, se aprecia que los estadísticos calculados para cada una de las pruebas son superlativamente mayores a los críticos, llevando a rechazar todas y cada una de las hipótesis nulas. Siendo imposible ajustar los datos mediante esta metodología de recolección de información.

Entre tanto, al medir las cuentas registradas por un mismo pixel en las 60 exposiciones para las tres jornadas, se aprecia como los estadísticos estimados son significativamente menores a los críticos, lo que permite aceptar diferentes hipótesis nulas (donde la lognormal y la gamma son las que más se repiten) para los pixeles a un nivel de significancia del 95%.

De acuerdo a lo desarrollado y concluido se infiere entonces que la corriente de oscuridad no presenta un comportamiento normal como es asumida en la mayoría de las investigaciones, en la literatura y en el estado del arte.

Al intentar identificar la función de distribución del fondo de cielo y de la estrella, se nota como con un nivel de significancia del 95% son aceptadas varias de las hipótesis nulas, siendo las que más se repiten la lognormal y la gamma. No ocurriendo lo mismo para la estrella. Para esta y al tomar los datos considerados como del astro, se observa como los estadísticos de prueba son inferiores a los hallados. Llevando a rechazar las hipótesis nulas.

Esto no impide la clasificación de los pixeles ya que se propone que con la distribución obtenida del fondo de cielo, se pueden evaluar los valores de los pixeles ambiguos en tal función y de acuerdo a la probabilidad determinada realizar la clasificación.

Al observar los resultados de las distancias, en las que se implementaron los métodos del centroide y del promedio este último presenta resultados más favorables tanto en posición como en dispersión. Esto, debido a que el método del promedio encuentra las distancias de todos los elementos que forman ambos cuerpos mientras que el centroide encuentra un centro común para cada objeto y de allí mide sus distancias.

RECOMENDACIONES

Apoyar las pruebas de bondad y ajuste estudiadas con el cálculo de los P-Valores para una mayor certeza en las respuestas obtenidas.

Implementar estrategias mas robustas como el de análisis de discriminantes o técnicas de agrupamiento para discernir con mayor grado de precisión que pixel es estrella o no.

Determinar la distribución que mejor se ajusta a cada uno de los pixeles y de esta forma estimar el valor esperado para cada uno de ellos y así realizar una extracción de la corriente de oscuridad con un mayor soporte estadístico.

En caso de obtener resultados en los que la función de distribución que mejor se ajusta a los datos consta de parámetros a los que difícilmente se les puede dar una interpretación física respecto al fenómeno de estudio, se recomienda hacer uso de una distribución cuyos parámetros si puedan encontrarse una explicación e interpretación.

Crear una base de datos de darks y con ello obtener mejores estimaciones en las pruebas.

Realizar una optimización del código y si es necesario programarlo en paralelo para mejorar los tiempos de procesado de los datos.

Realizar un análisis multivariado que investigue el impacto de la temperatura y del tiempo exposición respecto a las cuenteas registradas con el obturador cerrado.

BIBLIOGRAFÍA

- [1] MOHSEN, AMR M; TOMPSETT, Michael F y SEQUIN, Carlo H. Noise Measurements in Charge Couple Devices. IEEE Transactions On Electron Devices. Vol 22, N° 5. Mayo de 1975.
- [2] HEALEY, Glen E y KONDEPUDY, Raghava. Radiometric CCD Camera Calibration and Noise Estimation. IEEE Transactions On Pattern Analysis and Machine Intelligence. Vol 16, N° 3. Marzo 1994.
- [3] FARAJI, Hilda y MACLEAN, W. James. CCD Noise Removal In Digital Images. IEEE Transactions On Image Processing. Vol 15, N° 9. Septiembre 2006
- [4] WU, Jingjing; ZHANG, Zhenchuan y TIAN, Yong. A Method of CCD Noise removal in Digital Images. Second IEEE Conference on Industrial Electronics and Applications. 2007
- [5] LIU, Ce; SZELISKI, Richard; KANG, Sing Bing; ZITNICK, C. Lawrence y FREEMAN, William T. Automatic Estimation and Removal of Noise from a Single Image. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 30, N° 2. Febrero 2008
- [6] SANKOWSKI, Dominik y FABIJANSKA Anna. CCD Camera Instrumental Background Estimation Algorithm. Instrumentation and Measurement. Technology Conference. Warsaw, Polonia. Mayo 2007.
- [7] LEE, Shinhak. Point Accuracy Improvement Using Model Based Noise Reduction Method. Jet Propulsion Laboratory, California Institute of Technology.
- [8] HALLADO ARENALES, Belén. Búsqueda de Variables RR Lyrae en el Cúmulo Globular M 13. Cantabria, 2012, 46 p. Tesis (Máster en Física y Tecnologías Físicas). Universidad de Cantabria. Facultad de Ciencias.
- [9] DOMINGO GARAU, Albert. Fotometría Óptica Desde el Espacio: la Cámara OMC a Bordo de INTEGRAL. Barcelona, 2011, 310 p. Tesis (Doctor por la universidad de Barcelona). Universidad de Barcelona. Departamento de Astronomía y Meteorología.
- [10] REÁTEGUI WOLL, Jaime Cesar. Estimación Automática de la Varianza y Discriminación de Distribuciones de Ruido en Imágenes Digitales. Lima, 2012, 41 p. Tesis (Ingeniero Electrónico). Pontificia Universidad Católica del Perú. Facultad de Ciencias e Ingeniería.

[11] DEVORE, Jay L. Probabilidad y Estadística para Ingeniería y Ciencias. Sexta edición. Thomson, 2005.

[12] MURRAY R. Spiegel. Probabilidad y estadística. Segunda edición: Mc Graw Hill, 1977.

[13] MASON, Robert D y LIND, Douglas A. estadística para administradores y economía. Octava edición: Alfaomega, 1998.

[14] MARTÍNES FERNÁNDEZ, Laura. Métodos de Inferencia para la Distribución Weibull, Aplicación en Fiabilidad Industrial. España, 2011, 60 p. Tesis (Máster en Técnicas Estadísticas). Universidad de Vigo. Departamento de Estadística e Investigación Operativa.

[15] CARCAMO C, Ulises. Elementos de Probabilidad para el Estudio del Riesgo. Fondo editorial Universidad EAFIT. Medellín, 2000.

[16] GIBBONS DICKINSON, Jean y CHAKRABORTI Subhabrata. Nonparametric Statistical Inference. Fourth Edition. The University of Alabama.

[17] DE LA GARZA GARCÍA, Jorge; MORALES SERRANO, Blanca Nieves y GONZÁLES CAVAZOS, Beatriz Adriana. Análisis Estadístico Multivalente, Un Enfoque Teórico Práctico. Primera edición: Mc Graw Hill, 2013.

[18] PEÑA, Daniel. Análisis de Datos Multivariantes. Primera edición: Mc Graw Hill, 2002.

[19] MARTÍNES BERNCARDINO, Ciro. Estadística y Muestreo. Décimo Segunda Edición. Bogotá. Ecoe Ediciones, 2005.

[20] GONZALES, Rafael C y WOODS, Richard E. Procesamiento digital de imágenes. Tercera edición: Addison-Wesley/Diaz de Santos, 1996

ANEXOS

El código desarrollado para el presente trabajo fue realizado en Matlab y fue corrido en un SAMSUNG SF411 con procesador intel core i5-2410M de 2.30 GHz, memoria RAM instalada de 4 GB bajo un sistema operativo de 64 bits.

Anexo 1. Código para determinar el comportamiento de valores tomados de diferentes imágenes.

Anexo 1.1 Prueba_Funciones.

```
% calcula el estadístico Chi cuadrado y Kolmogorov-Smirnov a una matriz formada muestras
%aleatorias tomadas de cada imagen.

clc

clear all
tic
a=''; % el path con la dirección de las carpetas donde se encuentran las imágenes.
b='.fit'; % extensión de las imágenes.
Numero_Img=60;
romper_ciclo=1;
while romper_ciclo<15 % determina el número de corridas a realizar en este caso 14.
    [N,Y,Promedios,Varianzas]=Tama_muestra(a,b,Numero_Img);
    [M_val_Pixel]=Misma_muestra(N,Y,Numero_Img,Promedios);
    Matriz_para_mirar{romper_ciclo}=M_val_Pixel;
    [Matriz_Chi,Numero_clases_matriz,Matriz_D]=prueba_bya(M_val_Pixel);
    Matriz_Chi_R(romper_ciclo,:)=Matriz_Chi;
    Numero_clases_matriz_R(romper_ciclo,:)=Numero_clases_matriz;
    Matriz_D_R(romper_ciclo,:)=Matriz_D;
```

```

    romper_ciclo=romper_ciclo+1;
end
Matriz_Chi_R
Numero_clases_matriz_R
Matriz_D_R
Toc

```

Anexo 1.2 Tama_muestra

```

%carga las imágenes y genera un tamaño de muestra para cada una de ella a partir de una
%muestra aleatoria de 3000 pixeles estima la media y la varianza de cada imagen para
%posteriormente calcular los tamaños de las muestras a un intervalo de confianza del 97 y
%con un error del 1%.

```

```

function [N,Y,Promedios,Varianzas]=Tama_muestra(a,b,Numero_Img)
contar=3;

```

```

for i=1:Numero_Img
    contar=num2str(contar);
    x=strcat(a,contar,b);
    Y{i}=fitsread(x);
    contar=str2num(contar);
    contar=contar+10;
end

```

```

% se calculan las varianzas para cada una de las imágenes. Se generan
% números aleatorios diferentes para cada imagen.

```

```

romper=1;
while romper <= 60
    tama_muestra=3000;
    z=aleat_sin_rept(tama_muestra);
    z=sort(z);
    Matriz=Coor_Pixel(z);
    cont2=0;
    a=Y{romper};
    for i=1:tama_muestra
        cont2=cont2+1;
        M_val_Pixel(cont2,1)=a(Matriz(i,1),Matriz(i,2));
        Aleatorios=z;
        while M_val_Pixel(cont2,1) > 5000% el valor máximo que se espera como corriente de
            %oscuridad.

            tama_muestra=1; % Solo se genera un aleatorio.
            z1=aleat_sin_rept(tama_muestra); %genera un nuevo aleatorio.

            if (z1~=z)
                z=z1; % Se cambia el valor de "z" para que pueda ser leído por la función
                    % "Coor_Pixel(z)".
                Matriz1=Coor_Pixel(z); % Coordenadas del aleatorio generado.
                M_val_Pixel(cont2,1)=a(Matriz1(1,1),Matriz1(1,2));
                z=[Aleatorios;z];
                Aleatorios=z; %Acumula los aleatorios para que no se repitan.
            else
                M_val_Pixel(cont2,1);
            end
        end
    end
end
end

```

```

Matrices_M_val_Pixel{romper}=M_val_Pixel;
Varianzas(romper,1)=var(M_val_Pixel);
Promedios(romper,1)=mean(M_val_Pixel);
Error(romper,1)=Promedios(romper,1)*0.01;
Z=1.880793608;

No(romper,1)=(Varianzas(romper,1)*Z^2)/Error(romper,1)*(1+(2/3000));
N(romper,1)=No(romper,1)/(1+(No(romper,1)/1920000));
N=ceil(N); %redondea al entero superior. Tamaño de las muestras para cada una de las
           %imágenes.

romper=romper+1;
end
end

```

Anexo 1.3 aleat_sin_rept

%Esta función genera números aleatorios provenientes de una distribución uniforme todos ellos diferentes de 1 a 1920000 correspondientes a la cantidad de pixeles presentes en las imágenes y tantos como tamaño de muestra.

```

function z=aleat_sin_rept(tama_muestra)
a=0;
n=1;
while n <=tama_muestra
d=randi(1920000,1,1); % 1-1920000 Esta es la cantidad de pixeles en el sensor.

```

```

if (a ~= d)
    a=[d,a];
    n=n+1;
end
end
a=a';
a=a(1:(end-1),1);
z=a;
end

```

Anexo 1.4 Coor_Pixel

%Esta función encuentra las coordenadas en filas y en columnas que están relacionadas con %un aleatorio entre 1 y 19200000, un ejemplo de ello si el aleatorio es 3456 las %coordenadas del pixel asociadas son (3,256) correspondientes al número de fila y columna %respectivamente.

```

function Matriz=Coor_Pixel(z)
n=1;
Tama_aleator=size(z);
for i=1:1200
    if i==1
        while z(n,1)<=1600 % si el aleatorio es menor a 1600 se encuentra en la
            % primera fila.
            Matriz(n,:)=[i,z(n,1)]; %Genera la matriz con las coordenadas del
                % pixel.
            n=n+1;
            if n>Tama_aleator(1,1)

```

```

        break % Si la condición cumple, rompe el while.
    end
end
end
if n>Tama_aleator(1,1)
    break % Si la condición cumple, rompe el for.
end
    while z(n,1)<=1600*i
        Matriz(n,:)=[i,(-1600*(i-1)+z(n,1))]; % determina las coordenadas
            % cuando la fila es diferente de uno.
        n=n+1;
        if n>Tama_aleator(1,1)
            break % Si la condición cumple, rompe el while.
        end
    end
    if n>Tama_aleator(1,1)
        break % Si la condición cumple, rompe el for.
    end
end
end
end

```

Anexo 1.5 Misma_muestra

```

%Genera tantos números aleatorios como tamaño de muestra estimada para cada imagen, de
%manera que se tome una muestra aleatoria a cada una de ellas. A dichos pixeles
%muestreados se les realizara la medición de sus cuentas recopilando de esta manera los
%datos a los que se les aplicara las pruebas de bondad y ajuste.

```

```

function [M_val_Pixel]=Misma_muestra(N,Y,Numero_Img,Promedios)
cont=1;
cont2=0; % Índice para M_val_Pixel
while cont<=Numero_Img %% Corresponde al número de imágenes.
    tama_muestra=N(cont,1);
    z=aleat_sin_rept(tama_muestra); %Genera aleatorios tantos como tama_muestra
    z=sort(z); % Se ordenan los aleatorios de menos a mayor.
    Matriz=Coor_Pixel(z); % Transforma el aleatorio a coordenadas en la imagen.
    a=Y{cont}; % "a" Contiene la imagen en la posición "cont".
    for i=1:tama_muestra %% tamaño de la muestra.
        cont2=cont2+1;
        M_val_Pixel(cont2,1)=a(Matriz(i,1),Matriz(i,2)); % "M_val_Pixel" Es la matriz con
        %las coordenadas de los pixeles evaluadas.
        Aleatorios=z;
        % El ciclo siguiente es para no tener en cuenta los pixeles
        %calientes.
        while M_val_Pixel(cont2,1) > 1600
            tama_muestra=1; % Solo se genera un aleatorio.
            z1=aleat_sin_rept(tama_muestra); %genera un nuevo aleatorio.
            if (z1~=z)
                z=z1; % Se cambia el valor de "z" para que pueda ser leído por la función
                % "Coor_Pixel(z)".
                Matriz1=Coor_Pixel(z); % Coordenadas del aleatorio generado.
                M_val_Pixel(cont2,1)=a(Matriz1(1,1),Matriz1(1,2));
                z=[Aleatorios;z];
                Aleatorios=z; %Acumula los aleatorios para que no se repitan.
            else
                M_val_Pixel(cont2,1);
            end
        end
    end
end
cont=cont+1;

```

```

end
M_val_Pixel; %% Matriz con las coordenadas de los pixeles evaluados.
end

```

Anexo 1.6 prueba_bya

%Esta función realiza las pruebas de bondad y ajuste tanto Chi-cuadrado como Kolomogorov-Smirnov a la matriz que contiene los valores de los pixeles evaluados.

```
function [Matriz_Chi,Numero_clases_matriz,Matriz_D]=prueba_bya(M_val_Pixel)
```

```
M_val_Pixel=sort(M_val_Pixel); %% ordena los valores de menor a mayor.
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Grafica los histogramas%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
Var_discre=1; %1 si es discreta 0 si es continua.
```

```
[Limites,Frecuen_Rela_Inter]=Frecuencia_intervalos(M_val_Pixel,Var_discre);
```

```
% Determina la frecuencia y limites de cada intervalo.
```

```
tama_M_val_Pixel=size(M_val_Pixel); % Dimensiones de "tama_M_val_Pixel".
```

```

Matriz_tama_M_val_pixel(1,1)=tama_M_val_Pixel(1,1);

Tam_Limi=size(Limites); % Dimensiones de "Tam_Limi".

for u=1:Tam_Limi(1,1)
    centro(u,1)=mean(Limites(u,:)); % Calcula el centro entre limites para
                                     % graficar el histograma.
end

```

```

centro_histograma{1}=centro; % Matriz con las matrices de los centros.
figure,hist(M_val_Pixel,centro); % Grafica el histograma.
xlabel('Cuentas') % titulo para el eje "x".
ylabel('Frecuencia') % titulo para el eje "y".

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%BONDAD Y AJUSTE KS%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

D_Norm=K_S_Nor(M_val_Pixel); % Realiza una prueba K-S para Normalidad.
D_Wei=K_S_Weibull(M_val_Pixel); % Realiza una prueba K-S para Weibull.
D_Gamm=K_S_Gamma(M_val_Pixel); %Realiza una prueba K-S para Gamma.
D_Poisson=K_S_Poss(M_val_Pixel);
% "Matriz_D" contiene en valor "D" de cada una de las pruebas K-S.
Matriz_D(1,1)=D_Norm;
Matriz_D(1,2)=D_Gamm;
Matriz_D(1,3)=D_Wei;
Matriz_D(1,5)=D_Poisson;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%BONDAD Y AJUSTE CHI CUADRADO%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%DESCRIPTIVAS.
Media=mean(M_val_Pixel);
Desviacion=std(M_val_Pixel);
Numero_datos=tama_M_val_Pixel(1,1);

[Chi_Cuadra_Normal,numero_clases,Chi_Cua]=Chi_Cuad_Norm(Media,Desviacion,Numero_datos,Limites,Frecuen_Rela_Inter);% Realiza la prueba Chi para probar Normalidad.
Numero_clases_matriz(1,1)=numero_clases; % Numero de clases que se generaron en la prueba
%para la normalidad.
[Chi_Cuadra_Gamma,numero_clases]=Chi_Cuad_gamma(Numero_datos,Limites,Frecuen_Rela_Inter,M_val_Pixel); % Realiza la prueba Chi para Gamma.
Numero_clases_matriz(1,2)=numero_clases; %% Numero de clases que se generaron en la prueba
%Gamma.
[Chi_cuad_weibull,numero_clases]=Chi_Cuad_weibull(Numero_datos,Limites,Frecuen_Rela_Inter,M_val_Pixel); % Realiza la prueba Chi para probar weibull.
Numero_clases_matriz(1,3)=numero_clases;% Numero de clases que se generaron en la prueba
%para la weibull.
[Chi_Cuadra_Poisson,numero_clases]=Chi_Cua_Poiss(Media,Limites,Numero_datos,Frecuen_Rela_I
nter);
Numero_clases_matriz(1,5)=numero_clases; %numero de clases para la poisson
% "Matriz_Chi" Contiene los valores chi estimados.
Matriz_Chi(1,1)=Chi_Cuadra_Normal; % Prueba Chi para normal.
Matriz_Chi(1,2)=Chi_Cuadra_Gamma; % Prueba Chi para gamma.
Matriz_Chi(1,3)=Chi_cuad_weibull; % Prueba Chi para weibull.
Matriz_Chi(1,5)=Chi_Cuadra_Poisson;% Prueba Chi para Poisson.

%%%%%%%%%%
%%LOGARITMOS NATURALES%%
%%%%%%%%%%

```

```

% Se intenta determinar si los datos proceden de una distribución Log-Normal.
M_val_Pixel=log(M_val_Pixel); % Se toman los logaritmos naturales a la matriz.
%"M_val_Pixel"
D_Logorm=K_S_Nor1(M_val_Pixel); %Prueba K-S para log normal.
Matriz_D(1,4)=D_Logorm;
Var_discre=0; %1 si es discreta 0 si es continua la M_val_Pixel.
[Limites,Frecuen_Rela_Inter]=Frecuencia_intervalos(M_val_Pixel,Var_discre);
Media=mean(M_val_Pixel);
Desviacion=std(M_val_Pixel);
Numero_datos=tama_M_val_Pixel(1,1);
[Chi_Cuadra_Normal,numero_clases]=Chi_Cuad_Norm(Media,Desviacion,Numero_datos,Limites,Frec
uen_Rela_Inter); %prueba chi para log normal.
Numero_clases_matriz(1,4)=numero_clases; % Numero de clases de Lognormal.
Matriz_Chi(1,4)=Chi_Cuadra_Normal;
end

```

Anexo1.7 Frecuencia_intervalos

%Esta función determina el numero de clases, mediante la regla de Sturges, los límites inferiores y superiores de los datos generados así como también la frecuencia en cada uno de ellos.

```

function [Limites,Frecuen_Rela_Inter]=Frecuencia_intervalos(M_val_Pixel,Var_discre)
a=size(M_val_Pixel);
format short
rango=max(M_val_Pixel)-min(M_val_Pixel);
numero_clases=1+log2(a(1,1));
numero_clases=ceil(numero_clases);
ancho_clase=rango/numero_clases;

```

```

if Var_discre==1
    minimo=min(M_val_Pixel); %% el valor mínimo de la matriz.
    frecuencia=tabulate(M_val_Pixel);
    frecuencia=frecuencia(minimo:end,1:2);
elseif Var_discre==0 %la variable es continua.
    frecuencia=tabulate(M_val_Pixel);
    frecuencia=frecuencia(1:end,1:2);

end

for i=1:numero_clases
    if i==1
        Limites(i,1)=frecuencia(i,1);
        Limites(i,2)=frecuencia(i,1)+ancho_clase;
    elseif i>1
        Limites(i,1)=Limites(i-1,2);
        Limites(i,2)=Limites(i,1)+ancho_clase;

    end
end

b=size(frecuencia);
b1=size(Limites);

b=size(frecuencia);
b1=size(Limites);
for i=1:b1(1,1)
    suma_frecu=0;
    for j=1:b(1,1)
        if i>1 && frecuencia(j,1)>Limites(i-1,2) && frecuencia(j,1)<=Limites(i,2)
            suma_frecu=suma_frecu+frecuencia(j,2);
            Frecuen_Relacion_Inter(i,1)=suma_frecu;
        end
    end
end

```

```

elseif frecuencia(j,1)<=Limites(i,2) && i==1
    suma_frecu=suma_frecu+frecuencia(j,2);
    Frecuen_Rela_Inter(i,1)=suma_frecu;
elseif i==b1(1,1) && Limites(i,1)<frecuencia(j,1)
    suma_frecu=suma_frecu+frecuencia(j,2);
    Frecuen_Rela_Inter(i,1)=suma_frecu;
end
end
end
end

```

Anexo 1.8 K_S_Nor

```

%calcula el estadístico D para una prueba de normalidad.

function D=K_S_Nor(M_val_Pixel)
media=mean(M_val_Pixel);
desviacion=std(M_val_Pixel);
a=size(M_val_Pixel);
Pro=1/a(1,1);
cont1=0;
proba_empirica=0;
for i=1:a(1,1)
    X_Normalizada(i,1)=(M_val_Pixel(i)-media)/desviacion;
    Distri_nor_Es(i,1)=normcdf(X_Normalizada(i,1));
    cont1=cont1+Pro;
    proba_empirica(i,1)=cont1;
    Observaciones(i,1)=i; %% número total de observaciones
end
D(1,1)=max(abs(proba_empirica-Distri_nor_Es)); %% D+
D(1,2)=max(Distri_nor_Es-((Observaciones-1)/a(1,1))); %% D-

```

```
D=max(D);  
end
```

Anexo 1.9 K_S_Weibull

%Calcula el estadístico D para una distribución weibull.

```
function D=K_S_Weibull(M_val_Pixel)  
a=size(M_val_Pixel);  
Parametros=wblfit(M_val_Pixel);  
format long  
Pro=1/a(1,1);  
cont1=0;  
proba_empirica=0;  
for i=1:a(1,1)  
cont1=cont1+Pro;  
proba_empirica(i,1)=cont1;  
Observaciones(i,1)=i; %% número total de observaciones.  
Pro_Acum_Wei(i,1)= 1-exp(-(M_val_Pixel(i,1)/Parametros(1))^Parametros(2)); %distribución  
%acumulada.  
end  
Pro_Acum_Wei;  
proba_empirica;  
  
D(1,1)=max(abs(proba_empirica-Pro_Acum_Wei)); %% D+  
D(1,2)=max(Pro_Acum_Wei-((Observaciones-1)/a(1,1))); %% D-  
D=max(D);  
end
```

Anexo 1.10 K_S_Gamma

%calcula el estadístico D para una distribución de Gamma.

```
function D=K_S_Gamma(M_val_Pixel)
a=size(M_val_Pixel);
Parametros=gamfit(M_val_Pixel); %forma y escala;
Pro=1/a(1,1);
cont1=0;
proba_empirica=0;
for i=1:a(1,1)
cont1=cont1+Pro;
proba_empirica(i,1)=cont1;
Observaciones(i,1)=i; %% número total de obserbaciones.
Pro_Acum_gamma(i,1)= gamcdf(M_val_Pixel(i,1),Parametros(1),Parametros(2));
end
D(1,1)=max(abs(proba_empirica-Pro_Acum_gamma)); %% D+
D(1,2)=max(abs(Pro_Acum_gamma-((Observaciones-1)/a(1,1)))); %% D
D=max(D);
end
```

Anexo 1.11 K_S_Poss

%calcula el estadístico D para una distribución de poisson.

```

function D_Poisson=K_S_Poss(M_val_Pixel)
Media=mean(M_val_Pixel); %
a=size(M_val_Pixel);
cont1=0;
for i=1:a(1,1)
Pro=1/a(1,1);
cont1=cont1+Pro;
proba_empirica_acum(i,1)=cont1;
Observaciones(i,1)=i;
Proba_esperada_acum(i,1)=poisscdf(M_val_Pixel(i,1),Media);

end
D_Poisson(1,1)=max(abs(Proba_esperada_acum-proba_empirica_acum));
D_Poisson(1,2)=max(Proba_esperada_acum-((Observaciones-1)/a(1,1)));
D_Poisson=max(D_Poisson);
end

```

Anexo 1.12 Chi_Cuad_Norm

```

%determina el chi para una normalse toman los intervalos teniendo en cuenta los infinitos
%pues es una probabilidad teórica.

function[Chi_Cuadra_Normal,numero_clases,Chi_Cua]=Chi_Cuad_Norm(Media,Desviacion,Numero_da
tos,Limites,Frecuen_Rela_Inter)
b1=size(Limites);
Limites(1,1)=-inf;
Limites(b1(1,1),2)=inf;

```

```

a=size(Limites);
for i=1:a(1,1)
    X_Normalizada(i,1)=(Limites(i,1)-Media)/Desviacion; %Se normaliza la variable.
    X_Normalizada(i,2)=(Limites(i,2)-Media)/Desviacion;
    Distri_nor_Es(i,1)=normcdf(X_Normalizada(i,1));
    Distri_nor_Es(i,2)=normcdf(X_Normalizada(i,2));
    if i==1
        proba_interva(i,1)=Distri_nor_Es(i,2); % Se calcula la probabilidad del
            %intervalo.
    elseif i<a(1,1)
        proba_interva(i,1)=Distri_nor_Es(i,2)-Distri_nor_Es(i,1);
    else
        proba_interva(i,1)=1-Distri_nor_Es(i,1);
    end
    Frecuenc_Esperada(i,1)=proba_interva(i,1)*Numero_datos; % Se calcula la frecuencia
        %esperada.
    Chi_Cua(i,1)=( abs(Frecuen_Rela_Inter(i,1)-Frecuenc_Esperada(i,1)) -
        0.5)^2/Frecuenc_Esperada(i,1);

end

% Si la frecuencia esperada en algunos de los intervalos es inferior a 5.
% Se procede a realizar la corrección mediante la función "intervalo_corre".

Tama_Frecuenc_Esperada=size(Frecuenc_Esperada);
if Frecuenc_Esperada(Tama_Frecuenc_Esperada(1,1),1) < 5
    clear Chi_Cua

[Frecuenc_Esperada,Limites,Frecuen_Rela_Inter]=intervalo_corre(Frecuenc_Esperada,Limites,F
recuen_Rela_Inter);
a=size(Limites);

```

```

    for i=1:a(1,1)
        Chi_Cua(i,1)=(abs(Frecuen_Rela_Inter(i,1)-Frecuenc_Esperada(i,1))-
0.5)^2/Frecuenc_Esperada(i,1);
    end
end

numero_clases=a(1,1); % Numero de clases después de la corrección.
Chi_Cuādra_Normal=sum(Chi_Cua); %Chi cuadrado para la normal.
end

```

Anexo 1.13 Chi_Cuad_gamma

```

%determina el chi para Gamma.

function
[Chi_Cuadra_Gamma,numero_clases]=Chi_Cuad_gamma(Numero_datos,Limites,Frecuen_Rela_Inter,M_
val_Pixel)
Parametros=gamfit(M_val_Pixel);% determina los parámetros de la función gamma forma y
escala por el método de máxima verosimilitud.
a=size(Limites);
Limites(a(1,1),2)=inf; % esta función está definida [0,inf).
for i=1:a(1,1)
    if i==1
        Pro_Int_gamma(i,1)= gamcdf(Limites(i,2),Parametros(1),Parametros(2));
    elseif i<a(1,1)

```

```

        Pro_Int_gamma(i,1)= gamcdf(Limites(i,2),Parametros(1),Parametros(2))-
gamcdf(Limites(i,1),Parametros(1),Parametros(2));
    else
        Pro_Int_gamma(i,1)= 1-gamcdf(Limites(i,1),Parametros(1),Parametros(2));
    end
    Frecuenc_Esperada(i,1)=Pro_Int_gamma(i,1)*Numero_datos;
    Chi_Cua(i,1)=(abs(Frecuen_Rela_Inter(i,1)-Frecuenc_Esperada(i,1))-
        0.5)^2/Frecuenc_Esperada(i,1);
end
% Si la frecuencia esperada en algunos de los intervalos es inferior a 5
% Se procede a realizar la corrección mediante la función "intervalo_corre".

Tama_Frecuenc_Esperada=size(Frecuenc_Esperada);
if Frecuenc_Esperada(Tama_Frecuenc_Esperada(1,1),1) < 5
    clear Chi_Cua

    [Frecuenc_Esperada,Limites,Frecuen_Rela_Inter]=intervalo_corre(Frecuenc_Esperada,Limites,F
recuen_Rela_Inter);
    a=size(Limites);
    for i=1:a(1,1)
        Chi_Cua(i,1)=(abs(Frecuen_Rela_Inter(i,1)-Frecuenc_Esperada(i,1))-
0.5)^2/Frecuenc_Esperada(i,1);
    end
end
numero_clases=a(1,1);
Chi_Cuadra_Gamma=sum(Chi_Cua);
end

```

Anexo 1.4 Chi_Cuad_weibull

```
%Calcula el estadístico chi para weibull.
```

```
function
```

```
[Chi_cuad_weibull,numero_clases]=Chi_Cuad_weibull(Numero_datos,Limites,Frecuen_Rela_Inter,  
M_val_Pixel)
```

```
Parametros=wblfit(M_val_Pixel); % calcula los parámetros de la distribución
```

```
% weibull por el método de máxima verosimilitud.
```

```
a=size(Limites);
```

```
Limites(a(1,1),2)=inf; %La weibull solo está definida para valores >= 0.
```

```
for i=1:a(1,1)
```

```
    if i==1
```

```
        Pro_Int_weibull(i,1)= wblcdf(Limites(i,2),Parametros(1),Parametros(2));
```

```
        %probabilidad del intervalo.
```

```
    elseif i<a(1,1)
```

```
        Pro_Int_weibull(i,1)= wblcdf(Limites(i,2),Parametros(1),Parametros(2)) -
```

```
wblcdf(Limites(i,1),Parametros(1),Parametros(2));
```

```
    else
```

```
        Pro_Int_weibull(i,1)= 1-wblcdf(Limites(i,1),Parametros(1),Parametros(2));
```

```
    end
```

```
    Frecuenc_Esperada(i,1)=Pro_Int_weibull(i,1)*Numero_datos; %frecuencia esperada.
```

```
    Chi_Cua(i,1)=(abs(Frecuen_Rela_Inter(i,1)-Frecuenc_Esperada(i,1)) -
```

```
        0.5)^2/Frecuenc_Esperada(i,1);
```

```
end
```

```
% Si la frecuencia esperada en algunos de los intervalos es inferior a 5
```

```
% Se procede a realizar la corrección mediante la función "intervalo_corre".
```

```
Tama_Frecuenc_Esperada=size(Frecuenc_Esperada);
```

```
if Frecuenc_Esperada(Tama_Frecuenc_Esperada(1,1),1) < 5
```

```

clear Chi_Cua

[Frecuenc_Esperada,Limites,Frecuen_Rela_Inter]=intervalo_corre(Frecuenc_Esperada,Limites,F
recuen_Rela_Inter);
a=size(Limites);
for i=1:a(1,1)
Chi_Cua(i,1)=(abs(Frecuen_Rela_Inter(i,1)-Frecuenc_Esperada(i,1))-
0.5)^2/Frecuenc_Esperada(i,1);
end
end
Chi_cuad_weibull=sum(Chi_Cua);
numero_clases=a(1,1);
end

```

Anexo 1.15 Chi_Cua_Poiss

%determina el chi para una Poisson.

```

function
[Chi_Cuadra_Poisson,numero_clases]=Chi_Cua_Poiss(Media,Limites,Numero_datos,Frecuen_Rela_I
nter)
a=size(Limites);
Limites(a(1,1),2)=inf;

for i=1:a(1,1)
if i==1
probabilidad_Intervalo(i,1)=poisscdf(Limites(i,2),Media);
Frecuenc_Esperada(i,1)=probabilidad_Intervalo(i,1)*Numero_datos;
elseif i<a(1,1)

```

```

        probabilidad_Intervalo(i,1)=poisscdf(Limites(i,2),Media) -
poisscdf(Limites(i,1),Media);
        Frecuenc_Esperada(i,1)=probabilidad_Intervalo(i,1)*Numero_datos;
    else
        probabilidad_Intervalo(i,1)=1-sum(probabilidad_Intervalo);
        Frecuenc_Esperada(i,1)=probabilidad_Intervalo(i,1)*Numero_datos;
    end
    Chi_Cua(i,1)=(Frecuen_Rela_Inter(i,1)-Frecuenc_Esperada(i,1))^2/Frecuenc_Esperada(i,1);
end
Chi_Cuadra_Poisson=sum(Chi_Cua);

Tama_Frecuenc_Esperada=size(Frecuenc_Esperada);
if Frecuenc_Esperada(Tama_Frecuenc_Esperada(1,1),1) < 5
    clear Chi_Cua

[Frecuenc_Esperada,Limites,Frecuen_Rela_Inter]=intervalo_corre(Frecuenc_Esperada,Limites,F
recuen_Rela_Inter);
    a=size(Limites);
    for i=1:a(1,1)
        Chi_Cua(i,1)=(Frecuen_Rela_Inter(i,1)-Frecuenc_Esperada(i,1))^2/Frecuenc_Esperada(i,1);
    end
end
numero_clases=a(1,1);
Chi_Cuadra_Poisson=sum(Chi_Cua);
end

```

Anexo 1.16 intervalo_corre

%esta función corrige aquellas clases que frecuencias esperadas son menores a 5.

```
function[Frec_Esper_Corre,Limites_corre,Frecuen_Rela_Inter_corre]=intervalo_corre(Frecuenc
_Esperada,Limites,Frecuen_Rela_Inter)
Tama_Frecuenc_Esperada=size(Frecuenc_Esperada);
cont=Tama_Frecuenc_Esperada(1,1);
cont1=1;
cont2=1;
for i=Tama_Frecuenc_Esperada(1,1):-1:1

    if Frecuenc_Esperada(i,1) < 5 && cont==Tama_Frecuenc_Esperada(1,1)
        cont3 =0;
        Frec_Esper_Co=0;

        while Frecuenc_Esperada(i,1) < 5

            Frec_Esper_Co= Frecuenc_Esperada(cont,1)+ Frec_Esper_Co;
            cont=cont-1;
            Frecuenc_Esperada(i,1)=Frec_Esper_Co;
            cont3=cont3+1;
        end

        Frec_Esper_Corre=Frecuenc_Esperada(1:cont,1);
        Frec_Esper_Corre(cont+1,1)=Frec_Esper_Co;
        Limites_corre=Limites(1:cont,:);
        Limites_corre(cont+1,1)=Limites(cont+1,1);
        Limites_corre(cont+1,2)=Limites(Tama_Frecuenc_Esperada(1,1),2);
        Frecuen_Rela_Inter_corre=Frecuen_Rela_Inter(1:cont,1);

Frecuen_Rela_Inter_corre(cont+1,1)=sum(Frecuen_Rela_Inter(cont+1:Tama_Frecuenc_Esperada(1,
1),1));
    for il=cont+1:i
```

```

        Frecuenc_Esperada(i1,1)=Frec_Esper_Co;
    end
end
end
end

```

Anexo 2 Código desarrollado para identificar las posibles distribuciones de los pixeles en particular.

Anexo 2.1 Val_mismo_pixel

```

clc
clear all
tic

a=''; % Phat con la dirección de las imágenes.
b='.fit';
Numero_Img=60;
[M_val_Pixel,z]=Mismos_pixels(a,b,Numero_Img);
tamao_Mat_Val_Pix=size(M_val_Pixel);
Numero_datos=tamao_Mat_Val_Pix(1,1); %para calcular la cantidad de clases en las pruebas.
%Chi
M_val_Pixel1=M_val_Pixel; %mantiene la variable.

for i=1:tamao_Mat_Val_Pix(1,2)
    [M_val_Pixel]=M_val_Pixel1(:,i);
    [M_val_Pixel]=sort(M_val_Pixel);
    [Matriz_Chi,Numero_clases_matriz,Matriz_D]=prueba_bya(M_val_Pixel);
    Resultado_Chi(:,i)=Matriz_Chi';
    Numero_Clasas_R(:,i)=Numero_clases_matriz';
end

```

```

Resultado_K_S(:,i)=Matriz_D';
end
Resultado_Chi % resultados de las pruebas chi.
Resultado_K_S % resultados de las pruebas ks.
[Pxno_Cu_chi,FnDisPix,T_Norm_C,T_Gam_C,T_Wei_C,T_LoNor_C,T_Pos_C]=Res_Chi(Resultado_Chi,Nu
mero_Clases_R)

%resultados para la ks.
Tama_Re_K_S=size(Resultado_K_S);
Menor=min(Resultado_K_S);
Total_Normal_KS=0;
Total_Gamma_KS=0;
Total_Weibull_KS=0;
Total_Log_Normal_KS=0;
Total_Poisson_KS=0;
C_ind=1;
Pixel_no_cumple_KS=0;
for i=1:Tama_Re_K_S(1,2)
    if Menor(1,i) < 0.1756
        if Resultado_K_S(1,i)==Menor(1,i)
            Total_Normal_KS=Total_Normal_KS+1;
            Fn_Ds_KS(1,i)=1;
        elseif Resultado_K_S(2,i)==Menor(1,i)
            Total_Gamma_KS=Total_Gamma_KS+1;
            Fn_Ds_KS(1,i)=2;
        elseif Resultado_K_S(3,i)==Menor(1,i)
            Total_Weibull_KS=Total_Weibull_KS+1;
            Fn_Ds_KS(1,i)=3;
        elseif Resultado_K_S(4,i)==Menor(1,i)
            Total_Log_Normal_KS=1+Total_Log_Normal_KS;

```

```

        Fn_Ds_KS(1,i)=4;
    elseif Resultado_K_S(5,i)==Menor(1,i)
        Total_Poisson_KS=Total_Poisson_KS+1;
        Fn_Ds_KS(1,i)=5;
    end

else
    Pixel_no_cumple_KS(1,C_ind)=i;
    C_ind=C_ind+1;
end

end

Fn_Ds_KS % vector que muestra que distribución sigue cada pixel
Pixel_no_cumple_KS % pixel que rechaza la hipótesis nula.
Total_Normal_KS % total pixeles que se ajustan a una normal según esta prueba.
Total_Gamma_KS % total pixeles que se ajustan a una gamma.
Total_Weibull_KS % total pixeles que se ajustan a una weibull.
Total_Log_Normal_KS % total pixeles que se ajustan a una log normal.
Total_Poisson_KS % total pixeles que se ajustan a una poisson.
toc

```

Anexo 2.2 Mismos_pixels

```

%Mide el valor del pixel en las mismas coordenadas para diferentes imágenes (en
%profundidad).

```

```

function [M_val_Pixel,z]=Mismos_pixels(a,b,Numero_Img)
contar=3;

for i=1:Numero_Img
    contar=num2str(contar);
    x=strcat(a,contar,b);
    Y{i}=fitsread(x);
    contar=str2num(contar);
    contar=contar+10;
end
tama_muestra=10; % la cantidad de pixeles a ser seleccionados.
z=aleat_sin_rept(tama_muestra);
z=sort(z);
Matriz=Coor_Pixel(z);
Tama_Matriz=size(Matriz);

cont=1;

while cont<=Numero_Img
    a=Y{cont};
    for i=1:Tama_Matriz(1,1) %% tamaño de la muestra.
        M_val_Pixel(cont,i)=a(Matriz(i,1),Matriz(i,2));
    end
    cont=cont+1;
end
M_val_Pixel; % matriz con los valores de los pixeles las coordenadas ya evaluadas.
End

```

Anexo 2.3 Res_Chi

```
%realiza la prueba chi a un 95% de confianza además de identificar que pixeles se ajustan
%mejor a una distribución cuenta e identifica los pixeles que rechazan las hipótesis nulas
%y cuáles son estos.
```

```
function[Pxno_Cu_chi,FnDisPix,T_Norm_C,T_Gam_C,T_Wei_C,T_LoNor_C,T_Pos_C]=Res_Chi(Resultad
o_Chi,Numero_Clasas_R)
Tama_Numero_clases=size(Numero_Clasas_R);
T_Norm_C=0;
T_Gam_C=0;
T_Wei_C=0;
T_LoNor_C=0;
T_Pos_C=0;
cont=1;
Pxno_Cu_chi=0;
  for q=1:Tama_Numero_clases(1,2)%las columnas.
    for q1=1:Tama_Numero_clases(1,1)%las filas.
      Gra_Lib=Numero_Clasas_R(q1,q)-1-2;% grados de libertad.
      Chi_Critico(q1,q)=chi2inv(0.95,Gra_Lib);
      if q1==Tama_Numero_clases(1,1)
        Gra_Lib=Numero_Clasas_R(q1,q)-1-1; %grados libertad poisson.
        Chi_Critico(q1,q)=chi2inv(0.99,Gra_Lib);
      end
      if Resultado_Chi(q1,q) > Chi_Critico(q1,q)
        Resultado_Chi(q1,q)=inf;
      end
    end

    if min(Resultado_Chi(:,q))==inf
      FnDisPix(1,q)=6; % etiqueta para identificar que pixeles rechazan la hipótesis
%nula.
```

```

    Pxno_Cu_chi(1,cont)=q;
    cont=cont+1;
elseif Resultado_Chi(1,q)== min(Resultado_Chi(:,q))
    FnDisPix(1,q)=1; %normal.
    T_Norm_C=T_Norm_C+1;
elseif Resultado_Chi(2,q)== min(Resultado_Chi(:,q))
    FnDisPix(1,q)=2; % gamma.
    T_Gam_C=T_Gam_C+1;
elseif Resultado_Chi(3,q)== min(Resultado_Chi(:,q))
    FnDisPix(1,q)=3; %weibull.
    T_Wei_C=T_Wei_C+1;
elseif Resultado_Chi(4,q)== min(Resultado_Chi(:,q))
    FnDisPix(1,q)=4; %log normal.
    T_LoNor_C=T_LoNor_C+1;
elseif Resultado_Chi(5,q)== min(Resultado_Chi(:,q))
    FnDisPix(1,q)=5; % poisson.
    T_Pos_C=T_Pos_C+1;
end
end

```

```

end
Pxno_Cu_chi;%pixeles que no cumplen.
%Resultado_Chi.
FnDisPix; %función que arroja un chi estimado menor.
T_Norm_C; %total pixeles que se ajustan a una normal.
T_Gam_C; % total pixeles que se ajustan a una gamma.
T_Wei_C; % total pixeles que se ajustan a una weibull.
T_LoNor_C; % total pixeles que se ajustan a una log normal.
T_Pos_C; % total pixeles que se ajustan a una poisson.
end

```

Anexo 3 Método de los promedios para el cálculo de las distancias entre los clústers.

Anexo 3.1 Promedios

```
clc
clear all

% se ingresan los segundos de arco por pixel, el número de objetos.

Numero_Objetos=input('ingrese el número de objetos: ');
Seg_Arc_Pixel=input('ingrese segundos de arco por pixel: ');
Matriz_Declinacion=input('ingrese las declinaciones: ');
Tama_Matriz_Declinacion=size(Matriz_Declinacion);

% Los siguientes mensajes son para garantizar que existan tantas coordenadas como numero
%de objetos menos 1.

if Tama_Matriz_Declinacion(1,1) >= Numero_Objetos
    error('Numero de coordenadas incorrecto en Declinación');
    break
end

Matriz_AscensionRecta=input('ingrese la Ascensión recta: ');
Tama_Matriz_Asces=size(Matriz_AscensionRecta);

if Tama_Matriz_Asces(1,1) >= Numero_Objetos
    error('Numero de coordenadas incorrecto en Ascensión Recta');
```

```

    break
end

a=fitsread(''); % Ingresar el phat de la imagen más su extensión. No todos los formatos
%son %leídos en Matlab.
a=flipdim(a,1); % rota la imagen 180° en x.
a=imrotate(a,-6.4515; %rota un angulo negativo.
imshow(a,[0 750]); %mejora el contraste para poder observar las estrellas.

a1=size(Matriz_Declinacion);
n=a1(1,1);

%las matrices ingresadas son llevadas ambas a arco segundos teniendo en cuenta si la
%declinación es al norte o al sur.

for i=1:n
    if Matriz_Declinacion(i,1)<0
matriz_decli_arcosegu(i,:)=Matriz_Declinacion(i,1)*3600-Matriz_Declinacion(i,2)*60-
Matriz_Declinacion(i,3);
        else
matriz_decli_arcosegu(i,:)=Matriz_Declinacion(i,1)*3600+Matriz_Declinacion(i,2)*60+Matriz_
Declinacion(i,3);
        end
matriz_ascenc_horas(i,:)=Matriz_AscensionRecta(i,1)+Matriz_AscensionRecta(i,2)/60+Matriz_A
scensionRecta(i,3)/3600;
    end
matriz_ascens_arcosegu=matriz_ascenc_horas*54000; %Ascensión recta en arco segundos.
matriz_decli_arcosegu;
n=1;
while n<=Numero_Objetos
pause % Espera para seleccionar el objeto. Se recomienda presionar la tecla "Esc".
t=getrect;

```

```

t=round(t); %evita el error de redondeo.
Y=a((t(1,2):(t(1,2)+t(1,4))), (t(1,1):(t(1,1)+t(1,3)))); % genera una matriz con los
%pixeles de la selección anterior.
a1=size(Y); % determina las dimensiones de "Y".
w=0;
for i=1:a1(1,1);
    for j=1:a1(1,2);
        if Y(i,j) < .7*max(max(Y)); %se toman los valores mayores al 70% del valor máximo.
            Z(i,j)=0;
        else
            Z(i,j)=Y(i,j);
        end
        if Z(i,j)~=0
            w=w+1;
            P(w,:)=[i,j];
        end
    end
end
end

Q(:,2)=P(:,2)+(t(1,1)-1); % Las coordenadas de los pixeles de la matriz seleccionada.
Q(:,1)=P(:,1)+(t(1,2)-1); % Las coordenadas de los pixeles de la matriz seleccionada.
Q=round(Q); % Coordenadas en filas y columnas del objeto seleccionado.
t1=size(Q);

tamaos(n,:)=t1(:,:); %% Matriz con los tamaños de cada Q.
if n==1
    Matriz_coordenadas(:,:)=Q(:,:);
else
    Matriz_coordenadas((tam_Matri_Acumu(1,1)+1):(t1(1,1)+tam_Matri_Acumu(1,1)),:)=Q(:,:);
    %matriz con todas las coordenadas.

```

```

end
tam_Matri_Acumu=size(Matriz_coordenadas);% acumula los tamaños.
n=n+1;
clear Q
end

tama_tamaos=size(tamaos);
Sumar=0;
for i=1:tama_tamaos(1,1)
    Sumar=tamaos(i,1)+Sumar;
    Sum_Tama(i,1)=Sumar; % Matriz con la suma de los tamaños.
end

for j=1:tama_tamaos(1,1)-1
Combinaciones(j,1)=tamaos(1,1)*tamaos(j+1,1); %Total combinaciones.
end

n=1;
cont=0;
clear Y

%genera las combinaciones de cada uno de los pixeles del objeto C1 (es al que
% se le desconocen las coordenadas) a cada uno de los pixeles del objeto Cn.

while n<=(Numero_Objetos-1) %numero de objetos seleccionados.
    for i=1:Combinaciones(n,1)
        Y(i,:)=Matriz_coordenadas(Sum_Tama(n,1)+1+cont,1); %Declinación Y.
        X(i,:)=Matriz_coordenadas(Sum_Tama(n,1)+1+cont,2); %Ascensión recta X.
        cont=cont+1;
        if cont==tamaos(n+1,1)
            cont=0;
        end
    end
end

```

```

    end
cont1=0;
cont2=0;
for j=1:Combinaciones(n,1)
    Y1(j,:)=Matriz_coordenadas((1+cont1),1);% pixeles primer objeto seleccionado.
    X1(j,:)=Matriz_coordenadas((1+cont1),2);
    cont2=cont2+1;
    if cont2==tamaos(n+1,1)
        cont1=cont1+1;
        cont2=0;
    end
end
end
Decli(:,1)=Y1; % Declinación primer objeto.
Decli(:,2)=Y;
Ascr(:,1)=X1; % Ascensión recata primer objeto.
Ascr(:,2)=X;
Matriz_Decli{n}=Decli;
Matriz_Ascr{n}=Ascr;
clear X X1 Decli Y Y1 Ascr
Distan_Decli=Matriz_Decli{n};%% Combinación en declinación.
Distan_Ascr=Matriz_Ascr{n}; %% Combinación en ascensión recta.
Matriz_Dis_Decli{n}=abs(Distan_Decli(:,1)-Distan_Decli(:,2)); %%Distancia en declinación.
Matriz_Dis_Ascr{n}=abs(Distan_Ascr(:,1)-Distan_Ascr(:,2)); %%Distancia en ascensión.
%Recta
Matriz_Dis_Decli_segarc{n}=Matriz_Dis_Decli{n}*Seg_Arc_Pixel;
Matriz_Dis_Ascr_segarc{n}=Matriz_Dis_Ascr{n}*Seg_Arc_Pixel;
Promedio_Dis_Decli_arcs(n,1)=mean(Matriz_Dis_Decli_segarc{n}); %promedios.
Promedio_Dis_Asrec_arcs(n,1)=mean(Matriz_Dis_Ascr_segarc{n});
Desv_Dis_Decli_arcs(n,1)=std(Matriz_Dis_Decli_segarc{n}); %Desviaciones.
Desv_Dis_Asrec_arcs(n,1)=std(Matriz_Dis_Ascr_segarc{n});
clear Distan_Ascr Distan_Decli
n=n+1;

```

```

end

Distancias_Arco_seg(:,1)=Promedio_Dis_Decli_arcs; %declinación.
Distancias_Arco_seg(:,2)=Promedio_Dis_Asrec_arcs; %ascensión recta.

%Toma los pixeles más al norte y al este de cada grupo para la comparación y si se suma o
%resta a las ascensiones rectas y declinaciones.

for i=1:tama_tamaos(1,1)
    if i==1
        Posicion_Imag(i,1)=Matriz_coordenadas(1,1); % La primera columna es la declinación.
        Posicion_Imag(i,2)=min(Matriz_coordenadas((1:tamaos(1,1)),2)); % La segunda columna es
        %la ascensión recta.
    else
        Posicion_Imag(i,1)=min(Matriz_coordenadas((Sum_Tama(i-1)+1):Sum_Tama(i,1),1));
        Posicion_Imag(i,2)=min(Matriz_coordenadas((Sum_Tama(i-1)+1):Sum_Tama(i,1),2));
    end
end

[coordenadas_seguarc]=posicion_Promedio(Posicion_Imag,Distancias_Arco_seg,matriz_ascens_ar
cosegu,matriz_decli_arcosegu);
coordenadas_seguarc1=coordenadas_seguarc; %copia de la variable.
[ascencion_rectal,declinacion1]=Conversor1(coordnadas_seguarc)
coordenadas_seguarc(:,1)=Desv_Dis_Decli_arcs;
coordenadas_seguarc(:,2)=Desv_Dis_Asrec_arcs;
[Desv_ascencion_rectal,Desv_declinacion1]=Conversor1(coordnadas_seguarc)

```

Anexo 3.2 posicion_Promedio

```

function[coordenadas_seguarc]=posicion_Promedio(Posicion_Imag,Distancias_Arco_seg,matriz_a
scens_arcosegu,matriz_decli_arcosegu)
a=size(Posicion_Imag);
for i=1:(a(1,1)-1)
    if Posicion_Imag(1,1)<Posicion_Imag(i+1,1)
        Coor_Decli(i,:)=matriz_decli_arcosegu(i,1)+Distancias_Arco_seg(i,1);
    elseif Posicion_Imag(1,1)>Posicion_Imag(i+1,1)
        Coor_Decli(i,:)=matriz_decli_arcosegu(i,1)-Distancias_Arco_seg(i,1);
    else
        Coor_Decli(i,:)=matriz_decli_arcosegu(i,1);
    end
    if Posicion_Imag(1,2)<Posicion_Imag(i+1,2)
        Coor_Asc(i,:)=matriz_ascens_arcosegu(i,:)+Distancias_Arco_seg(i,2);
    elseif Posicion_Imag(1,2)>Posicion_Imag(i+1,2)
        Coor_Asc(i,:)=matriz_ascens_arcosegu(i,:)-Distancias_Arco_seg(i,2);
    else
        Coor_Asc(i,:)=matriz_ascens_arcosegu(i,:);
    end
end
coordenadas_seguarc(:,2)=Coor_Asc;
coordenadas_seguarc(:,1)=Coor_Decli;

```

Anexo 3.3 Conversor1

```

function [ascension_rectal,declinacion1]=Conversor1(coordenadas_seguarc)
format('long')
a=size(coordenadas_seguarc);
for lr=1:a(1,1)

```

```

    coordenadas_seguarc(lr,2)=coordenadas_seguarc(lr,2)*(24/(3600*360));
    coordenadas_seguarc(lr,2)=coordenadas_seguarc(lr,2)*3600;
end
coordenadas_seguarc;
for i=1:a(1,1)
    segundos=coordenadas_seguarc(i,2); %halla la cantidad de minutos.
    w=0;
    while w*60<=segundos
        min=w;
        w=1+w;
    end
    seg=segundos-min*60;
if min>=60
    m=0;
    while m*60<=min
        horas=m; %halla la cantidad de horas.
        m=1+m;
    end
    min=min-horas*60;
else
    horas=0;
end
ascencion_rectal(i,:)=[horas min seg]; % coordenadas en declinación.
end
for i1=1:a(1,1)
    if coordenadas_seguarc(i1,1) < 0
        coordenadas_seguarc(i1,1)=abs(coordenadas_seguarc(i1,1))
        segundos2=coordenadas_seguarc(i1,1);
        in=0;
    while in*60<=segundos2
        min1=in; % halla los minutos de arco (angulares).
        in=in+1;
    end
end

```

```

end
seg1=segundos2-min1*60;
if min1>=60
    jn=0;
    while jn*60<=min1
        grad1=jn; % Halla los grados.
        jn=jn+1;
    end
    min1=min1-grad1*60;
else
    grad1=0;
end
declinacion1(i1,:)=[-grad1 min1 seg1]; %Declinación.
else
    segundos2=coordenadas_seguarc(i1,1);
in=0;
while in*60<=segundos2
    min1=in;
    in=in+1;
end
seg1=segundos2-min1*60;
if min1>=60
    jn=0;
    while jn*60<=min1
        grad1=jn;
        jn=jn+1;
    end
    min1=min1-grad1*60;
else
    grad1=0;
end
declinacion1(i1,:)=[grad1 min1 seg1]; %Declinación.

```

```
end
end
```

Anexo 4. Método del centroide para las coordenadas y el cálculo de las distancias.

Anexo 4.1 Centroide.

```
% El primer objeto a seleccionar ha de ser el desconocido.
% El orden de las matrices con las coordenadas así como el de los objetos a seleccionar
%debe de coincidir.

clear all
clc
Numero_Objetos=input('ingrese el numero de bojetos: ');
Seg_Arc_Pixel=input('ingrese segundos de arco por pixel: ');
Matriz_Declinacion=input('ingrese las declinaciones: ');
Tama_Matriz_Declinacion=size(Matriz_Declinacion);

if Tama_Matriz_Declinacion(1,1) >= Numero_Objetos
    error('Numero de coordenas incorrecto en Declinacion');
    break
end

Matriz_AscensionRecta=input('ingrese la Ascensión recta: ');
Tama_Matriz_Ascen=size(Matriz_AscensionRecta);

if Tama_Matriz_Ascen(1,1) >= Numero_Objetos
    error('Numero de coordenadas incorrecto en Ascensión Recta');
    break
end
```

```

end

a=imread('');
imshow(a);

a1=size(Matriz_Declinacion);
n=a1(1,1);
for i=1:n
    if Matriz_Declinacion(i,1)<0
        matriz_decli_arcosegu(i,:)=Matriz_Declinacion(i,1)*3600-Matriz_Declinacion(i,2)*60-
Matriz_Declinacion(i,3);%coordenas en arco segundos declinación.
    else

matriz_decli_arcosegu(i,:)=Matriz_Declinacion(i,1)*3600+Matriz_Declinacion(i,2)*60+Matriz_
Declinacion(i,3);%coordenas en arco segundos declinación.
    end

matriz_ascenc_horas(i,:)=Matriz_AscensionRecta(i,1)+Matriz_AscensionRecta(i,2)/60+Matriz_A
scensionRecta(i,3)/3600;%coordenasdas en horas para la ascensión recta.
end
matriz_ascens_arcosegu=matriz_ascenc_horas*54000; %ascensión recta en arco segundos.
matriz_decli_arcosegu;

n=1;

while n<=Numero_Objetos
pause %% sirve para que el usuario interactúe con la imagen. Para la selección
% se recomienda presionar Esc.
t=getrect;
Y=a((t(1,2):(t(1,2)+t(1,4))), (t(1,1):(t(1,1)+t(1,3))));
a1=size(Y);
w=0;

```

```

for i=1:a1(1,1);
    for j=1:a1(1,2);
        if Y(i,j) < .7*max(max(Y));
            Z(i,j)=0;
        else
            Z(i,j)=Y(i,j);
        end
        if Z(i,j)~=0
            w=w+1;
            P(w,:)=[i,j];
        end
    end
end
end
Q(:,2)=P(:,2)+(t(1,1)-1);
Q(:,1)=P(:,1)+(t(1,2)-1);
Q=round(Q);% Coordenadas en filas y columnas del objeto seleccionado.
centroide(n,:)=mean(Q); %matriz con los centroides.
Desviaciones(n,1)=std(Q(:,1));% y se mueve para las filas. La desviación en y.
Desviaciones(n,2)=std(Q(:,2));% x se mueve para las columnas. La desviación en x.
Desviaciones;% matriz con las desviaciones.
clear Z Q P
if n>1
    Distancias(n-1,1)=abs(centroide(1,1)-centroide(n,1));
    Distancias(n-1,2)=abs(centroide(1,2)-centroide(n,2));% Distancia en pixeles.

    Matriz_Desv_Dis(n-1,1)=Desviaciones(1,1)+Desviaciones(n,1);% y, declinación,
dispersión en el eje y.
    Matriz_Desv_Dis(n-1,2)=Desviaciones(1,2)+Desviaciones(n,2); %x, ascensión recta
dispersión en el eje x.
elseif Numero_Objetos==1
    disp('solo ha seleccionado un objeto');
    break
end

```

```

end

n=n+1;
end
Distancias_Arco_seg=Distancias*Seg_Arc_Pixel;% distancias en arco segundos.
Coordenadas_Seguarc=posicion_centro(centroide,Distancias_Arco_seg,matriz_decli_arcosegu,ma
triz_ascens_arcosegu);

Coordenadas_Seguarcl=Coordenadas_Seguarc; %% para que no se pierda la variable.
[ascencion_recta,declinacion]=Conversor(Coordenadas_Seguarc)

Coordenadas_Seguarc=Matriz_Desv_Dis;
[Desv_ascencion_recta,Desv_declinacion]=Conversor(Coordenadas_Seguarc)

```

Anexo 4.2 posicion_centro.

```

function [Coordenadas_Seguarc]=posicion_centro(centroide,Distancias_Arco_seg,matriz_ascens_
arcosegu,matriz_decli_arcosegu)
a=size(centroide);
for i=1:(a(1,1)-1)
    if centroide(1,1)<centroide(i+1,1)
        Coor_Decli(i,:)=matriz_decli_arcosegu(i,1)-Distancias_Arco_seg(i,1);
    elseif centroide(1,1)>centroide(i+1,1)
        Coor_Decli(i,:)=matriz_decli_arcosegu(i,1)+Distancias_Arco_seg(i,1);
    else
        Coor_Decli(i,:)=matriz_decli_arcosegu(i,1)
    end
    if centroide(1,2)<centroide(i+1,2)
        Coor_Asc(i,:)=matriz_ascens_arcosegu(i,1)-Distancias_Arco_seg(i,2);
    elseif centroide(1,2)>centroide(i+1,2)
        Coor_Asc(i,:)=matriz_ascens_arcosegu(i,1)+Distancias_Arco_seg(i,2);
    end
end

```

```

        else
            Coor_Asc(i,:)=matriz_ascens_arcosegu(i,1)
        end
    end
end
Coordenadas_Seguarc(:,1)=Coor_Asc;
Coordenadas_Seguarc(:,2)=Coor_Decli;

```

Anexo 4.3 Conversor.

```

function [ascencion_recta,declinacion]=Conversor(Coordenadas_Seguarc)
format('long')
a=size(Coordenadas_Seguarc);

for lr=1:a(1,1)
    Coordenadas_Seguarc(lr,2)=Coordenadas_Seguarc(lr,2)*(24/(3600*360));
    Coordenadas_Seguarc(lr,2)=Coordenadas_Seguarc(lr,2)*3600;
end
Coordenadas_Seguarc;
for i=1:a(1,1)
    segundos=Coordenadas_Seguarc(i,2);
    w=0;
    while w*60<=segundos
        min=w;
        w=1+w;
    end
    min;
    seg=segundos-min*60;
if min>=60
    m=0;
    while m*60<=min

```

```

        horas=m;
        m=1+m;
    end
    min=min-horas*60;
else
    horas=0;
end
ascencion_recta(i,:)=[horas min seg];
end

for i1=1:a(1,1)
    if Coordenadas_Seguarc(i1,1)<0
        Coordenadas_Seguarc(i1,1)=abs(Coordenadas_Seguarc(i1,1));
        segundos2=Coordenadas_Seguarc(i1,1);
        in=0;
        while in*60<=segundos2
            min1=in;
            in=in+1;
        end
        seg1=segundos2-min1*60;
        if min1>=60
            jn=0;
            while jn*60<=min1
                grad1=jn;
                jn=jn+1;
            end
            min1=min1-grad1*60;
        else
            grad1=0;
        end
        declinacion(i1,:)=[-grad1 min1 seg1];
    else

```

```

                segundos2=Coordenadas_Seguarc(i1,1);
in=0;
while in*60<=segundos2
    min1=in;
    in=in+1;
end
seg1=segundos2-min1*60;
if min1>=60
    jn=0;
    while jn*60<=min1
        grad1=jn;
        jn=jn+1;
    end
    min1=min1-grad1*60;
else
    grad1=0;
end
declinacion(i1,:)= [grad1 min1 seg1];
end
end

```

Anexo 5. Código para la estimación de la distribución del fondo de cielo.

```

clc
clear all
y6=fitsread('C:\Desktop \Final5.fit');
re=[590.0000 476.0000 912.0000 642.0000];
a=y6((re(1,2):(re(1,2)+re(1,4))), (re(1,1):(re(1,1)+re(1,3))));
tama_muestra=3000;
z=aleat_sin_rept(tama_muestra);
z=sort(z);

```

```

Matriz=Coor_Pixel1(z);
cont2=0;
for i=1:tama_muestra
    cont2=cont2+1;
    M_val_Pixel(cont2,1)=a(Matriz(i,1),Matriz(i,2));
    Aleatorios=z;
    while M_val_Pixel(cont2,1) > 450% el valor máximo que se espera como corriente de
        %oscuridad.

        tama_muestra=1; % Solo se genera un aleatorio.
        z1=aleat_sin_rept(tama_muestra); %genera un nuevo aleatorio.

        if (z1~=z)
            z=z1; % Se cambia el valor de "z" para que pueda ser leído por la función
                % "Coor_Pixel(z)".
            Matriz1=Coor_Pixel1(z); % Coordenadas del aleatorio generado.
            M_val_Pixel(cont2,1)=a(Matriz1(1,1),Matriz1(1,2));
            z=[Aleatorios;z];
            Aleatorios=z; %Acumula los aleatorios para que no se repitan.
        else
            M_val_Pixel(cont2,1);
        end
    end
end
end
Varianza=var(M_val_Pixel);
Promedio=mean(M_val_Pixel);
Error=Promedio*0.01;
Z=1.880793608;
No=((Varianza*Z^2)/Error)*(1+(2/3000)); %3000
    % de la muestra previa que hay que tomar.
N=No/(1+(No/587059));% 587059 es el total
    % de pixeles, tamaño de la población.

```

```

N=ceil(N);
tama_muestra=N;
z=aleat_sin_rept(tama_muestra); %Genera aleatorios tantos como tama_muestra.
z=sort(z); % Se ordenan los aleatorios de menos a mayor.
Matriz=Coor_Pixel1(z); % Transforma el aleatorio a coordenadas en la imagen.

clear M_val_Pixel
cont2=0;
for i=1:tama_muestra %% tamaño de la muestra.
    cont2=cont2+1;
    M_val_Pixel(cont2,1)=a(Matriz(i,1),Matriz(i,2)); % "M_val_Pixel" Es la matriz con
        %las coordenadas de los pixeles evaluadas.
    Aleatorios=z;

while M_val_Pixel(cont2,1) > 450 % Promedios(cont,1)*1.9% el valor máximo que se
    %espera como corriente de oscuridad HAY QUE ESTIMARLO.

    tama_muestra=1; % Solo se genera un aleatorio
    z1=aleat_sin_rept(tama_muestra); %genera un nuevo aleatorio.

    if (z1~=z)
        z=z1; % Se cambia el valor de "z" para que pueda ser leído por la función
            % "Coor_Pixel(z)".
        Matriz1=Coor_Pixel1(z); % Coordenadas del aleatorio generado.
        M_val_Pixel(cont2,1)=a(Matriz1(1,1),Matriz1(1,2));
        z=[Aleatorios;z];
        Aleatorios=z; %Acumula los aleatorios para que no se repitan.
    else
        M_val_Pixel(cont2,1);
    end
end
end

```

```
end
```

```
[Matriz_Chi,Numero_clases_matriz,Matriz_D]=prueba_bya(M_val_Pixel)
```

Anexo 6. Código para identificar el posible ajuste de los pixeles de la estrella.

```
clc
clear all
y5=fitsread('');
re=[944 697 182 161];
a1=y5((re(1,2):(re(1,2)+re(1,4))), (re(1,1):(re(1,1)+re(1,3))));
imshow(a1,[0 750])
%figure,imshow(y6,[])
a=size(a1);
cont=0;
for i=1:a(1,1);
    for j=1:a(1,2);
        if a1(i,j) > 10000;
            cont=cont+1;
            M_val_Pixel(cont,1)=a1(i,j);
        end
    end
end
end

M_val_Pixel=sort(M_val_Pixel);
[Matriz_Chi,Numero_clases_matriz,Matriz_D]=prueba_bya(M_val_Pixel)
```

Anexo 7. Resultados distribución todo el sensor

Anexo 7.1. Resultado *Chi* todo el sensor primera jornada.

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson
1	1790	1660	13850	1170	195120
2	1730	1700	14290	1110	241560
3	1350	1500	12270	1060	134880
4	1660	950	10440	650	113770
5	2070	1190	12180	960	110560
6	1430	1680	14170	1130	171150
7	1790	1140	15470	1400	262640
8	1980	1210	15970	1600	242370
9	1720	1070	12850	1270	226470
10	1890	1110	10570	890	228740
11	1680	1080	14310	1120	215340
12	1690	1070	14700	1370	215040
13	1860	1080	11100	790	113310
14	1380	880	13270	1250	198340
15	1580	1470	13130	1140	206530
Promedio	1701	1253	13238	1127	191721

Anexo 7.2. Número de clases de la prueba *Chi* todo el sensor primera jornada.

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson
1	10	11	10	12	8
2	10	11	10	12	8
3	9	10	9	11	7
4	10	10	9	11	7
5	10	10	9	11	7
6	10	11	10	12	8
7	10	10	10	12	8
8	10	10	10	12	8
9	10	10	9	12	8
10	10	10	9	11	8

11	10	10	10	12	8
12	10	10	10	12	8
13	10	10	9	11	7
14	10	10	10	12	8
15	10	11	10	12	8

Anexo 7.3. Resultados K-S todo el sensor. Primera jornada

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson	N° Datos
1	0.0245	0.0219	0.1171	0.0209	0.1485	71462
2	0.0226	0.0204	0.1111	0.0194	0.1479	70455
3	0.0228	0.0187	0.1182	0.0180	0.1462	65170
4	0.0220	0.0186	0.1051	0.0174	0.1443	62278
5	0.0228	0.0192	0.1226	0.0180	0.1440	63861
6	0.0215	0.0196	0.1203	0.0185	0.1460	72496
7	0.0211	0.0186	0.1184	0.0180	0.1463	74753
8	0.0225	0.0199	0.1197	0.0187	0.1448	71055
9	0.0237	0.0202	0.1193	0.0187	0.1440	69433
10	0.0223	0.0186	0.1049	0.0179	0.1418	65078
11	0.0227	0.0186	0.1118	0.0174	0.1450	70658
12	0.0234	0.0195	0.1215	0.0182	0.1460	70869
13	0.0248	0.0201	0.1098	0.0185	0.1464	64007
14	0.0202	0.0171	0.1180	0.0161	0.1430	66579
15	0.0233	0.0199	0.1130	0.0187	0.1473	69199

Anexo 7.4. Resultado *Chi* todo el sensor segunda jornada.

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson
1	4350	3120	32070	3960	748840
2	4000	2860	37950	4070	764200
3	4560	3400	37550	4090	760140
4	4720	3260	42090	4900	933680
5	3720	2730	35770	3950	671490
6	3220	2280	34240	3990	759570
7	4950	3800	33670	3900	637060
8	5180	4180	37450	3810	488950
9	4500	3160	35340	4330	805770
10	4360	3210	34560	3820	728880

11	4360	3200	34520	3780	667440
12	5240	4130	37230	4010	571200
13	4470	3360	33080	4340	745930
14	4060	3330	33380	3470	420530
15	3870	2870	34220	3580	635390
Promedio	4371	3259	35541	4000	689271

Anexo 7.5. Número de clases de la prueba *Chi* todo el sensor segunda jornada.

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson
1	8	8	8	10	6
2	8	8	9	10	6
3	8	8	9	10	6
4	8	8	9	10	6
5	8	8	9	10	6
6	8	8	9	10	6
7	8	8	8	10	6
8	8	8	9	10	6
9	8	8	8	10	6
10	8	8	8	10	6
11	8	8	8	10	6
12	8	8	9	10	6
13	8	8	8	10	6
14	8	8	9	10	6
15	8	8	9	10	6

Anexo 7.6. Resultados K-S todo el sensor. Segunda jornada

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson	N° Datos
1	0.037	0.0359	0.185	0.0352	0.1942	96196
2	0.0343	0.0337	0.1894	0.0333	0.1932	103160
3	0.0364	0.0345	0.186	0.0338	0.1942	103547
4	0.0355	0.0339	0.1977	0.0332	0.1937	106271
5	0.035	0.0333	0.1919	0.0326	0.1918	96614
6	0.0354	0.0338	0.189	0.0331	0.1929	95368
7	0.0382	0.0364	0.1805	0.0356	0.1913	103974

8	0.0367	0.0353	0.1825	0.0344	0.1923	108957
9	0.0368	0.035	0.1854	0.0341	0.1923	104041
10	0.0361	0.0348	0.1835	0.0341	0.1932	102626
11	0.0355	0.034	0.1855	0.0333	0.1913	100183
12	0.0373	0.0362	0.1824	0.0357	0.1943	104338
13	0.0378	0.0365	0.1802	0.0358	0.1948	102717
14	0.0366	0.0342	0.1845	0.033	0.1909	98321
15	0.0368	0.0349	0.1885	0.0342	0.1924	93530

Anexo 7.7. Resultado *Chi* todo el sensor tercera jornada.

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson
1	1988	1592	44272	1839	18194
2	1840	1266	42095	3442	18827
3	1468	1053	44893	2570	21200
4	1345	953	42786	2241	20240
5	1812	1447	45926	2227	20835
6	1572	1184	48802	2378	21656
7	853	604	36991	2109	20366
8	1270	877	46276	2433	20867
9	1193	785	44776	2476	19789
10	1108	705	39520	1442	21127
11	1332	854	37694	1396	20971
12	1252	768	35012	1591	21850
13	1248	931	41636	2172	21458
14	1457	1077	44839	2198	21056
15	2141	1772	41407	1734	18802
Promedio	1459	1058	42462	2150	20483

Anexo 7.8. Número de clases de la prueba *Chi* todo el sensor tercera jornada.

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson
1	6	6	8	8	5
2	6	6	7	8	5
3	6	6	8	8	5
4	6	6	8	8	5

5	6	6	8	8	5
6	6	6	8	8	5
7	6	6	7	8	5
8	6	6	8	8	5
9	6	6	8	8	5
10	6	6	7	7	5
11	6	6	7	7	5
12	6	6	7	7	5
13	6	6	8	8	5
14	6	6	8	8	5
15	6	6	7	8	5

Anexo 7.9. Resultados K-S todo el sensor. Tercera jornada

N° de Simulación	Normal	Gamma	Weibull	Log Normal	Poisson	N° Datos
1	0.0274	0.0243	0.2596	0.0245	0.1132	65910
2	0.0298	0.0231	0.2558	0.0238	0.1141	63602
3	0.0252	0.0249	0.2502	0.0248	0.1117	70448
4	0.0263	0.0242	0.2502	0.0241	0.1116	66794
5	0.0263	0.0261	0.251	0.0262	0.1118	73148
6	0.0256	0.0247	0.2533	0.0254	0.1125	76136
7	0.026	0.0253	0.2428	0.0252	0.1098	66202
8	0.0265	0.0255	0.2549	0.0252	0.1109	71388
9	0.0272	0.0236	0.2549	0.0241	0.1114	67603
10	0.0256	0.0242	0.2543	0.0243	0.1121	62370
11	0.0257	0.0255	0.2442	0.0258	0.112	65487
12	0.0261	0.026	0.237	0.026	0.1128	65008
13	0.0266	0.0274	0.2476	0.0277	0.1154	68783
14	0.0257	0.0245	0.2511	0.0244	0.1128	70887
15	0.0255	0.0253	0.2429	0.0257	0.1116	72130

Anexo 8. Resultados distribución todo el sensor

Anexo 8.1. Resultado *Chi* pixeles particulares primera jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	5.0869	1.3773	2.0013	14.594	2.6051	3.7403	5.9169	1.1055	20.6792	6.0686
Gamma	4.6167	1.1983	1.7657	14.467	2.3712	3.7934	5.405	0.835	19.7883	5.755
Weibull	15.3683	8.7505	8.4612	20.0507	9.3581	5.9092	17.1458	9.7566	38.5301	15.0778
Log Normal	5.7733	0.5103	1.9815	9.7368	1.9631	3.3971	5.2508	2.7648	14.337	2.2071
Poisson	6.4765	6.3202	2.6982	17.9643	3.9555	6.2037	7.4288	1.8402	25.8823	9.9073

Anexo 8.2. Número de clases *Chi* pixeles particulares primera jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	6	5	6	6	6	6	5	5	6	6
Gamma	6	5	6	6	6	6	5	5	6	6
Weibull	6	5	6	6	6	6	6	5	6	6
Log Normal	6	5	6	6	6	6	6	5	6	6
Poisson	6	6	6	6	6	6	5	4	6	7

Anexo 8.3. Resultado *K-S* pixeles particulares primera jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	0.1265	0.0761	0.0832	0.1046	0.1253	0.1074	0.1143	0.1062	0.1815	0.0995

Gamma	0.1238	0.0743	0.0815	0.1027	0.1221	0.1071	0.1092	0.1008	0.1808	0.098
Weibull	0.1553	0.123	0.1182	0.1402	0.1541	0.1084	0.1599	0.1427	0.1868	0.1175
Log Normal	0.1213	0.0721	0.0789	0.1007	0.1198	0.1044	0.1074	0.0985	0.1786	0.0956
Poisson	0.1226	0.086	0.0791	0.1191	0.1173	0.1239	0.1138	0.1224	0.1633	0.1081

Anexo 8.4. Resultado *Chi* pixeles particulares para la segunda jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	3.777	0.5323	0.4663	1.7457	2.2365	1.3354	1.7057	0.4724	4.9816	4.0232
Gamma	3.6261	0.4545	0.4105	1.5981	2.1654	1.4676	1.4556	0.6196	4.7152	3.8654
Weibull	12.7254	4.5742	5.7664	6.9245	7.1293	2.6185	10.4316	0.9515	14.4129	10.2826
Log Normal	1.3262	0.4136	0.6569	1.6934	0.7046	2.1213	1.8367	0.4062	5.4926	4.1645
Poisson	35.2745	26.1513	33.5275	19.4267	14.7491	7.8907	10.8812	1.5136	9.2069	5.5653

Anexo 8.5. Número de clases *Chi* pixeles particulares para la segunda jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	5	6	5	6	6	5	6	6	6	6
Gamma	5	6	5	6	6	5	6	6	6	6
Weibull	5	6	5	6	6	5	6	6	6	6
Log Normal	5	6	5	6	6	5	6	6	6	6
Poisson	7	7	7	7	7	6	7	7	6	6

Anexo 8.6. Resultados K-S pixeles particulares para la segunda jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	0.0949	0.0707	0.0629	0.0727	0.0677	0.0593	0.0924	0.0808	0.1199	0.1073
Gamma	0.0946	0.0729	0.0598	0.0698	0.0668	0.0562	0.0911	0.0799	0.1165	0.1034
Weibull	0.1266	0.0892	0.1234	0.1258	0.0941	0.1128	0.1096	0.097	0.16	0.1595
Log Normal	0.0928	0.0742	0.0605	0.0708	0.0645	0.0584	0.089	0.0775	0.1151	0.1021
Poisson	0.2161	0.1706	0.2103	0.1582	0.1479	0.1265	0.1248	0.0864	0.1144	0.1043

Anexo 8.7. Resultado Chi pixeles particulares para la tercera jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	3.6363	13.4972	0.3646	2.4862	0.9871	0.1502	3.1808	3.474	2.1104	0.6948
Gamma	3.5882	14.6365	0.3986	2.3973	1.1048	0.1442	3.2418	3.3324	2.097	0.583
Weibull	9.2792	3.248	1.1941	7.355	1.9859	2.4226	6.8473	9.6073	5.9079	8.682
Log Normal	1.4044	15.9832	2.958	0.9355	1.4602	1.0327	3.146	2.201	0.9325	0.3642
Poisson	27.858	26.8081	30.8402	20.7181	31.2074	11.5558	13.205	38.9318	13.3735	26.1422

Anexo 8.8. Número de clases Chi pixeles particulares para la tercera jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	6	7	6	6	6	6	7	6	6	5
Gamma	6	7	6	6	6	6	7	6	6	5
Weibull	6	7	6	6	6	6	7	6	6	5

Log Normal	6	7	7	6	6	6	7	6	6	5
Poisson	7	7	7	7	7	7	7	7	7	7

Anexo 8.9. Resultados K-S pixeles particulares para la tercera jornada.

	Pixel 8898	Pixel 162117	Pixel 204774	Pixel 498952	Pixel 767583	Pixel 1487829	Pixel 1536132	Pixel 1569223	Pixel 1667894	Pixel 1846845
Normal	0.0887	0.1173	0.1019	0.0727	0.1158	0.0577	0.0677	0.1231	0.0451	0.0795
Gamma	0.0889	0.1184	0.1045	0.0695	0.118	0.0579	0.0697	0.1205	0.0484	0.0766
Weibull	0.0995	0.0732	0.0765	0.1297	0.0938	0.0977	0.0717	0.1746	0.0937	0.1362
Log Normal	0.0866	0.121	0.1043	0.0693	0.1191	0.0601	0.0677	0.1204	0.049	0.0782
Poisson	0.1889	0.1982	0.2053	0.154	0.1907	0.1404	0.1293	0.1971	0.1224	0.2072

Anexo 9. Coordenadas obtenidas mediante las metodologías de análisis de Clúster.

Anexo 9.1. Coordenadas en declinación implementando el método del promedio ponderado.

N° Objeto	Grados	Minutos	Segundos	Desviación
2	-1	53	55.97398	2.3154601
3	-1	54	13.3190982	2.87012001
4	-1	54	20.8605657	3.36975023
5	-1	53	30.9523486	3.27311561

Anexo 9.2. Coordenadas en ascensión recta implementando el método del promedio ponderado.

N° Objeto	Horas	Minutos	Segundos	desviación
2	5	40	41.975664	0.2927166
3	5	40	43.3254461	0.22645771
4	5	40	40.1897611	0.27634138
5	5	40	43.4368887	0.2485226

Anexo 9.3. Coordenadas en declinación implementando el método del centroide.

N° Objeto	Grados	Minutos	Segundos	Desviación
2	-1	56	15.9017242	5.40961052
3	-1	56	37.87376	5.24292231
4	-1	53	32.16352	5.02325946
5	-1	58	30.22112	4.44318244

Anexo 9.4. Coordenadas en ascensión recta implementando el método del centroide.

N° Objeto	Horas	Minutos	Segundos	Desviación
2	5	40	32.6444012	0.45272485
3	5	40	33.686048	0.44476504
4	5	40	43.403584	0.44958987
5	5	40	23.495776	0.43052599

Anexo 10. Distribución fondo de cielo.

Anexo 10.1. Valores del estadístico *Chi* para el fondo de cielo.

N° Simulación	Normal	Gamma	Weibull	LogNor	Poisson
1	11.9358	8.513	71.4253	23.2562	22.5007
2	5.2198	4.2457	49.3882	7.5637	37.1852
3	4.8065	4.2475	53.0808	3.0413	5.7527
4	8.7389	10.2876	39.9523	16.8946	20.1739
5	6.41	5.0699	51.9303	4.6725	13.8467
6	17.5404	14.9737	66.372	7.1337	30.0276
7	18.3882	13.8473	88.8487	9.3398	25.6642
8	8.1292	8.4902	59.6219	10.8371	18.0136
9	9.4881	8.6591	58.5493	12.459	10.196
10	22.8165	16.6023	66.3155	19.9843	43.3233
Promedio	11.34734	9.49363	60.54843	11.51822	22.66839

Anexo 10.2. Número de clases para los estadísticos *Chi* del fondo de cielo.

N° Simulación	Normal	Gamma	Weibull	LogNormal	Poisson
1	9	9	9	10	9
2	9	10	9	10	9
3	9	10	9	10	9
4	9	9	9	9	9
5	9	9	9	9	9
6	9	9	9	10	9

7	9	9	9	9	9
8	9	10	9	10	9
9	9	9	9	10	9
10	10	10	9	10	9

Anexo 10.3. Valores del estadístico K-S para el fondo de cielo.

N° Simulación	Normal	Gamma	Weibull	LogNormal	Poisson	Numero de datos
1	0.0599	0.0524	0.1091	0.0489	0.0352	444
2	0.0451	0.0379	0.0957	0.0347	0.0529	439
3	0.0338	0.0295	0.0828	0.0285	0.0285	474
4	0.0479	0.0552	0.0821	0.0588	0.0627	439
5	0.0538	0.0471	0.104	0.0441	0.0308	432
6	0.0489	0.0414	0.0971	0.0378	0.0407	455
7	0.0593	0.0524	0.1112	0.0493	0.0416	456
8	0.043	0.0371	0.0956	0.0345	0.0429	440
9	0.0321	0.0341	0.0852	0.0357	0.0293	453
10	0.0517	0.0466	0.0931	0.0437	0.0684	449

Anexo 11. Estadísticos calculados para la distribución de la estrella.

Anexo 11.1. Valores del estadístico Chi para la distribución de la estrella.

	Normal	Gamma	Weibull	LogNormal	Poisson
Chi calculado	1158.1	1065.7	1277.6	914.1	Inf
numero de clases	12	12	12	12	8

Anexo 11.2. Valores del estadístico K-S para la distribución de la estrella.

Estaditico K-S	0.2669	0.2678	0.2724	0.2645	0.5725
numero de datos	1846				