



Tecnológico de Costa Rica

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

PROGRAMA DE MAESTRÍA EN COMPUTACIÓN

Modelo de sensibilidad de los parámetros de frecuencia de la Transformada de Trazo para la clasificación de imágenes aéreas

Tesis

Presentado en cumplimiento parcial de los requisitos para
el grado de Magister Scientiæ en Ingeniería en
Computación con énfasis en Ciencias de la Computación

AUTOR:

Ricardo Román Brenes, Bach.

PROFESOR CONSEJERO:

Francisco J. Torres Rojas, Ph.D.

Junio 2015

Resumen

Las imágenes aéreas o satélites se puede utilizar para producir mapas de cobertura de terreno, que son un herramienta útil e importante para tomadores de decisiones en varios campos de trabajo, incluyendo biodiversidad, telecomunicaciones y gestión de desastres naturales entre otros.

El método de la Transformada de Trazo puede usarse para procesar estas imágenes. Este método extrae características de las imágenes al aplicar una serie de funcionales de manera sucesiva para producir una representación numérica que se evalúa en la clasificación más adelante. Este modelo depende de varios factores para alcanzar una operación eficiente, entre ellos, los parámetros de frecuencia de las trazas, el tipo de clasificador utilizado y los tipos de cobertura de terreno a evaluar.

Es el propósito de esta investigación producir un modelo de sensibilidad a estos parámetros para la clasificación de texturas en imágenes aéreas utilizando la Transformada de Trazo.

La experimentación sobre el tiempo de extracción y precisión de clasificación revelo que los parámetros de frecuencia, en especial el parámetro de selección de píxeles, $\Delta\tau$ y el clasificador utilizado tiene un gran impacto sobre ambas medidas.

Palabras clave: Transformada de trazo, característica triple, procesamiento de imágenes, tiempo de extracción, precisión de clasificación, imágenes aéreas.

Abstract

Aerial or satellital images can be used to produce terrain coverage maps, which in time are a very useful and important tool for decision-making people in several fields including biodiversity, telecommunications and natural disaster management.

The Trace Transform method can be used to process these images. This method extracts features from the images by applying a series of functionals to produce a numeric representation that will be used for classification later on. This model depends on several factors in order to have an efficient operation, among them, the frequency parameters of the traces, classifier type and land coverage.

The purpose of this research is to create a model of sensibility to the frequency parameters of the Trace Transform for the classification of textures in aerial images.

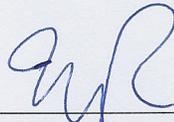
Experimentation on feature extraction time and precision rate revealed that the frequency parameters, specially the pixel selection parameter, $\Delta\tau$, and the classifier type have a big effect both of them.

Keywords: Trace transform, triple feature, image processing, extraction time, classification precision, aerial image.

APROBACIÓN DE LA TESIS

“Modelo de sensibilidad de los parámetros de frecuencia de la Transformada de Trazo para la clasificación de imágenes aéreas”

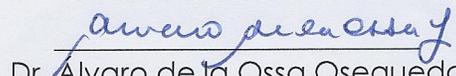
TRIBUNAL EXAMINADOR



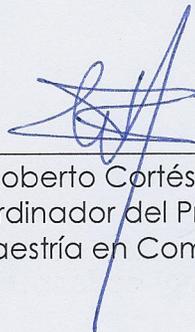
PhD. Francisco Torres Rojas
Profesor Asesor



M.Sc. Eddy Ramírez Jiménez
Profesor Lector



Dr. Alvaro de la Ossa Osegueda
Profesor Externo



Dr. Roberto Cortés Morales
Coordinador del Programa
de Maestría en Computación

Junio, 2015

AGRADECIMIENTOS

Quisiera agradecer a mis compañeros en el CNCA/CeNAT por su ayuda y consejo, en especial a Renato Garita y Álvaro de la Ossa.

A mi padre y madre por su compañía a lo largo de los años y no solo de esta investigación.

A mi tutor Francisco Torres por la oportunidad y la guía; a Andrés Messeguer y a Rodolfo Mora por su ayuda en las correcciones.

Al personal de la USES en la UCR, particularmente a María Isabel González, que sin su ayuda aún estaríamos buscando salidas.

También agradecer al programa de becas CeNAT, que contribuyó a financiar parte de esta investigación.

ÍNDICE GENERAL

Acrónimos	XIII
1. Introducción y antecedentes	1
1.1. Introducción	1
1.2. Transformada de trazo	3
1.2.1. Complejidad de la Transformada de Trazo	6
1.3. Planteamiento del problema	7
1.4. Hipótesis	8
1.5. Trabajo relacionado	8
2. Objetivos y aportes	11
2.1. Objetivo general	11
2.2. Objetivos específicos	11
2.3. Alcances y limitaciones	12
3. Experimentación	13
3.1. Experimento A: Tiempo de ejecución de la extracción de características	14
3.1.1. Conjunto de imágenes	14
3.1.2. Funcionales	15
3.1.3. Implementación del algoritmo de la TT	19
3.2. Experimento B: Precisión de clasificación	24
3.2.1. Cálculo de Precisión	26
3.2.2. Clasificadores	26
3.2.2.1. K-Means	26
3.2.2.2. Máquina de Soporte Vectorial	27
3.2.2.3. Regresión	27
3.2.2.4. Redes bayesianas	28

4. Resultados y análisis	29
4.1. Experimento A	31
4.1.1. Verificación de supuestos	31
4.1.1.1. Transformaciones	34
4.1.1.2. Modelo Lineal Generalizado	38
4.1.1.3. Método Box-Cox	38
4.1.1.4. Transformación por Rangos	41
4.1.2. Análisis de resultados	42
4.2. Experimento B	48
4.2.1. Verificación de supuestos	48
4.2.1.1. Transformaciones	50
4.2.1.2. Modelo Lineal Generalizado	54
4.2.1.3. Método Box-Cox	54
4.2.1.4. Transformación por Rangos	56
4.2.2. Análisis de Resultados	56
5. Conclusiones y trabajo futuro	61
5.1. Aportes	63
5.2. Trabajo futuro	63
Bibliografía	65

ÍNDICE DE CUADROS

3.1. Ejemplo de tipos de cobertura. Cada columna presenta dos imágenes de un tipo de cobertura distinto, respectivamente: agua, bosque, rural, sembradío y urbano.	15
3.2. Funcionales de trazo utilizadas en la extracción de características. . . .	17
3.3. Funcionales diamétricas utilizadas en la extracción de características. . .	18
3.4. Funcionales de circo utilizadas en la extracción de características. . . .	19
3.5. Salida de la implementación original de la TT. Cada celda en la tabla representa una TT, donde T_n indica la funcional de trazo, D_m la funcional diamétrica y C_p la funcional de circo usadas para generarla.	23
3.6. Salida de la implementación modificada de la TT. Cada celda en la línea representa una TT, donde T_n indica la funcional de trazo, D_m la funcional diamétrica y C_p la funcional de circo usadas para generarla.	23
4.1. Grupos estudiados en el experimento A.	29
4.2. Grupos estudiados en el experimento B.	30
4.3. Tabla ANOVA para los rangos de tiempo del experimento A.	42
4.4. Tabla ANOVA para la precisión en la clasificación.	57

ÍNDICE DE FIGURAS

1.1.	Parametros de la TT. En la figura se presenta desde el punto de origen O , la rotación ϕ , la distancia desde el origen ρ y el trazo τ tangente al circulo de radio ρ	4
1.2.	Representación gráfica del proceso de la Transformada de Trazo.	5
1.3.	Complejidad del algoritmo de la TT.	6
3.1.	Ciclo original, utilizando $\Delta\phi = 1$	21
3.2.	Ciclo modificado, utilizando $\Delta\phi = 3$	22
3.3.	Fragmento de la salida de un clasificador SMO en Weka. Se pueden identificar datos importantes como el número y porcentaje de instancias clasificadas correcta e incorrectamente, la matriz de confusión y un bloque detallando la precisión por categoria.	25
3.4.	Técnica de Validación Cruzada de 4 iteraciones. El conjunto de datos está compuesto por ítemes de los grupos A y B. En cada iteración, el grupo de elementos en azul se usa para validación y el grupo en verde para entrenamiento. Estos grupos van rotando tantas veces como iteraciones se dispongan.	28
4.1.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo sin transformaciones.	32
4.2.	Prueba de Lilliefors para normalidad en R. El estadístico es mayor que el valor-p por lo que los datos no son normales.	33
4.3.	Prueba de Levene para homogeneidad de varianzas en R. El estadístico es mayor que el valor-p por lo que los datos no tiene varianzas homogéneas.	33
4.4.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo transformados con una potencia e	35
4.5.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo transformados con una logaritmo base 10.	36

4.6.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo transformados con una raíz cuadrada.	37
4.7.	Método Box-Cox. El punto óptimo λ , con un 95 % de confianza se encuentra cerca de -1.	39
4.8.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo transformados con el método Box-Cox.	40
4.9.	Transformación por rangos de los datos de tiempo tomados en el experimento A.	41
4.10.	Invocación de la función aov para Análisis de Varianza en R. El parámetro recibido muestra las relaciones entre las variables: $rank(t)$ la variable de respuesta se debe analizar en función de $\Delta\tau$, $\Delta\rho$, $\Delta\phi$, tipo de cobertura y todas sus combinaciones.	42
4.11.	Promedio del efecto de los factores principales del experimento A. . .	43
4.12.	Promedio de interacciones de segundo nivel en el experimento A. . .	45
4.13.	Promedio de la interacción de tercer nivel para el experimento A. . .	47
4.14.	Prueba de Lilliefors para normalidad en R. El estadístico es mayor que el valor-p por lo que los datos no son normales.	48
4.15.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión.	49
4.16.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión transformados con una potencia e	51
4.17.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión transformados con una logaritmo base 10. . . .	52
4.18.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión transformados con una raíz cuadrada.	53
4.19.	Método Box-Cox. El punto óptimo λ , con un 95 % de confianza se encuentra cerca de 2.	54
4.20.	Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión transformados con el método Box-Cox.	55
4.21.	Transformación por rangos de los datos de precisión tomados en el experimento B.	56
4.22.	Promedio del efecto de los factores $\Delta\tau$ y clasificador sobre la precisión.	58
4.23.	Promedio de la interacción de segundo nivel entre $\Delta\tau$ y el clasificador para la precisión en el experimento B.	59
4.24.	Promedio del efecto de los factores $\Delta\tau$ y clasificador sobre la precisión.	60

ACRÓNIMOS

TT Transformada de trazo (siglas en inglés)
TF Característica Triple (siglas en inglés)
CeNAT Centro Nacional de Alta Tecnología
CNCA Colaboratorio Nacional de Computación Avanzada
PRIAS Programa de Investigaciones Aerotransportadas y Sensores Remotos
SVM Máquina de Soporte Vectorial (siglas en inglés)
SMO Optimización Secuencial Mínima (siglas en inglés)
Q-Q Cuantiles contra Cuantiles
RvVA Residuos contra Valores Ajustados
ANOVA Análisis de varianza

CAPÍTULO 1

INTRODUCCIÓN Y ANTECEDENTES

1.1. Introducción

Actualmente en el Programa de Investigaciones Aerotransportadas y Sensores Remotos (PRIAS), uno de los laboratorios del Centro Nacional de Alta Tecnología (CeNAT) se crean mapas de cobertura de terreno utilizando imágenes aéreas o satelitales[1].

Los mapas de cobertura de terreno son una herramienta importante para los tomadores de decisiones, políticos, administrativos o directores en áreas como control de riesgo, gestión de desastres naturales o biodiversidad [1], entre otros. Estos productos muestran de forma gráfica la distribución de algún atributo sobre un mapa [2].

Para construir dichos mapas se realiza un proceso en el cual el geógrafo toma la imagen y sobre ella marca polígonos según el tipo de terreno que sea. Este es un trabajo monótono y extenso, que se ha automatizado parcialmente con dos familias de

métodos, los supervisados y los no-supervisados[3]. El PRIAS cuenta con alrededor de 13000 imágenes aéreas o satelitales sin procesar, provenientes de varios proyectos [4] [5].

El método de la Transformada de Trazo (TT), que dentro de los métodos automatizados cabe en los supervisados, [6] ha podido clasificar con éxito diferentes tipos de imágenes [7] [8] [9]. Este método extrae características asociadas a la imagen utilizando diferentes funcionales, uno tras el otro, para lograr una reducción en la dimensionalidad. Esta versión reducida de la imagen se conoce como característica triple (*Triple Feature* o TF, por sus siglas en inglés) y es utilizada para la clasificación.

El Laboratorio Nacional de Computación Avanzada (CNCA), también en el CeNAT, ha creado un prototipo de la TT y se han hecho pruebas de clasificación con datos experimentales y se está preparando para hacer clasificaciones con datos reales [10].

La TT puede tener un tiempo de ejecución considerable, dependiendo de varios factores que se explicarán más adelante. Con imágenes de 5000×5000 píxeles se ha tomado en promedio 3,7 horas [10]. Para lograr una buena clasificación es necesario emplear varios funcionales y los mapas de cobertura utilizan imágenes de más de 4000×4000 píxeles [5].

Debido a esto es necesario que la TT utilice parámetros que le permitan realizar su trabajo de forma eficiente. En esta tesis se explora el comportamiento de estos parámetros sobre imágenes aéreas.

En el resto del capítulo 1, se detalla el funcionamiento de la TT y su complejidad computacional, el planteamiento del problema, la hipótesis de investigación y una revisión bibliográfica del trabajo relacionado. En el capítulo 2 se enumeran los objetivos, general y específicos de la investigación, así como los alcances y limitaciones. En el capítulo 3 se explican los dos experimentos que se desarrollaron para el estudio. En el capítulo 4, se encuentran los resultados y el análisis de los mismos. Por último

las conclusiones y el trabajo futuro aparecen en el capítulo 5.

1.2. Transformada de trazo

La TT es un método de extracción de características para su posterior clasificación. Este método se basa en trazar líneas sobre la imagen, de dónde se tomarán datos para ir transformándola a una versión más sencilla pero con igual poder de representación.

Una descripción detallada del algoritmo de extracción de características puede verse en [11], mas aquí se mencionarán los pasos más relevantes.

Es posible recorrer toda una imagen I si se trazan líneas sobre toda la superficie de la imagen. Estas líneas se definen con 2 parámetros: ρ y ϕ . Sea o el centro de I . A partir de o se define una distancia ρ y un ángulo ϕ . Se traza entonces una recta τ , tangente al círculo de radio ρ en el ángulo ϕ , como se muestran en la figura 1.1.

El modelo también cuenta con 3 parámetros de frecuencia:

- $\Delta\rho$: define cada cuántos píxeles se traza una línea. Si $\Delta\rho$ tiene un valor de 1 píxel se trazarán líneas cada píxel de la imagen, con un valor de 2 píxeles, cada 2 píxeles y así sucesivamente.
- $\Delta\phi$: determina cada cuántos grados se trazará una línea. Con un $\Delta\phi$ de 1° se trazarán líneas cada grado para un total de 360, con un valor de 90° , cada ángulo recto para un total de 4 líneas.
- $\Delta\tau$: fija, sobre la línea τ , cuales píxeles se tomarán en cuenta para el procesamiento. Si $\Delta\tau$ se carga con un valor de 1 píxel, se utilizarán todos los píxeles de la línea, si tiene un valor de 2 píxeles se tomará una muestra cada 2 y así sucesivamente.

Los píxeles que son intersecados por estas líneas τ y que cumplan con el parámetro $\Delta\tau$ serán la entrada para generar la primera representación de I , aplicando el

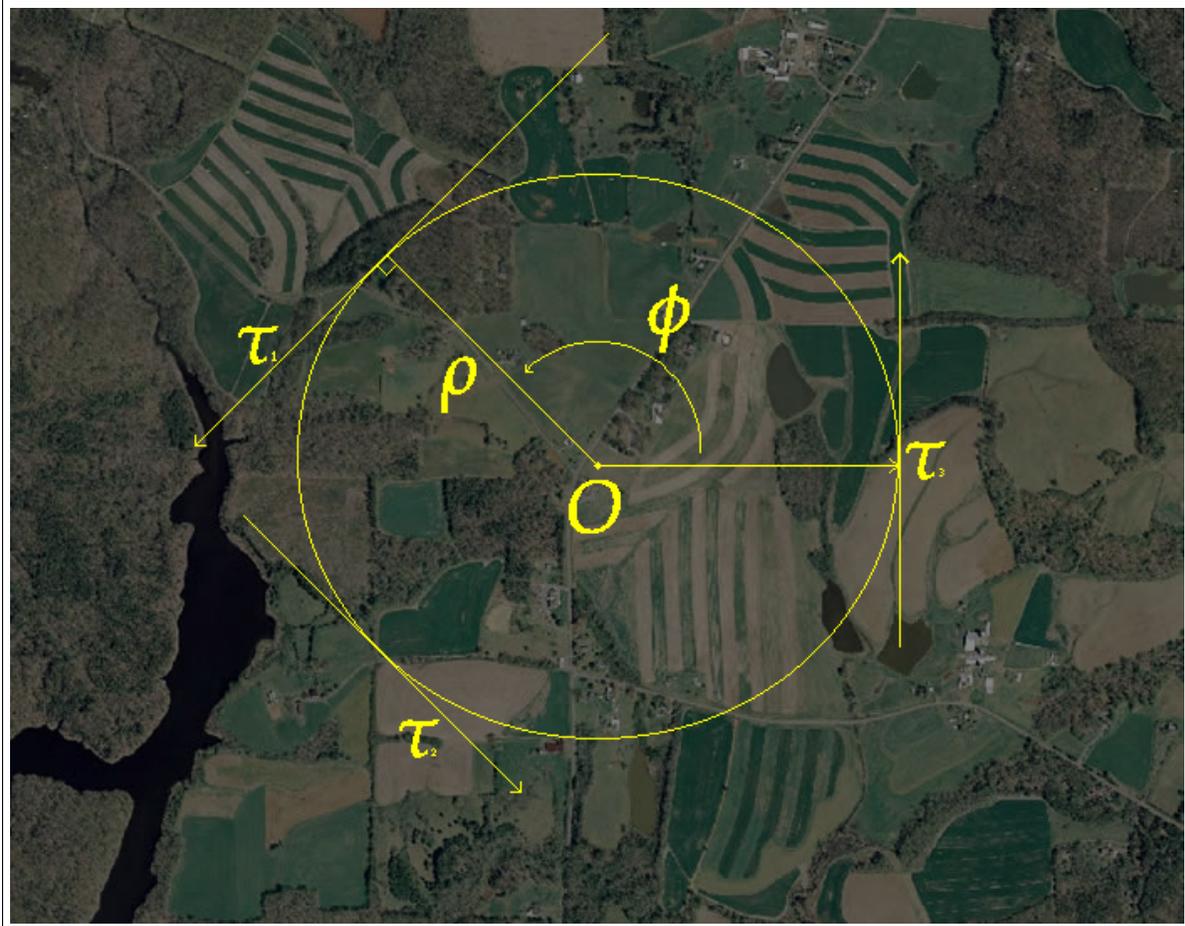


Figura 1.1: Parametros de la TT. En la figura se presenta desde el punto de origen O , la rotación ϕ , la distancia desde el origen ρ y el trazo τ tangente al circulo de radio ρ .

funcional T o de trazo. El resultado será un arreglo de dos dimensiones (que puede ser interpretado como una imagen 2D), I_T , definida en parámetros de ϕ y ρ . El siguiente paso es tomar cada columna de I_t y usarlas como entrada para el funcional P o diamétrico. Esto va a generar un arreglo I_p , que es una función del parámetro ϕ . A este arreglo se le aplica el funcional Φ o de circo, para dar como resultado un sólo número I_ϕ o característica triple. Este proceso se puede ver gráficamente en al figura 1.2.

Ahora bien, se da el caso en que no sólo es un funcional T, sino una serie, que se pueden indizar en una tabla de funcionales T. Esto produciría múltiples I_T que se pueden denominar I_T^τ , donde τ es el índice del funcional en la tabla de funcionales

T. Esto puede ocurrir y es deseable que ocurra con las otras dos funcionales, ya que esto permite la creación de vectores de TF que puede ser más representativos de I que una sola TF.

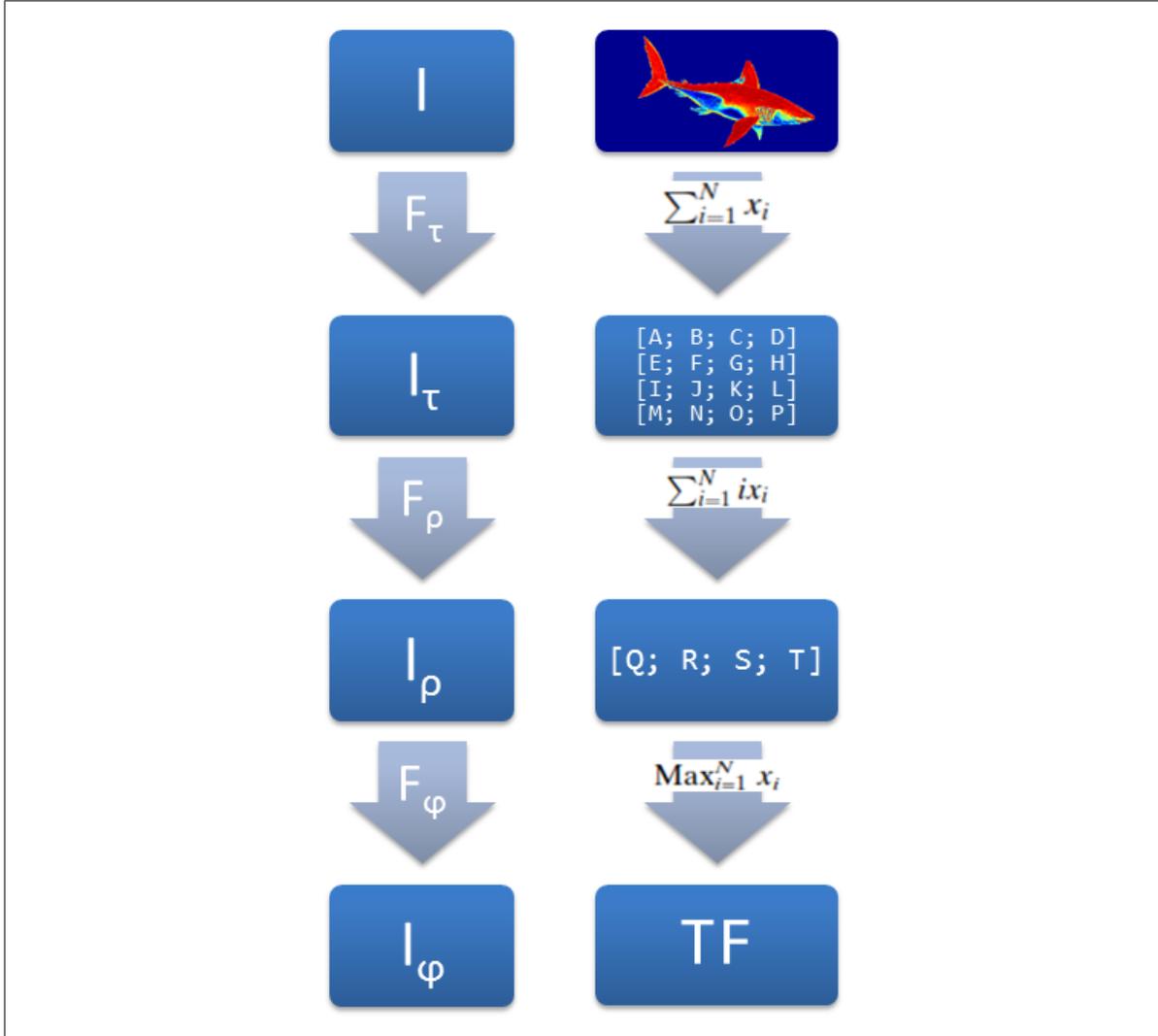


Figura 1.2: Representación gráfica del proceso de la Transformada de Trazo.

Una vez obtenidas las TF para las imágenes originales, estos vectores de TF deben ser clasificados. Para esto es posible utilizar cualquier clasificador computacional, ya que al obtener una representación vectorial de la imagen se independiza del dominio original. Clasificadores como redes neuronales artificiales, máquinas de soporte vectorial, agrupaciones, regresiones, entre otros, son todas técnicas válidas para realizar este paso[12]. Como con cualquier clasificador, es necesario hacer una partición

del conjunto original de datos; un subconjunto para el entrenamiento y otro para la clasificación.

1.2.1. Complejidad de la Transformada de Trazo

$$TF_{N_\phi N_\rho N_\tau} = \frac{C_T N_T n_\tau n_\rho n_\phi}{2} + C_P N_P N_T n_\rho n_\phi + C_\phi N_\phi N_P N_T n_\phi$$

Figura 1.3: Complejidad del algoritmo de la TT.

La complejidad del algoritmo de la TT está dada por la ecuación de la figura 1.3, donde:

- $TF_{N_\phi N_\rho N_\tau}$: Total de características triples generadas.
- C_T : cantidad de operaciones para la funcional de trazo hecha sobre cada muestra.
- C_P : cantidad de operaciones para la funcional diamétrica hecha por cada muestra.
- C_ϕ : cantidad de operaciones para la funcional de circo hecha por cada muestra.
- N_T : cantidad de funcionales de trazo.
- N_P : cantidad de funcionales diamétricas.
- N_ϕ : cantidad de funcionales de circo.
- n_τ : cantidad de muestras del parámetro τ a lo largo del trazo diagonal más extenso.
- n_ρ : cantidad de muestras del parámetro ρ .
- n_ϕ : cantidad de muestras del parámetro ϕ .

El primer término domina considerablemente la complejidad del algoritmo, dado que n_τ , n_ρ y n_ϕ son los términos que dependen del tamaño de la entrada mientras que N_τ , N_ρ y N_ϕ son los mismos para cualquier tamaño de entrada.

El parámetro de frecuencia $\Delta\tau$ está presente en los tres términos; está representado por n_τ , por lo que es de especial importancia escoger un valor que minimice la cantidad de instrucciones. De igual forma ocurre con $\Delta\rho$ y $\Delta\phi$ pero en menor medida.

1.3. Planteamiento del problema

Como se mencionó previamente, la TT puede tener un tiempo de ejecución considerable, dependiendo de:

- **Los funcionales utilizados:** existen 31 funcionales asociados a texturas [9] que han tenido éxito en sus procesos de clasificación. Son necesarias muchas características para lograr una buena clasificación [9].
- **El tamaño de las imágenes:** las imágenes aéreas utilizadas en el PRIAS tienen tamaños de al menos 5000×5000 pixeles, lo cual es considerable si se toma en cuenta que los estudios anteriores trabajan sobre imágenes mucho más pequeñas [13] [14] [15] [8] [9].
- **Los parámetros de frecuencia (o Δ):** diferentes estudios muestran que variando estos parámetros es posible obtener buenos resultados [16] [9], mas no existe un estudio que muestre como afectan estas frecuencias de muestreo a los resultados.

Si bien los dos primeros ítemes no se pueden cambiar, el tercero sí. Al no conocerse con exactitud el efecto que pueda tener las variaciones en los parámetros δ sobre la precisión y el tiempo de ejecución de la TT, se puede estar cayendo en alguna de estas dos situaciones: más trabajo del necesario para lograr una buena clasificación, o bien, es posible que con un poco más de trabajo se pueda lograr una clasificación con una tasa de precisión muy alta.

1.4. Hipótesis

Basado en la ecuación de complejidad y la revisión del trabajo relacionado, se intuye que los parámetros de frecuencia pueden influir en el tiempo de extracción así como en la calidad de las TF que se generan para la clasificación, lo que lleva a plantear la siguiente hipótesis de investigación:

La Transformada de Trazo presenta mejores resultados estadísticamente, en precisión de clasificación y en tiempo de extracción de características, al variar los parámetros de frecuencia, el tipo de clasificador y el tipo de cobertura.

1.5. Trabajo relacionado

Desde la elaboración conceptual de la TT en el 2001 [6], esta ha producido varias publicaciones de casos de estudio en diferentes áreas.

La TT es una generalización de la transformada de Radon [17] [16] en la que se calcula como funcional de trazo la integral del mismo trazo. Este método ha sido utilizado en tomografía e imagenología médica [11].

Previo a la publicación de la TT, ya se manejaba la idea de extraer características invariantes de imágenes para la clasificación. En [13], se realizó una extracción de características para determinar si un paciente padecía de Alzheimer. Dicho estudio tuvo una precisión del 88 % al clasificar dichas imágenes.

Es posible identificar huellas de insectos utilizando la TT junto con otros métodos para la segmentación de imágenes. En [7] se muestra una mejora de entre 7 % y 63 % en comparación con el método convencional [18].

La TT se puede combinar con otros métodos para extraer características robustas. La combinación con algoritmos evolutivos mostró que da buenos resultados al clasificar TF [15].

Las características de invarianza o sensibilidad de las TF se pueden comparar en [19]. Aquí una serie de imágenes de referencia se compara con imágenes que han sido sujetas a transformaciones afines: rotación, desplazamiento y escalamiento. El estudio concluyó que la TT, usando la combinación de funcionales correctas tiene una precisión de entre 98 % y 100 %.

La TT se puede utilizar para clasificar o reconocer imágenes más complejas como rostros. En [8] se compara el método Eigenfaces[20] con una modificación de la TT llamada: “Masked Trace Transform” + “Weighted Trace Transform”. El método basado en la TT tuvo mejor desempeño en modificaciones como rotación, escalamiento y cambio en la expresión facial.

La variación de parámetros para producir las TF no es muy común y se utiliza una configuración estándar. Aun así, hay varios estudios donde se obtuvieron buenos resultados. En [16] se realizaron experimentos para crear firmas únicas para imágenes utilizando características extraídas con la TT. Aquí, los parámetros para $\Delta\rho$, $\Delta\tau$ y $\Delta\phi$ fueron respectivamente, 2 píxeles, 2 píxeles y 3° . El estudio reveló que las TF tienen un desempeño similar a las características extraídas con otros métodos, pero las primeras son más robustas y resistentes al cambio.

En [9] se comparan las características extraídas por métodos automáticos como la TT contra otros más rígidos como con el uso de matrices de co-ocurrencia, con base a la forma en como las personas interpretan las texturas. En general, las características extraídas con la TT se puede modelar de manera semejante a la forma en que los humanos perciben las texturas. En los experimentos realizados en este estudio, se utilizó, a diferencia de los experimentos previos, un $\Delta\rho$ de 2.

CAPÍTULO 2

OBJETIVOS Y APORTES

2.1. Objetivo general

Estudiar el comportamiento de la TT en términos de precisión y tiempo de extracción de características, utilizando diferentes modelos de parámetros, clasificadores y tipos de terreno.

2.2. Objetivos específicos

1. Comparar el tiempo de extracción de características y precisión de la clasificación de la TT utilizando diferentes combinaciones de los parámetros de frecuencia, cuatro distintos clasificadores y cinco tipos de cobertura de terreno.
2. Analizar estadísticamente los resultados observados para determinar la influencia de los parámetros en el tiempo de extracción de características y precisión de la clasificación

2.3. Alcances y limitaciones

Dentro del alcance de esta propuesta se definen los siguientes entregables:

1. Prototipo de un programa para la extracción de características usando la TT.
En este prototipo deben ser parametrizables, los parámetros de frecuencia y el clasificador.
2. Prototipo de los clasificadores.
3. Programas auxiliares para la ejecución y el control de los dos anteriores.
4. Análisis estadístico para contrastar los resultados de los experimentos.
5. Artículo científico que se entregará al comité editorial de alguna revista o conferencia, con miras a su publicación.

Por motivos de tiempo y extensión, en esta investigación no se tomarán en cuenta:

- Imágenes con formas irregulares, sólo se utilizarán imágenes rectangulares.
- Imágenes de tamaños variables. Se utilizarán imágenes en formato PNG, con un tamaño de 500×500 píxeles.
- Funcionales fuera de aquellas revisadas en la literatura. Se utilizarán los funcionales descritas en [9].
- Una ventana de análisis en la imagen, ni deslizante ni de tamaño variable. La ventana de análisis será del tamaño total de la imagen.

Tampoco es la intención de esta investigación presentar un sistema completo para la clasificación de imágenes o para la generación de mapas de cobertura.

Por último, no se tomará en cuenta cualquier otro resultado, documento, software o producción que no esté contemplado en esta propuesta.

CAPÍTULO 3

EXPERIMENTACIÓN

Con el fin de dar evidencia a favor de la hipótesis, se diseñaron dos experimentos siguiendo la metodología del diseño multifactorial[21].

Los resultados de ambos experimentos se estudiarán por medio de un **Análisis de Varianza**[21] (ANOVA por sus siglas en inglés). Aquí se toman n muestras de k poblaciones y estos datos se estudian para confirmar que vienen de un mismo origen (o sea, que todos se comportan estadísticamente igual) o no.

Es necesario establecer una hipótesis nula y una alternativa:

- Hipótesis nula H_0 : la media entre los grupos estudiados es la misma y no afecta el resultado.

$$\mu_0 = \mu_1 = \dots = \mu_n.$$

- Hipótesis alternativa H_1 : la media de al menos uno de los grupos estudiados es diferente y afecta el resultado.

$$\exists i, j / \mu_i \neq \mu_j.$$

3.1. Experimento A: Tiempo de ejecución de la extracción de características

El primer experimento se realizó para medir la influencia de los parámetros de frecuencia y del tipo de cobertura de terreno sobre la variable de respuesta, el **tiempo requerido para generar las características** de una imagen. Siguiendo el modelo multifactorial[21], los niveles y factores para el experimento A fueron:

- $\Delta\tau$ (píxeles): 1, 2 y 3.
- $\Delta\rho$ (píxeles): 1, 2 y 3.
- $\Delta\phi$ (grados): $0,5^\circ$, 1° y 2° .
- Cobertura de terreno (tipos): bosque, agua, rural, urbano y sembradío.

Con 32 imágenes por tipo de cobertura y los niveles de las variables anteriores, se tienen 4320 ($3 \times 3 \times 3 \times 32 \times 5$) ejecuciones por cada una de las tres réplicas. Esto suma un gran total de 12960 ejecuciones de la TT.

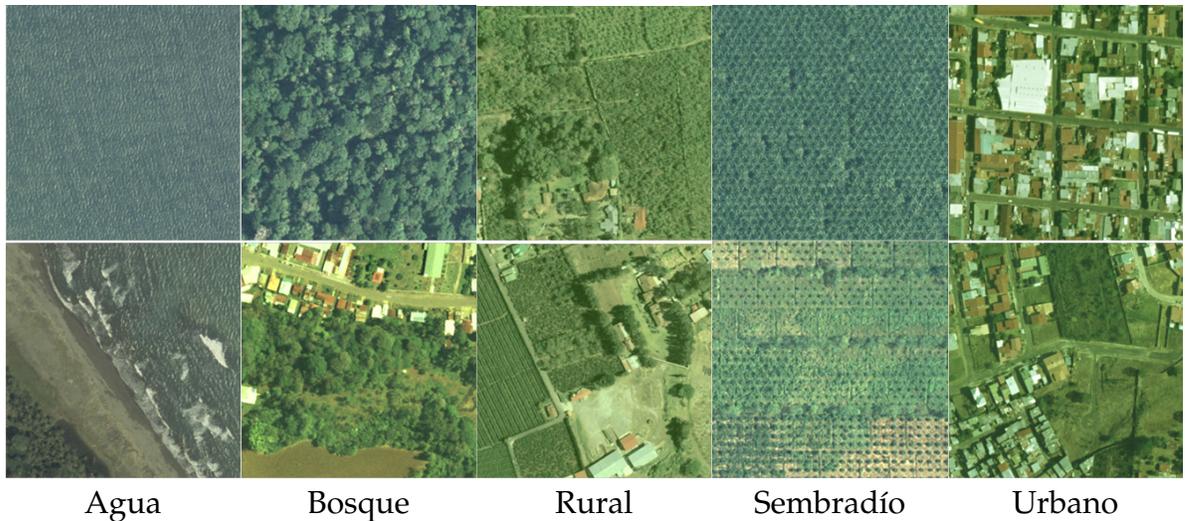
Por cada ejecución de la TT se producen dos resultados: un arreglo de TF por cada imagen y el tiempo de ejecución de la extracción de características. El primero se utilizó para el experimento B y el segundo se evaluó con ANOVA.

A continuación se explican los detalles del experimento A.

3.1.1. Conjunto de imágenes

Las imágenes utilizadas para la extracción de características provienen del proyecto Carta 2005[4] y fueron proporcionadas por el PRIAS. A su vez, éstas fueron recortadas y clasificadas previamente por un experto geógrafo.

Se contó con 32 imágenes por tipo de cobertura, lo que suma 160 imágenes en total. Cada imagen tiene una dimensión de 500×500 píxeles almacenadas en formato PNG y se eligió este tamaño previendo el tiempo de extracción de alrededor 130 segundos por imagen, según [10].



Cuadro 3.1: Ejemplo de tipos de cobertura. Cada columna presenta dos imágenes de un tipo de cobertura distinto, respectivamente: agua, bosque, rural, sembradío y urbano.

El cuadro 3.1 muestra diferentes ejemplos de cada tipo de textura. En la primera fila se muestran texturas más homogéneas y en la segunda fila texturas más heterogéneas de cada tipo de cobertura.

3.1.2. Funcionales

Las funcionales utilizadas en cada uno de los tres pasos de reducción son críticas para extraer características útiles para la etapa de clasificación y puede en realidad ser un tema de investigación por sí solo. Es por esto que las funcionales que se utilizaron en el experimento A son aquellas que mostraron buenos resultados al clasificar texturas en [9] y [11].

#	Funcional
1.	$\sum_{i=1}^N x_i$
2.	$\sum_{i=1}^N ix_i$

3.1. EXPERIMENTO A: TIEMPO DE EJECUCIÓN DE LA EXTRACCIÓN DE CARACTERÍSTICAS

#	Funcional
3.	$\frac{1}{N} \sqrt{\sum_{i=1}^N (x_i - \hat{x})^2}$
4.	$\sqrt{\sum_{i=1}^N x_i^2}$
5.	$\max_{i=1}^N x_i$
6.	$\sum_{i=1}^{N-1} x_{i+1} - x_i $
7.	$\sum_{i=1}^{N-1} x_{i+1} - x_i ^2$
8.	$\sum_{i=4}^{N-3} x_{i-3} + x_{i-2} + x_{i-1} - x_{i+1} - x_{i+2} - x_{i+3} $
9.	$\sum_{i=3}^{N-2} x_{i-2} + x_{i-1} - x_{i+1} - x_{i+2} $
10.	$\sum_{i=5}^{N-4} x_{i-4} + x_{i-3} + \dots + x_{i-1} - x_{i+1} - \dots - x_{i+3} - x_{i+4} $
11.	$\sum_{i=6}^{N-5} x_{i-5} + x_{i-4} + \dots + x_{i-1} - x_{i+1} - \dots - x_{i+4} - x_{i+5} $
12.	$\sum_{i=7}^{N-6} x_{i-6} + x_{i-5} + \dots + x_{i-1} - x_{i+1} - \dots - x_{i+5} - x_{i+6} $
13.	$\sum_{i=8}^{N-7} x_{i-7} + x_{i-6} + \dots + x_{i-1} - x_{i+1} - \dots - x_{i+6} - x_{i+7} $
14.	$\sum_{i=5}^{N-4} \sum_{k=1}^4 x_{i-k} - x_{i+k} $
15.	$\sum_{i=6}^{N-5} \sum_{k=1}^5 x_{i-k} - x_{i+k} $
16.	$\sum_{i=7}^{N-6} \sum_{k=1}^6 x_{i-k} - x_{i+k} $
17.	$\sum_{i=8}^{N-7} \sum_{k=1}^7 x_{i-k} - x_{i+k} $
18.	$\sum_{i=11}^{N-10} \sum_{k=1}^{10} x_{i-k} - x_{i+k} $
19.	$\sum_{i=16}^{N-15} \sum_{k=1}^{15} x_{i-k} - x_{i+k} $
20.	$\sum_{i=21}^{N-20} \sum_{k=1}^{20} x_{i-k} - x_{i+k} $
21.	$\sum_{i=26}^{N-25} \sum_{k=1}^{25} x_{i-k} - x_{i+k} $
22.	$\sum_{i=11}^{N-10} \left((1 + \sum_{k=1}^{10} x_{i-k} - x_{i+k}) / (1 + \sum_{k=-10}^9 x_{i-k} - x_{i+k}) \right)$

#	Funcional
23.	$\sum_{i=11}^{N-10} \sqrt{\left \left(\sum_{k=1}^{10} x_{i-k} - x_{i+k} \right)^2 / \left(1 + \sum_{k=-10}^9 x_{i-k} - x_{i+k} \right) \right }$
24.	$\sum_{i=1}^{N-2} x_i - 2x_{i+1} + x_{i+2} $
25.	$\sum_{i=1}^{N-3} x_i - 3x_{i+1} + 3x_{i+2} - x_{i+3} $
26.	$\sum_{i=1}^{N-4} x_i - 4x_{i+1} + 6x_{i+2} - 4x_{i+3} + x_{i+4} $
27.	$\sum_{i=1}^{N-5} x_i - 5x_{i+1} + 10x_{i+2} - 10x_{i+3} + 5x_{i+4} - x_{i+5} $
28.	$\sum_{i=1}^{N-2} x_i - 2x_{i+1} + x_{i+2}x_{i+1} $
29.	$\sum_{i=1}^{N-3} x_i - 3x_{i+1} + 3x_{i+2} - x_{i+3}x_{i+1} $
30.	$\sum_{i=1}^{N-4} x_i - 4x_{i+1} + 6x_{i+2} - 4x_{i+3} + x_{i+4}x_{i+2} $
31.	$\sum_{i=1}^{N-5} x_i - 5x_{i+1} + 10x_{i+2} - 10x_{i+3} + 5x_{i+4} - x_{i+4}x_{i+2} $

Cuadro 3.2: Funcionales de trazo utilizadas en la extracción de características.

Las 31 funcionales de trazo se presentan en el cuadro 3.2. Muchas de las funcionales utilizadas aquí exploran un área sobre τ . Por ejemplo, las funcionales 10, 11, 12 y 13 suman, dado el punto de tangencia, el valor de los píxeles en ambas direcciones de la recta; cada vez en un vecindario más grande. Otras funcionales calculan valores interesantes del trazo, el punto máximo (5) o la suma de todos los puntos (1).

#	Funcional
1.	$\max_{i=1}^N x_i$
2.	$\min_{i=1}^N x_i$
3.	$\sqrt{\sum_{i=1}^N x_i^2}$

3.1. EXPERIMENTO A: TIEMPO DE EJECUCIÓN DE LA EXTRACCIÓN DE CARACTERÍSTICAS

#	Funcional
4.	$\frac{\sum_{i=1}^N ix_i}{\sum_{i=1}^N x_i}$
5.	$\sum_{i=1}^N ix_i$
6.	$\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x})^2$
7.	c tal que: $\sum_{i=1}^c x_i = \sum_{i=c}^N x_i$
8.	$\sum_{i=1}^{N-1} x_{i+1} - x_i $
9.	c tal que: $\sum_{i=1}^c x_{i+1} - x_i = \sum_{i=c}^{N-1} x_{i+1} - x_i $
10.	$\sum_{i=1}^{N-4} x_i - 4x_{i+1} + 6x_{i+2} - 4x_{i+3} + x_{i+4} $

Cuadro 3.3: Funcionales diamétricas utilizadas en la extracción de características.

Las 10 funcionales de diámetro utilizadas se presentan en el cuadro 3.3. De estas, algunas son de la misma familia que varias funcionales de trazo, pero aparecen aquí algunas totalmente distintas, como la 7 y la 9. En la 7, se calcula un número c sobre el trazo en donde la suma de los píxeles antes de c sea igual a la suma de los píxeles luego de c , esto con el fin de encontrar un punto de equilibrio sobre el trazo. En caso de no existir, se utiliza el valor de 0.

#	Funcional
1.	$\sum_{i=1}^{N-1} x_{i+1} - x_i ^2$
2.	$\sum_{i=1}^{N-1} x_{i+1} - x_i $
3.	$\sqrt{\sum_{i=1}^N x_i^2}$
4.	$\sum_{i=1}^N x_i$
5.	$\max_{i=1}^N x_i$

#	Funcional
6.	$\max_{i=1}^N x_i - \min_{i=1}^N x_i$
7.	i tal que $x_i = \min_{i=1}^N x_i$
8.	i tal que $x_i = \max_{i=1}^N x_i$
9.	i tal que $x_i = \min_{i=1}^N x_i$ sin la primera armónica
10.	i tal que $x_i = \max_{i=1}^N x_i$ sin la primera armónica
11.	Amplitud de la primera armónica
12.	Fase de la primera armónica
13.	Amplitud de la segunda armónica
14.	Fase de la segunda armónica
15.	Amplitud de la tercera armónica
16.	Fase de la tercera armónica
17.	Amplitud de la cuarta armónica
18.	Fase de la cuarta armónica

Cuadro 3.4: Funcionales de circo utilizadas en la extracción de características.

Las funcionales de circo se presentan en el cuadro 3.4 e introducen propiedades de las armónicas de las transformadas diamétricas interpretadas como ondas (de la 11 a la 18). Éstas exponen características físicas que puede facilitar la identificación de las imágenes originales.

3.1.3. Implementación del algoritmo de la TT

La implementación de la TT se basó en el código [22]. Esta consiste de 3 etapas:

1. Carga de imágenes

Las imágenes se cargan una a la vez, y en el caso de tener que procesar múltiples imágenes se puede utilizar paralelismo a nivel de tareas (programas) y

3.1. EXPERIMENTO A: TIEMPO DE EJECUCIÓN DE LA EXTRACCIÓN DE CARACTERÍSTICAS

cargar una imagen por cada programa. La carga de imágenes se realizó utilizando la biblioteca para manipulación de imágenes *ImageMagick*¹. Es necesario convertir los píxeles de la imagen a una representación en escala de grises, lo cual se lleva a cabo con la función `MagickQuantizeImage` [23], usando como tercer parámetro `GRAYColorspace`. Esta función analiza los colores de la imagen y los mapea al espacio lineal de grises usando el algoritmo `quantize`[24]. Esto permite utilizar una sola capa de color para el cálculo de las funcionales. Una vez que se haya levantado la imagen en memoria, se pasa a la segunda etapa de preprocesamiento.

2. Preprocesamiento de las imágenes

La etapa de preprocesamiento calcula datos comunes que serán utilizados durante la extracción de características con el fin de no tener que recomputarlos. Aquí es importante tomar en cuenta las características de memoria del equipo donde corre la implementación, ya que por cada imagen se debe almacenar una cantidad considerable de información que tiene una relación directa con el tamaño de la imagen y los parámetros Δ . Estos son:

- Alto y Ancho.
- $\Delta\tau$, $\Delta\rho$ y $\Delta\phi$.
- Tamaño de la cuadrícula, dependiente de la arquitectura para maximizar el detalle.
- Píxel de inicio y fin de cada trazo.
- El desplazamiento en los ejes x y y para cada trazo, dependiente de los parámetros Δ .

3. Extracción de características

Por último, se aplican los tres juegos de funcionales a los datos. Primero, se seleccionan los puntos involucrados en las funcionales de trazo, utilizando los

¹<http://www.imagemagick.org/>

```

1   int phi;
2   for (phi = 0; phi < 360; phi++) { ... }

```

Figura 3.1: Ciclo original, utilizando $\Delta\phi = 1$.

datos precalculados en la etapa anterior, para luego aplicar de forma iterativa todas las funcionales. Cada funcional de cada paso (trazo, diamétrica o de circo) podría ser ejecutada de forma independiente ya que la dependencia existe entre cada juego de funcionales y no entre las funcionales de cada juego. El pseudocódigo en el algoritmo 1 muestra este proceso.

Algoritmo 1 Procesamiento de la imagen y cálculo de las características triples (I_ϕ)

```

I = Imagen a procesar
for a en ángulos a procesar do
  for d en trazos a procesar do
    for t en puntos a procesar do
      // Calcular las funcionales de trazo
       $I_\tau = \text{calcularFuncionalesTrazo}(I, a, d, t)$ 
    // Luego de procesadas todas las funcionales de trazo
    for d en trazos a procesar do
      // Calcular las funcionales diamétricas para cada funcional de trazo
       $I_\rho = \text{calcularFuncionalesDiametrica}(I_\tau)$ 
    // Luego de procesadas todas las funcionales de trazo y diamétricas
    for a en ángulos a procesar do
      // Calcular las funcionales de circo para cada funcional trazo+diamétrica
       $I_\phi = \text{calcularFuncionalesCirco}(I_\rho)$ 

```

La implementación[22] debió ser modificada, ya que no aceptaba parámetros de frecuencia distintos a los originales. Esto requirió cambiar todos los ciclos cuyo índice o límite tuviera relación con $\Delta\phi = 1$.

La figura 3.1 muestra uno de los ciclos originales. En este se puede notar como el bucle toma en cuenta, grado por grado, todas las rotaciones en 360 grados. Luego de modificarse, el límite del ciclo es la variable `_360EntredPhi`, que se define como la razón de todos los grados en el círculo (360°) y el parámetro

3.1. EXPERIMENTO A: TIEMPO DE EJECUCIÓN DE LA EXTRACCIÓN DE CARACTERÍSTICAS

```

1     double dPhi = 3; // dPhi puede tomar cualquier valor.
2     int phi;
3     int _360EntredPhi = 360 / dPhi;
4     for (phi = 0; phi < _360EntredPhi; phi++) { ... }

```

Figura 3.2: Ciclo modificado, utilizando $\Delta\phi = 3$.

de frecuencia $\Delta\phi$, dPhi. En este caso particular, dPhi toma un valor de 3 pero podría ser cualquier otro.

Dado que el programa de [22] se enfocaba en la paralelización, los parámetros de ejecución son dos: la imagen a procesar y la cantidad de hilos o procesos a utilizar para la paralelización. La salida del programa, los TF de cada imagen, eran desplegadas en pantalla. Las modificaciones hechas aquí también cambiaron estos comportamientos. Los parámetros necesarios ahora son seis, de los cuales, el segundo parámetro del programa se fijó en 1. Estos parámetros son:

- a) La imagen a procesar.
- b) El número de hilos a utilizar.
- c) El parámetro $\Delta\tau$
- d) El parámetro $\Delta\rho$
- e) El parámetro $\Delta\phi$
- f) La ruta de un archivo de salida donde se almacena las TT de la imagen.

El programa original presenta las TF en líneas separadas por tabulaciones. Esto se puede interpretar como una tabla, donde cada TF en las filas está separado por una tabulación como se muestra en el cuadro 3.5. Este comportamiento también se modificó para que las TF fueran más fáciles de procesar en los pasos siguientes.

	1	2	...	17	18
1	$T_0D_0C_0$	$T_0D_0C_1$...	$T_0D_0C_{17}$	$T_0D_0C_{18}$

	1	2	...	17	18
2	$T_0D_1C_0$	$T_0D_1C_1$...	$T_0D_1C_{17}$	$T_0D_1C_{18}$
3	$T_0D_2C_0$	$T_0D_2C_1$...	$T_0D_2C_{17}$	$T_0D_2C_{18}$
...
10	$T_0D_9C_0$	$T_0D_9C_1$...	$T_0D_9C_{17}$	$T_0D_9C_{18}$
11	$T_1D_0C_0$	$T_1D_0C_1$...	$T_1D_0C_{17}$	$T_1D_0C_{18}$
12	$T_1D_1C_0$	$T_1D_1C_1$...	$T_1D_1C_{17}$	$T_1D_1C_{18}$
13	$T_1D_2C_0$	$T_1D_2C_1$...	$T_1D_2C_{17}$	$T_1D_2C_{18}$
...
20	$T_1D_9C_0$	$T_1D_9C_1$...	$T_1D_9C_{17}$	$T_1D_9C_{18}$
21	$T_2D_0C_0$	$T_2D_0C_1$...	$T_2D_0C_{17}$	$T_2D_0C_{18}$
...
310	$T_{30}D_9C_0$	$T_{30}D_9C_1$...	$T_{30}D_9C_{17}$	$T_{30}D_9C_{18}$

Cuadro 3.5: Salida de la implementación original de la TT. Cada celda en la tabla representa una TT, donde T_n indica la funcional de trazo, D_m la funcional diamétrica y C_p la funcional de circo usadas para generarla.

La versión modificada utiliza un formato separado por coma, con una sola fila donde van todas las TF, como se muestra en el cuadro 3.6.

	1	...	18	19	20	...	5580
1	$T_0D_0C_0$...	$T_0D_0C_{17}$	$T_0D_1C_0$	$T_0D_1C_1$...	$T_{30}D_9C_{18}$

Cuadro 3.6: Salida de la implementación modificada de la TT. Cada celda en la línea representa una TT, donde T_n indica la funcional de trazo, D_m la funcional diamétrica y C_p la funcional de circo usadas para generarla.

La implementación modificada corrió en el *cluster* de procesamiento del CN-CA. Este *cluster* cuenta con cinco nodos con 32 GBytes de memoria RAM y doble procesador Xeon E5530, por lo que se decidió extraer las características

de una imagen por nodo por unidad de tiempo. Esto minimiza los problemas de cambio de contexto o acceso a *cache*.

3.2. Experimento B: Precisión de clasificación

El propósito del segundo experimento es medir la influencia de los parámetros de frecuencia y del tipo de clasificador sobre la variable de respuesta, la precisión de clasificación. En este caso, los factores y niveles fueron:

- $\Delta\tau$ (píxeles): 1, 2 y 3.
- $\Delta\rho$ (píxeles): 1, 2 y 3.
- $\Delta\phi$ (grados): 0,5°, 1° y 2°.
- Clasificador (tipos): KMeans, BayesNet, SMO y LogitBoost.

A diferencia del Experimento A, aquí basta con 1 sola replica, dado que los algoritmos siempre darán el mismo resultado y no se están tomando tiempos. Entonces se ejecutaron 108 corridas del experimento ($3 \times 3 \times 3 \times 4 \times 1$).

A todos los arreglos de TF de todas las imágenes por cada configuración del experimento A se les añadió al final un valor más: la categoría de la imagen de la cual se extrajeron las características. Luego de esto, los arreglos se concatenaron en un solo archivo, lo que resulta en 27 archivos de entrada para el experimento B.

Los archivos de entrada siguen el formato ARFF², utilizado por la herramienta de minería de datos y aprendizaje automático Weka[25], la cual se usó para evaluar los clasificadores. Se escogió Weka por varias razones:

- Es una herramienta aceptada por la comunidad académica y cuya funcionalidad está demostrada[26][27][28].
- La estandarización de entrada y salida de los clasificadores.
- La forma de particionar el conjunto de datos en “entrenamiento” y “evalua-

²<http://www.cs.waikato.ac.nz/ml/weka/arff.html>

ción”.

- La capacidad de guardar el resultado de la clasificación, para poder explorar más de cerca los datos de ser necesario.
- El detalle con el cual se muestra la salida de cada clasificación. En la figura 3.3 se muestra un ejemplo de salida para una clasificador SMO.

```

=== Summary ===

Correctly Classified Instances      139           86.875 %
Incorrectly Classified Instances    21           13.125 %
Kappa statistic                    0.8359
Mean absolute error                 0.2467
Root mean squared error            0.3264
Relative absolute error             77.077 %
Root relative squared error        81.5594 %
Total Number of Instances          160

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0.75    0.047    0.8        0.75    0.774      0.885    bosque
          0.875   0.031    0.875     0.875   0.875      0.978    urbano
          0.969   0.008    0.969     0.969   0.969      0.971    sembradio
          0.969    0        1          0.969   0.984      1        agua
          0.781   0.078    0.714     0.781   0.746      0.92     rural
Weighted Avg.  0.869   0.033    0.872     0.869   0.87       0.951

=== Confusion Matrix ===

  a  b  c  d  e  <-- classified as
24  1  0  0  7  |  a = bosque
 2 28  0  0  2  |  b = urbano
 0  1 31  0  0  |  c = sembradio
 0  0  0 31  1  |  d = agua
 4  2  1  0 25  |  e = rural
    
```

Figura 3.3: Fragmento de la salida de un clasificador SMO en Weka. Se pueden identificar datos importantes como el número y porcentaje de instancias clasificadas correcta e incorrectamente, la matriz de confusión y un bloque detallando la precisión por categoría.

3.2.1. Cálculo de Precisión

La métrica de precisión que se utilizó para medir el desempeño de este experimento fue provista por el mismo Weka. El estadístico *correctly classified instances* muestra tanto la cantidad de imágenes clasificadas correctamente así como un porcentaje. Este valor se calcula utilizando la matriz de confusión, como se muestra en la siguiente función:

$$\text{Precisión} = \frac{M_{1,1} + M_{2,2} + \dots + M_{n,n}}{\text{Total de instancias}} \quad (3.1)$$

Donde:

- M es la matriz de confusión.
- n es el total de categorías, en este caso tipos de cobertura.

3.2.2. Clasificadores

Se seleccionaron cuatro clasificadores para evaluar la precisión. Cada uno tiene una naturaleza distinta: agrupamiento, basado en funciones, regresión y teoría Bayesiana. A continuación se describen cada uno de los clasificadores.

3.2.2.1. K-Means

El primer clasificador es un algoritmo básico de agrupamiento (o *clustering* en inglés), llamado **K-Means**[29].

A grandes rasgos, este algoritmo interpreta cada arreglo de TF como un vector euclidiano en un espacio n -dimensional. Sobre este espacio se lanzan k puntos, los *centroides*, de manera aleatoria. Posteriormente, se hacen i iteraciones en las cuales se miden los vectores más cercanos a cada centroide el cual se reposiciona hacia el centro de los vectores cercanos. La idea es que luego de varias pasadas, los centroides irán gravitando hacia el centro de las nubes de puntos que representan las categorías

(en este caso imágenes).

En Weka, la implementación de K-Means se denomina “simpleKMeans”. Los parámetros de ejecución fueron:

- Uso de todo el conjunto de datos para entrenamiento.
- 500 iteraciones.
- Uso de distancia euclidiana para el cálculo de cercanía.

3.2.2.2. Máquina de Soporte Vectorial

El segundo clasificador es un algoritmo de aprendizaje automático llamado **Máquina de Soporte Vectorial** (o *Support Vector Machine* en inglés), en este caso particular usando el algoritmo de Optimización Secuencial Mínima[30] (SMO por sus siglas en inglés). Las Máquinas de Soporte Vectorial se utilizan normalmente en datos con alta dimensionalidad. Aquí, los datos se intentan dividir (clasificar) por medio de un hiperplano. Este plano no siempre existe por lo que los datos son transformados por una función no lineal llamada *kernel* con el fin de facilitar su determinación. En Weka, este algoritmo se denomina “SMO” y sus parámetros de ejecución fueron:

- Entrenamiento usando Validación Cruzada de 10 iteraciones. Esta técnica se muestra en la figura 3.4.
- Función *Kernel* polinomial.
- Error aceptable de $1,0 \times 10^{-12}$.
- Normalización de los datos.

3.2.2.3. Regresión

El siguiente clasificador sigue un modelo aditivo generalizado, (o *boosting* en inglés). Esta técnica transforma un algoritmo débil en uno robusto al aplicarlo en sucesiones e ir ajustando una serie de pesos a lo largo de cada iteración. El algoritmo débil utilizado en este caso es la clásica técnica estadística de regresión, particularmente la

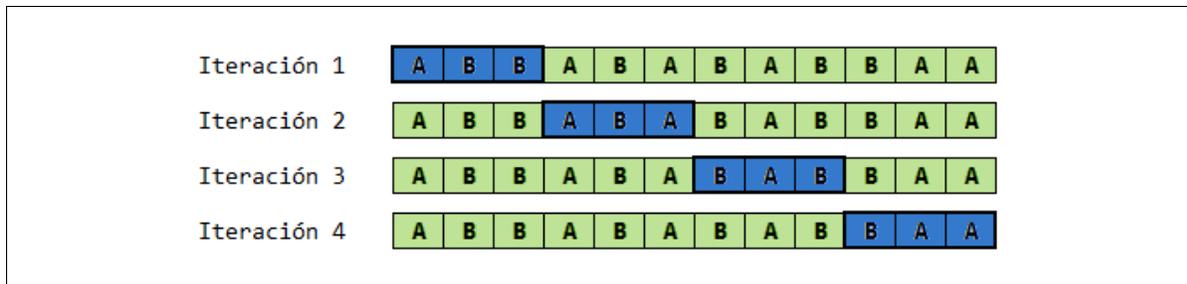


Figura 3.4: Técnica de Validación Cruzada de 4 iteraciones. El conjunto de datos está compuesto por ítems de los grupos A y B. En cada iteración, el grupo de elementos en azul se usa para validación y el grupo en verde para entrenamiento. Estos grupos van rotando tantas veces como iteraciones se dispongan.

regresión logística. La implementación de esta técnica se denomina **LogitBoost**[31]. En Weka se conoce con el mismo nombre y se utilizó con los parámetros por defecto.

3.2.2.4. Redes bayesianas

El último clasificador es el algoritmo de **red bayesiana**[32]. Las redes bayesianas son un modelo probabilístico que representa un conjunto de variables aleatorias y las condiciones de sus dependencias a través de un grafo dirigido cuyas aristas son probabilidades de que, dada una variable, se dé otra. En el caso de la TT, se debe entender cada TF como una variable aleatoria y la red bayesiana intenta mapear las dependencias entre cada TF con el fin de maximizar la probabilidad de que todo el vector corresponda con la imagen original. En Weka se denomina “BayesNet” y se usó con los parámetros por defecto.

CAPÍTULO 4

RESULTADOS Y ANÁLISIS

Los datos experimentales se analizaron por medio de ANOVA, donde los grupos estudiados para el experimento A son la combinación de cada uno de los factores como muestra el cuadro 4.1.

$\Delta\tau$ (píxeles)	$\Delta\rho$ (píxeles)	$\Delta\phi$ (grados)	Cobertura
1	1	0,5	Agua
1	1	0,5	Bosque
1	1	0,5	Urbano
...
1	1	1	Agua
1	1	1	Bosque
...
1	1	2	Agua
1	1	2	Bosque
...
3	3	2	Urbano

Cuadro 4.1: Grupos estudiados en el experimento A.

Para el experimento B, los grupos estudiados fueron se presentan en el cuadro 4.2.

$\Delta\tau$ (píxeles)	$\Delta\rho$ (píxeles)	$\Delta\phi$ (grados)	Clasificador
1	1	0,5	K-Means
1	1	0,5	SMO
1	1	0,5	LogitBoost
...
1	1	1	K-Means
1	1	1	SMO
...
1	1	2	K-Means
1	1	2	SMO
...
3	3	2	BayesNet

Cuadro 4.2: Grupos estudiados en el experimento B.

Para poder utilizar ANOVA es necesario validar una serie de supuestos[21], que en caso de no hacerlo, no se puede asegurar la validez del procedimiento. Estos supuestos son:

1. **Independencia y aleatoriedad** de las observaciones: esto se valida con el diseño experimental, donde se muestra que cada corrida del experimento A y B es independiente de cualquier otra. Este supuesto se aseguró al ejecutar una sola extracción por nodo y al ejecutar las extracciones de manera aleatoria.
2. **Normalidad** de los residuos: Esto se puede verificar con la prueba de Lilliefors (Kolmogorov-Smirnov)[33] o de manera visual con un gráfico de cuantiles contra cuantiles (Q-Q en inglés). Para la prueba, el estadístico D debe ser menor que el valor-p para confirmar normalidad. En el gráfico Q-Q, se grafican los residuos estandarizados de la variable medida por cuantiles contra los cuantiles teóricos de la variable de respuesta y una línea de tendencia, mientras más se alejen los puntos de esta línea, menos normales son los datos.

3. **Homogeneidad** de varianzas (homocedasticidad): Esto se puede verificar con la prueba de Levene[34] o de manera visual con un gráfico de residuos contra valores ajustados (RvVA) de ANOVA. Para la prueba, el estadístico W debe ser menor que el valor-p para confirmar homogeneidad. En el gráfico, los puntos deben estar dispersos y no seguir algún patrón fácilmente reconocible, en caso contrario, los datos no son homocedásticos.

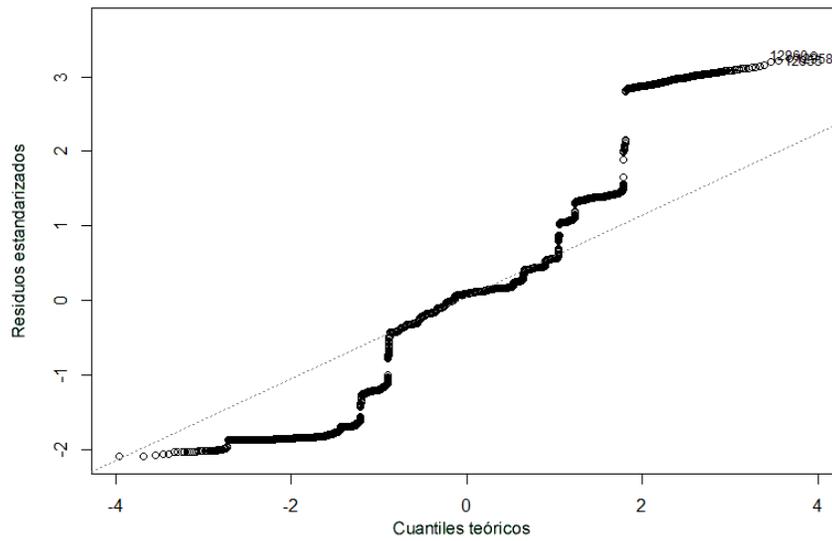
El paquete estadístico R¹ se utilizó para todo el análisis siguiente.

4.1. Experimento A

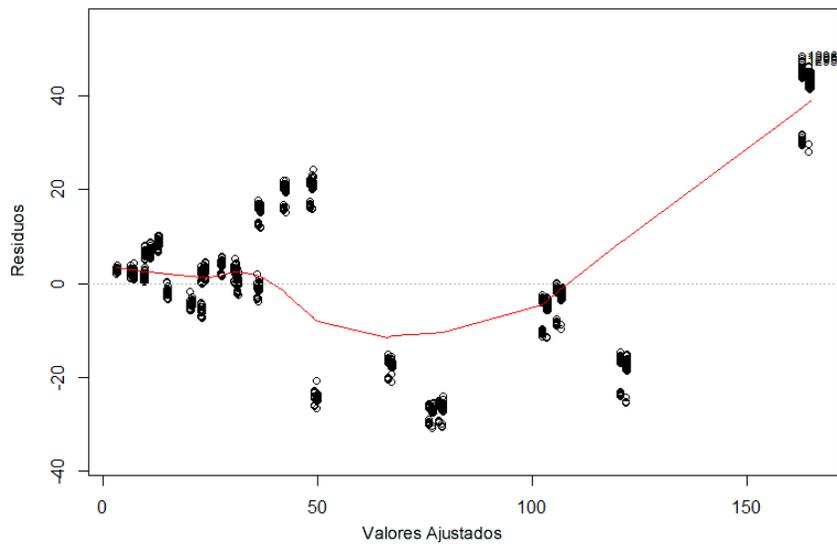
4.1.1. Verificación de supuestos

Para los datos de tiempo recolectados en el experimento A, se utilizó la verificación visual inicialmente. En la figura 4.1 se muestra el gráfico Q-Q y el gráfico RvVA. En ambos gráficos se puede notar como los supuestos no se cumplen; los puntos se alejan de forma variable en un patrón de S de la línea de tendencia para la normalidad y se nota un patrón de cono y varios cúmulos de puntos en el gráfico de la homogeneidad. Para confirmar estas hipótesis se ejecutaron las pruebas estadísticas para normalidad y homogeneidad, que confirmaron lo observado en los gráficos. El resultado de la prueba de Lilliefors se muestra en la figura 4.2 y el resultado de la prueba de Levene en la figura 4.3. En ambos casos el estadístico calculado descarta ambos supuestos.

¹<http://www.r-project.org/>



(a) Cuantiles contra Cuantiles



(b) Residuos contra Valores Ajustados

Figura 4.1: Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo sin transformaciones.

```
> lillie.test(data-t)
      Lilliefors (Kolmogorov-Smirnov) normality test
data:  data-t
D = 0.2085, p-value < 2.2e-16
```

Figura 4.2: Prueba de Lilliefors para normalidad en R. El estadístico es mayor que el valor-p por lo que los datos no son normales.

```
> levene.test(data-t, data-grupo)
      modified robust Brown-Forsythe Levene-type test
      based on the absolute deviations from the median
data:  data-t
Test Statistic = 21.8013, p-value < 2.2e-16
```

Figura 4.3: Prueba de Levene para homogeneidad de varianzas en R. El estadístico es mayor que el valor-p por lo que los datos no tiene varianzas homogéneas.

Según [21] en el caso de que los supuestos de ANOVA no se cumplan se pueden tomar varios caminos:

1. Aplicar transformaciones a los datos para que cumplan los supuestos y hacer un análisis basado en los nuevos datos.
2. Utilizar un Modelo Lineal Generalizado.
3. Usar el método Box-Cox.

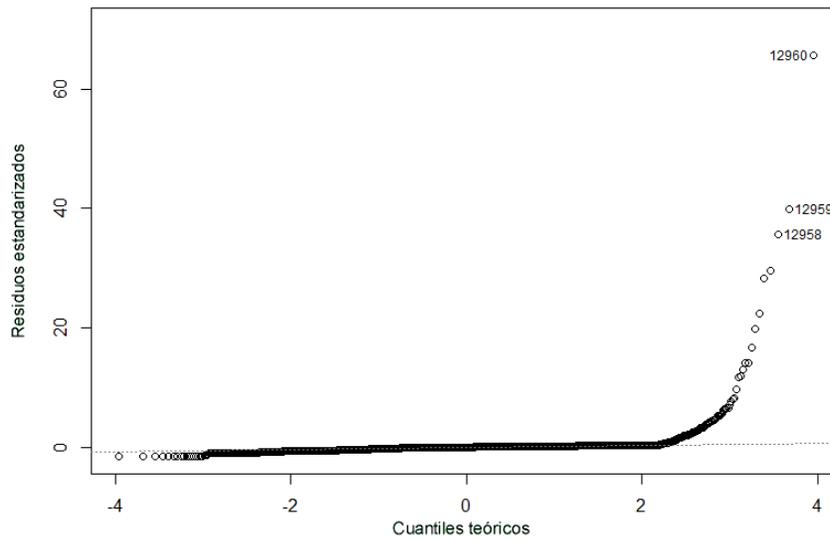
Todas las estrategias se siguieron y fallaron. A continuación, se presentara el detalle de cada una y la solución final, que consistía en hacer una transformación por rangos[35].

4.1.1.1. Transformaciones

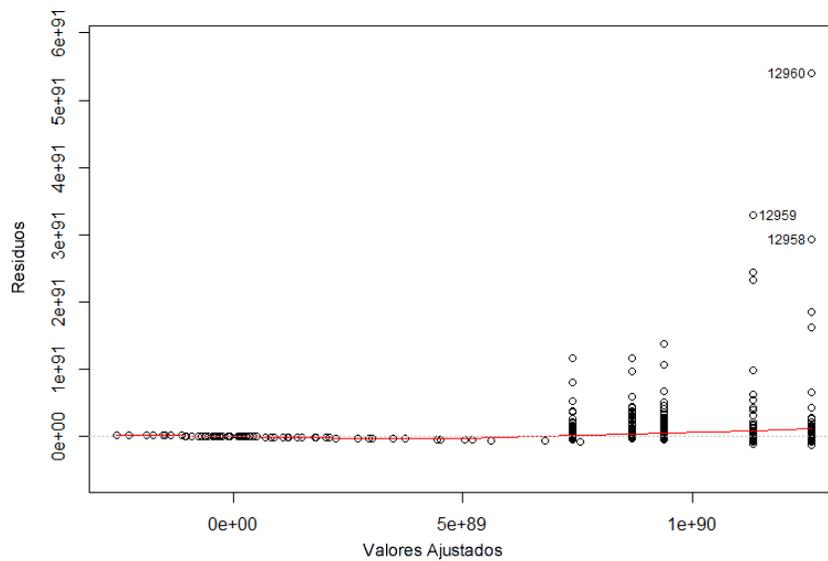
Según la literatura, se pueden aplicar transformaciones según se vea conveniente. Para este caso, se aplicaron las transformaciones de potencia e , raíz cuadrada y logaritmo base 10. Estas son las más comunes y a su vez las que tienen más sentido a la hora de la interpretación de los nuevos datos transformados: una transformación de potencia o logaritmo amplía o reduce la distancia de los puntos más altos, en este caso, los tiempos de mayor duración. Es mucho más complejo interpretar una transformación como la siguiente:

$$t - \text{transformado} = (t + 24)^2 - \text{sen}(2 * \pi * t) \quad (4.1)$$

Desgraciadamente ninguna de éstas transformaciones normalizó o cumplió la homogeneidad de varianzas, la exploración visual se hizo como al inicio y reveló los **mismos problemas**. Las figuras 4.4, 4.5 y 4.6 muestran los gráficos Q-Q y RvVA de estas transformaciones. En todos los gráficos Q-Q se nota como los puntos se alejan de la línea de tendencia. Los gráficos RvAF también tiene patrones bien marcados, de conos o cúmulos de puntos.



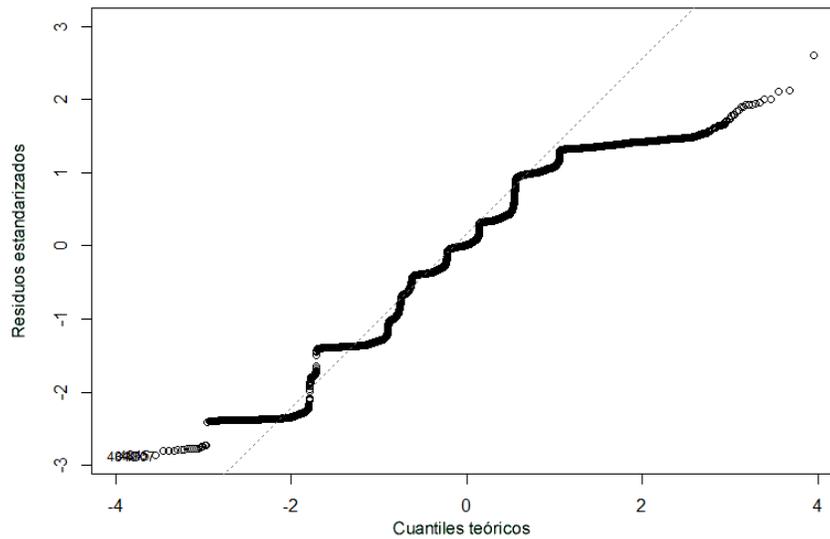
(a) Cuantiles contra Cuantiles de la Transformación con potencia e .



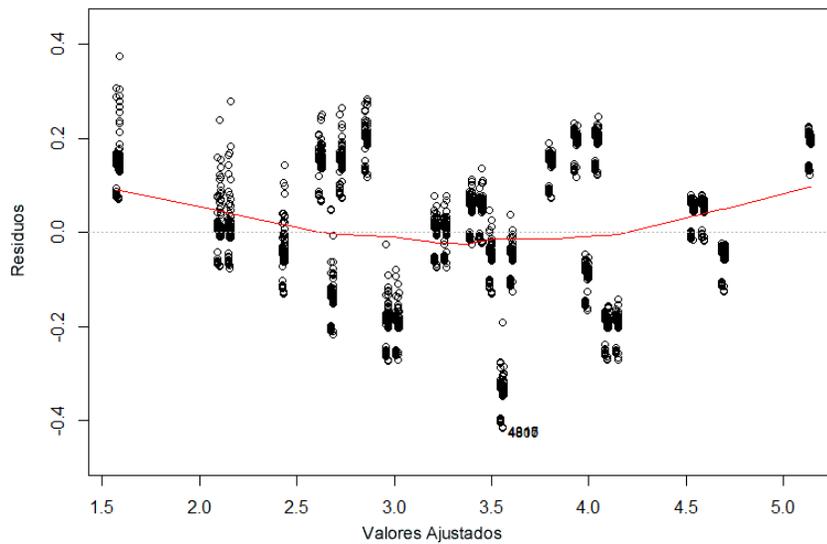
(b) Residuos contra Valores Ajustados de la Transformación con potencia e .

Figura 4.4: Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo transformados con una potencia e .

4.1. EXPERIMENTO A

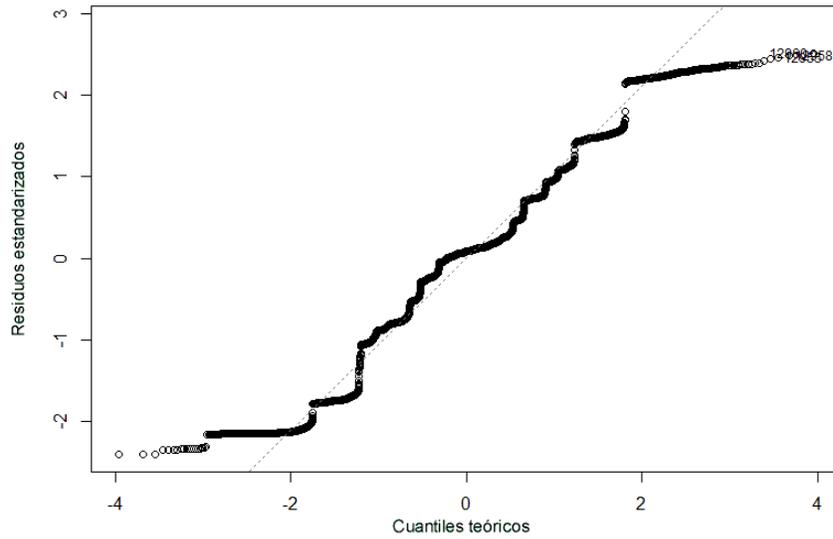


(a) Cuantiles contra Cuantiles de la Transformación con logaritmo base 10.

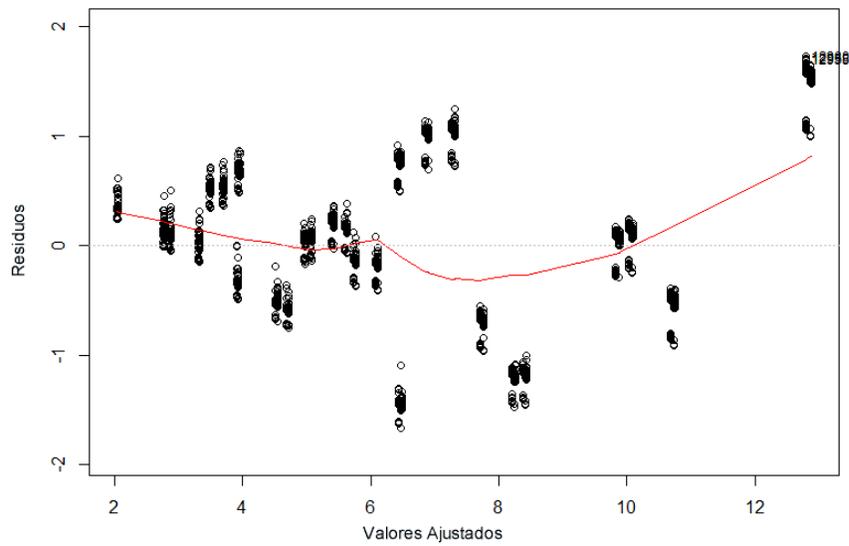


(b) Residuos contra Valores Ajustados de la Transformación con logaritmo base 10.

Figura 4.5: Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo transformados con una logaritmo base 10.



(a) Cuantiles contra Cuantiles de la Transformación con raíz cuadrada.



(b) Residuos contra Valores Ajustados de la Transformación con raíz cuadrada.

Figura 4.6: Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo transformados con una raíz cuadrada.

4.1.1.2. Modelo Lineal Generalizado

El Modelo Lineal Generalizado (GLM por sus siglas en inglés) es una generalización de la regresión lineal, que no toma como un hecho la distribución gaussiana del error en la variable de respuesta[21].

Para utilizar un GLM se necesita:

- Una función de enlace.
- Una distribución de probabilidad para los errores.

La escogencia de estos dos requerimientos hace que nuevamente se hagan supuestos sobre los datos y debido a esto, se descartó.

4.1.1.3. Método Box-Cox

El método Box-Cox[36] es una función de optimización, donde se minimiza la desviación estándar de un conjunto de datos al calcular una transformación de potencia λ . La figura 4.7 muestra el resultado del proceso Box-Cox en R, donde se puede notar que la transformación óptima es cerca de -1 .

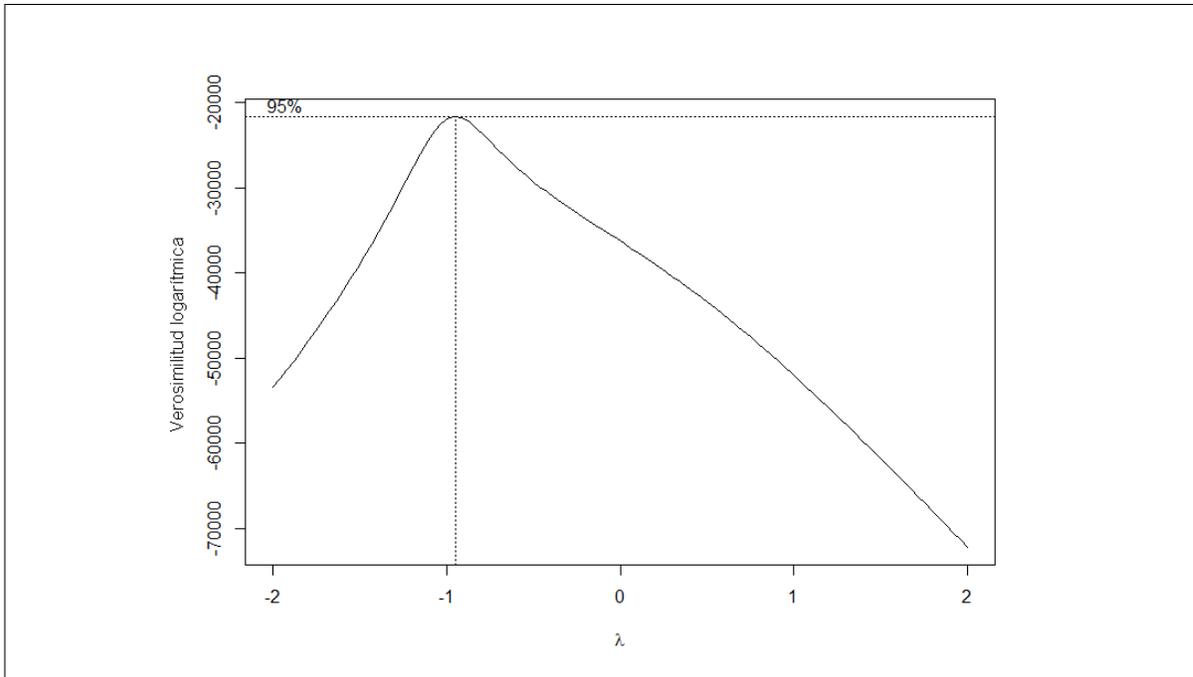
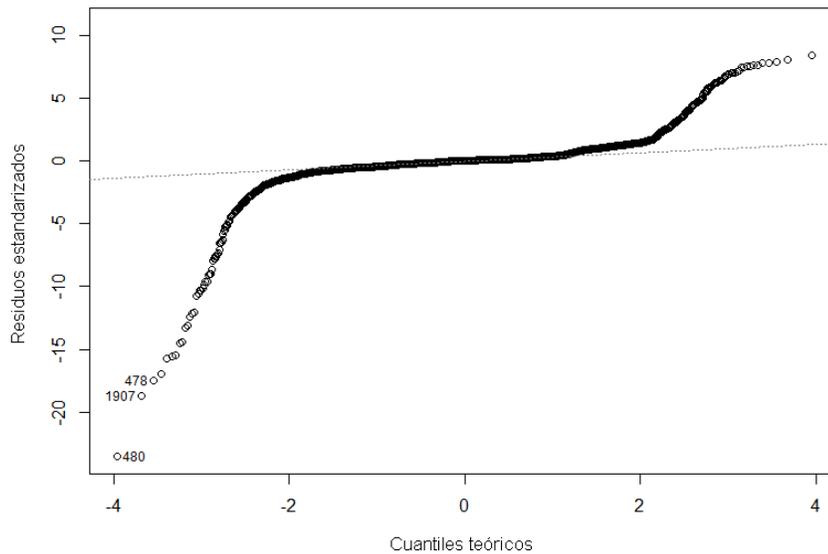


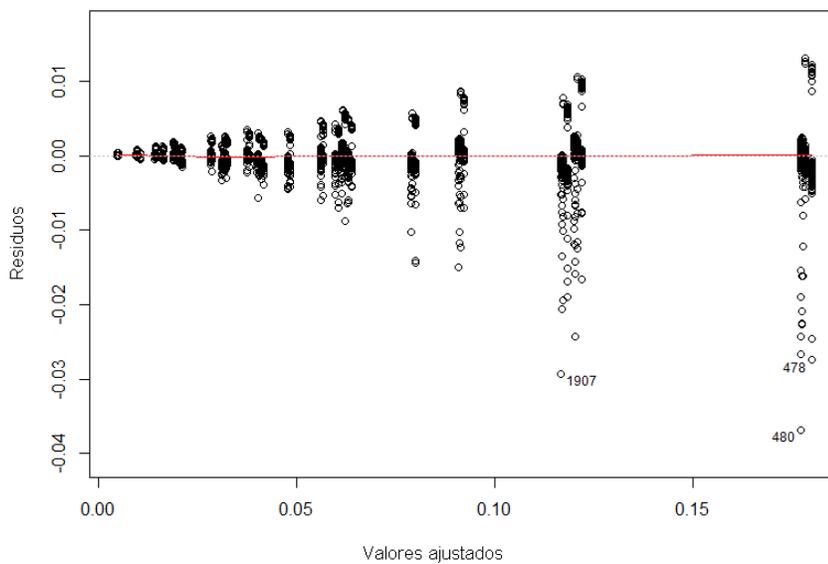
Figura 4.7: Método Box-Cox. El punto óptimo λ , con un 95 % de confianza se encuentra cerca de -1.

Al aplicar una transformación de potencia -1 , la exploración visual nuevamente falla al validar los supuestos de ANOVA. La figura 4.8 muestra el gráfico Q-Q y RvAF de la transformación sugerida por el método Box-Cox.

4.1. EXPERIMENTO A



(a) Cuantiles contra Cuantiles de la Transformación con el método Box-Cox.



(b) Residuos contra Valores Ajustados de la Transformación con el método Box-Cox.

Figura 4.8: Exploración visual de la normalidad y homogeneidad de varianza en los datos de tiempo transformados con el método Box-Cox.

4.1.1.4. Transformación por Rangos

Luego de haber agotado las opciones brindadas por la literatura, se decidió consultar a la USES, donde se concertaron dos reuniones de trabajo donde se decidió por criterio experto hacer una transformación por Rangos, que elimina los supuestos de ANOVA[37][35].

La transformación por rangos es un proceso estadístico que toma el mejor valor de una variable y le asigna el rango de 1, toma el siguiente mejor valor y asigna el rango de 2 sucesivamente hasta el peor valor de la variable y se le asigna el ultimo rango.

Para los tiempos en la TT, un tiempo bajo es un buen valor y un tiempo alto es un valor malo, así, el tiempo de 5,177 segundos tiene el rango 1 y el tiempo de 216,606 segundos, el rango 12960. Esta conversión sigue la función en la gráfica de la figura 4.9.

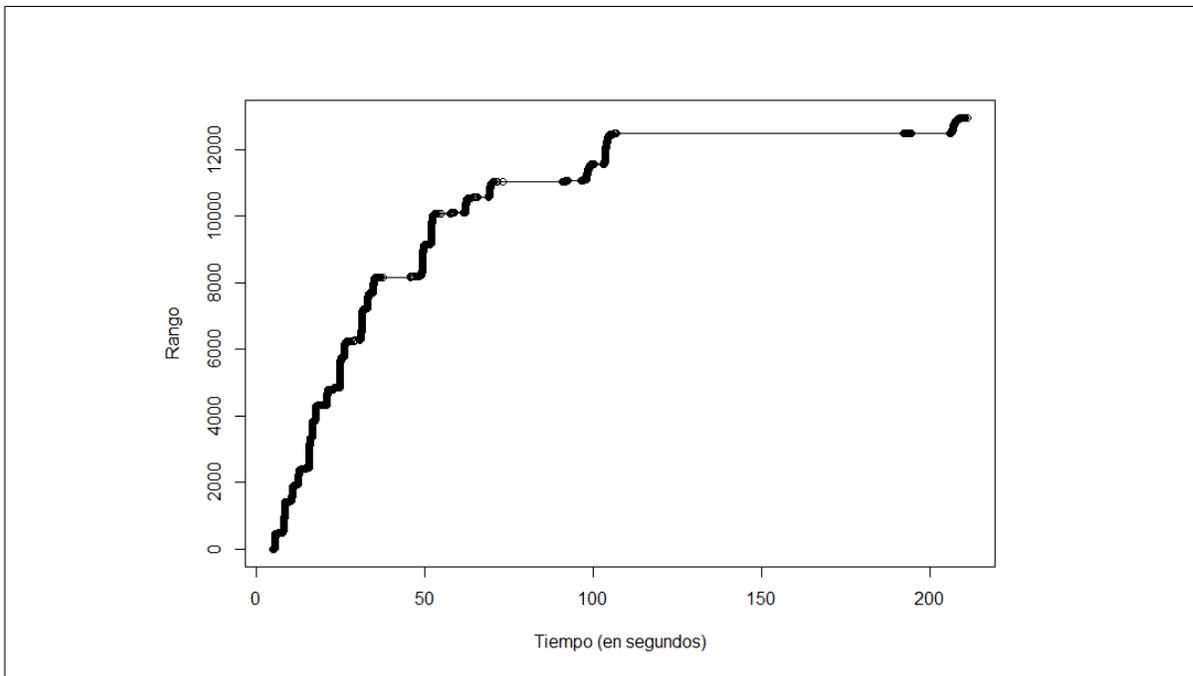


Figura 4.9: Transformación por rangos de los datos de tiempo tomados en el experimento A.

4.1. EXPERIMENTO A

Factor	Df	Sum Sq	Mean Sq	F-value	Pr(>F)	Sig.
dTau	1	6.056e+10	6.056e+10	9.681e+04	<2e-16	0
dPhi	1	7.188e+10	7.188e+10	1.149e+05	<2e-16	0
dRho	1	3.796e+10	3.796e+10	6.068e+04	<2e-16	0
cobertura	4	2.135e+07	5.338e+06	8.533e+00	7.15e-07	0
dTau:dPhi	1	5.291e+07	5.291e+07	8.458e+01	<2e-16	0
dTau:dRho	1	1.383e+08	1.383e+08	2.211e+02	<2e-16	0
dPhi:dRho	1	1.108e+03	1.108e+03	2.000e-03	0.966	
dTau:cobertura	4	2.380e+06	5.951e+05	9.510e-01	0.433	
dPhi:cobertura	4	8.064e+04	2.016e+04	3.200e-02	0.998	
dRho:cobertura	4	1.340e+05	3.351e+04	5.400e-02	0.995	
dTau:dPhi:dRho	1	2.709e+09	2.709e+09	4.330e+03	<2e-16	0
dTau:dPhi:cobertura	4	1.172e+06	2.929e+05	4.680e-01	0.759	
dTau:dRho:cobertura	4	6.292e+04	1.573e+04	2.500e-02	0.999	
dPhi:dRho:cobertura	4	8.207e+05	2.052e+05	3.280e-01	0.859	
dTau:dPhi:dRho:cobertura	4	4.000e+05	1.000e+05	1.600e-01	0.959	
Residuos	12920	8.082e+09	6.256e+05			

Cuadro 4.3: Tabla ANOVA para los rangos de tiempo del experimento A.

4.1.2. Análisis de resultados

Una vez superado los supuestos de ANOVA, se procedió a hacer el análisis de los tiempos transformados a rangos.

El procedimiento `aov` en R es el utilizado para hacer el análisis de varianza. Esta función reporta dos salidas: una tabla de ANOVA y el cuadro resumen, de los cuales la tabla es la que nos interesa.

La figura 4.10 muestra la invocación de la función `aov` y la tabla 4.3 muestra la relevancia de cada combinación de factores.

```
> aov(rank(t) ~ dTau * dRho * dPhi * cobertura)
```

Figura 4.10: Invocación de la función `aov` para Análisis de Varianza en R. El parámetro recibido muestra las relaciones entre las variables: $rank(t)$ la variable de respuesta se debe analizar en función de $\Delta\tau$, $\Delta\rho$, $\Delta\phi$, tipo de cobertura y todas sus combinaciones.

La tabla indica que existen siete combinaciones de factores que influyen sobre la variable de respuesta. Estos son los cuatro factores principales, las interacción de segundo nivel $\Delta\tau$ con $\Delta\rho$ y $\Delta\tau$ con $\Delta\phi$, y la interacción de tercer nivel de los parámetros de frecuencia.

El efecto de los factores principales se evidencia en la figura 4.11. Todos los parámetros de frecuencia muestran un decremento en tiempo mientras más crecen (un rango alto representa un tiempo largo), lo que es esperable: **a más fino el barrido, más tiempo se tarda extraer las características.**

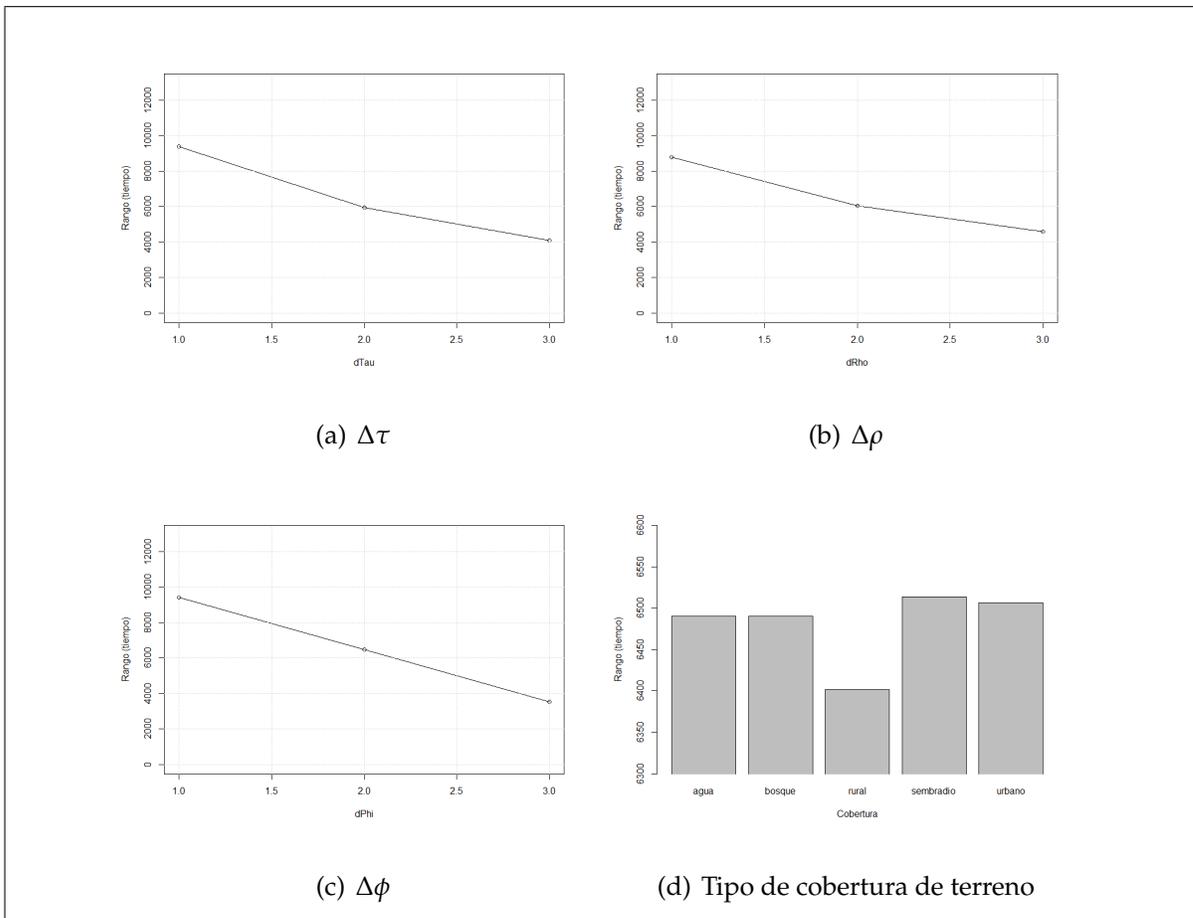


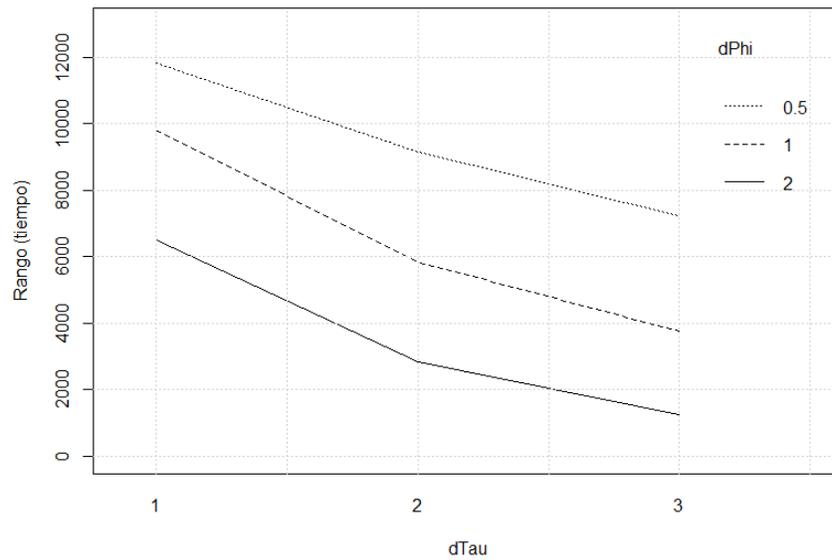
Figura 4.11: Promedio del efecto de los factores principales del experimento A.

El efecto que tiene el tipo de cobertura sobre el tiempo es poco, pero dado el análisis, es significativo. El tipo de cobertura rural, aunque muestre un mejor tiempo en promedio, la diferencia es ínfima: del rango 6400 al 6500 la diferencia es de

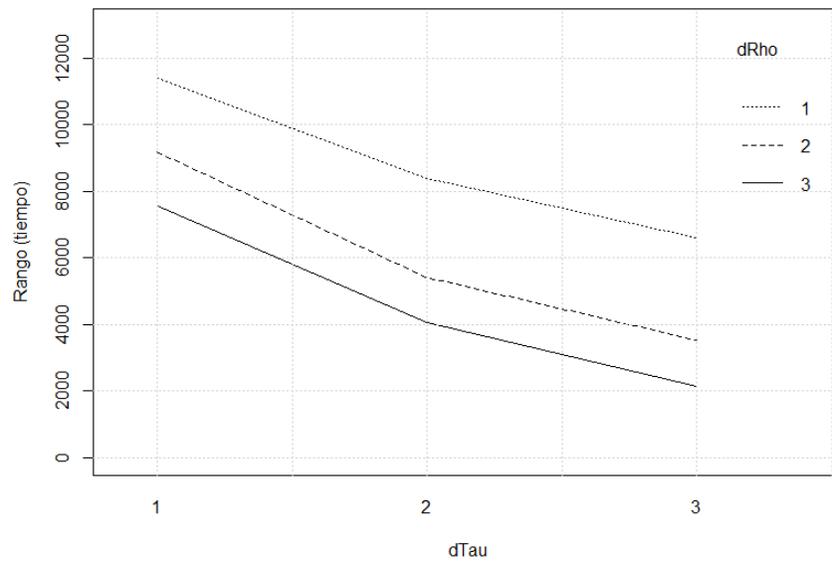
4.1. EXPERIMENTO A

0,03033333 segundos.

Las interacciones de segundo nivel se muestran en la figura 4.12. La subfigura (a) expone el promedio de los rangos cuando $\Delta\tau$ y $\Delta\phi$ cambian juntos, de igual manera pero con $\Delta\rho$ en la subfigura (b).



(a) $\Delta\tau$ y $\Delta\phi$



(b) $\Delta\tau$ y $\Delta\rho$

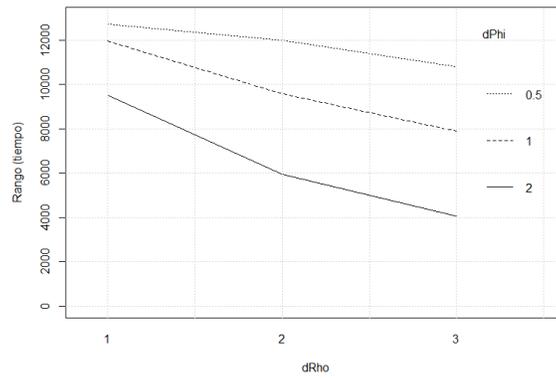
Figura 4.12: Promedio de interacciones de segundo nivel en el experimento A.

En ambos, se muestra un comportamiento decreciente en los rangos y tiempos a como se incrementan los parámetros de frecuencia, siendo los puntos extremos

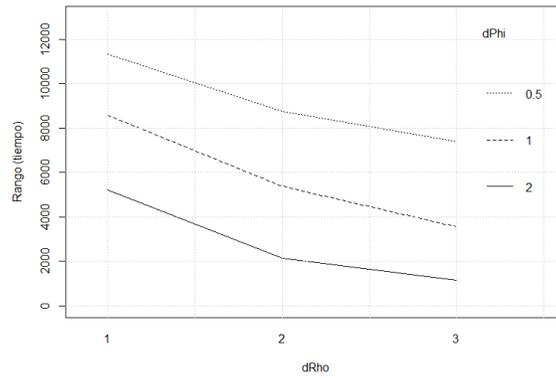
nuevamente las combinaciones $\Delta\tau = 3, \Delta\phi = 2$ y $\Delta\tau = 3, \Delta\rho = 3$. El efecto es ligeramente más marcado con $\Delta\phi$, donde los rangos son más bajos.

Por último, la interacción de tercer nivel se muestra en la figura 4.13. Esta interacción toma en cuenta todas las variaciones de los parámetros de frecuencia y deja por fuera el tipo de cobertura de terreno.

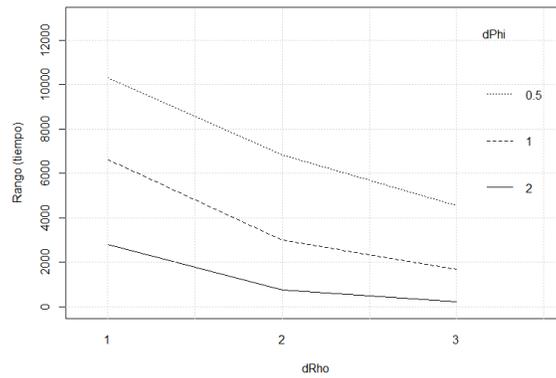
En todos los casos el comportamiento no es una sorpresa, es decreciente en bloques: con $\Delta\tau = 1$ se tarda más en promedio, con cualquier valor de $\Delta\rho$ y $\Delta\phi$, que con $\Delta\tau = 3$. Esto significa, que al igual que como se indican con las interacciones anteriores, el tiempo de extracción decrece a como los parámetros aumentan ya sea de manera independiente o conjunta.



(a) $\Delta\tau=1$ y $\Delta\rho$ y $\Delta\phi$



(b) $\Delta\tau=2$ y $\Delta\rho$ y $\Delta\phi$



(c) $\Delta\tau=3$ y $\Delta\rho$ y $\Delta\phi$

Figura 4.13: Promedio de la interacción de tercer nivel para el experimento A.

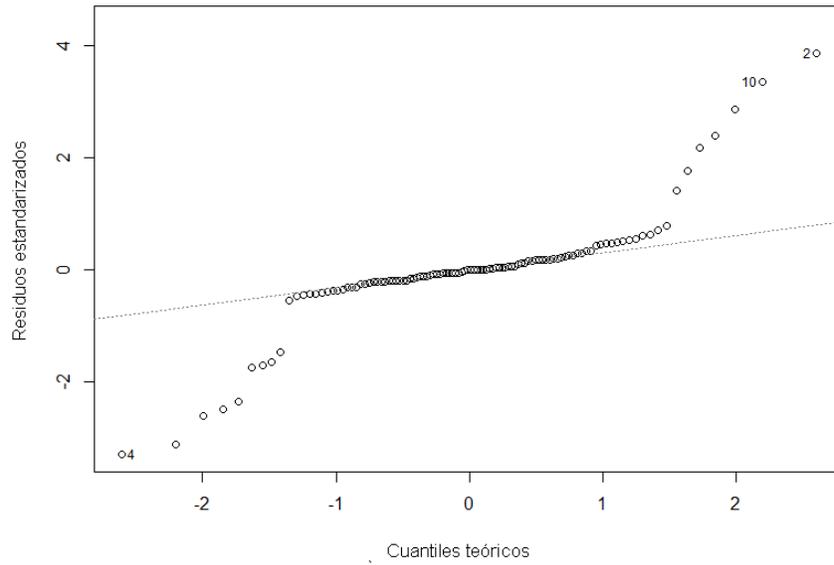
4.2. Experimento B

4.2.1. Verificación de supuestos

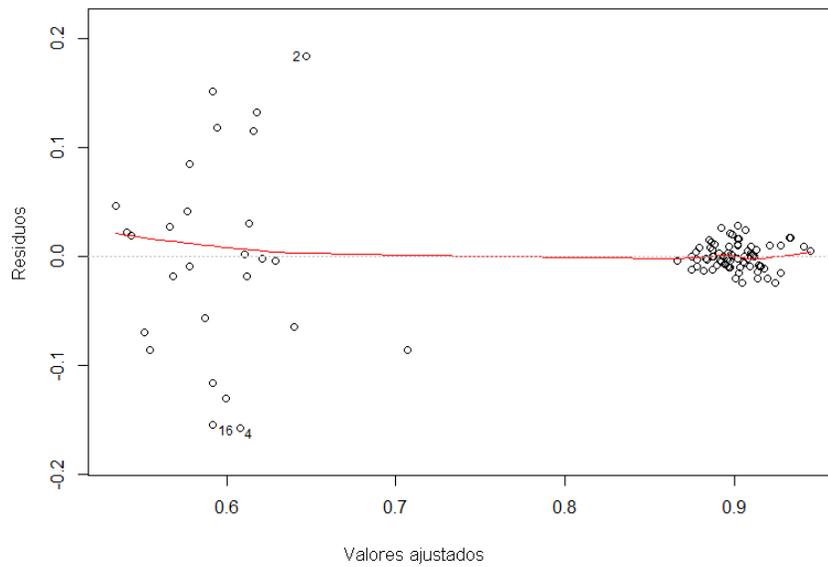
Al igual que con el experimento A, se realizó una exploración visual que rechazó los supuestos de ANOVA, como se indica en la figura 4.15. También se confirmó con la prueba de Lilliefors como lo demuestra la figura 4.14; el estadístico calculado descarta normalidad, por lo que se prosiguió a aplicar transformaciones a los datos.

```
> lillie.test(data-p)
      Lilliefors (Kolmogorov-Smirnov) normality test
data:  data-p
D = 0.3546, p-value < 2.2e-16
```

Figura 4.14: Prueba de Lilliefors para normalidad en R. El estadístico es mayor que el valor-p por lo que los datos no son normales.



(a) Cuantiles contra Cuantiles.



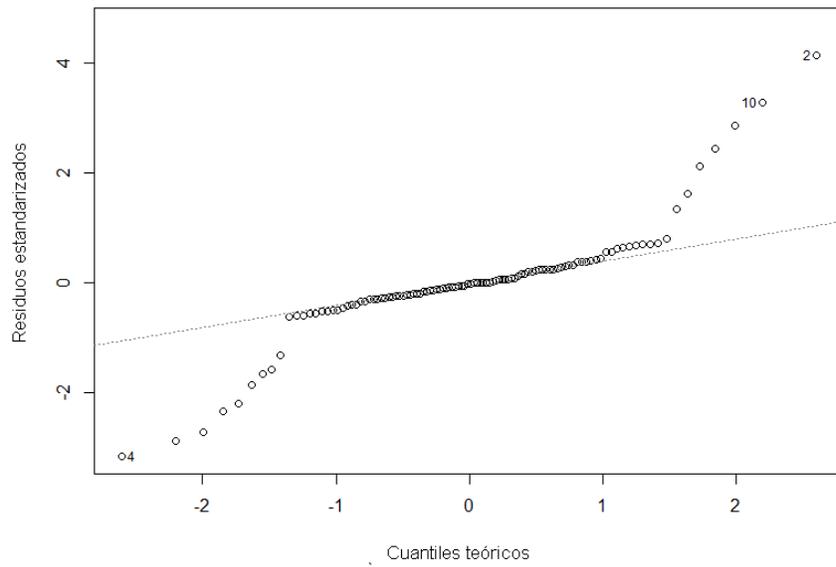
(b) Residuos contra Valores Ajustados.

Figura 4.15: Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión.

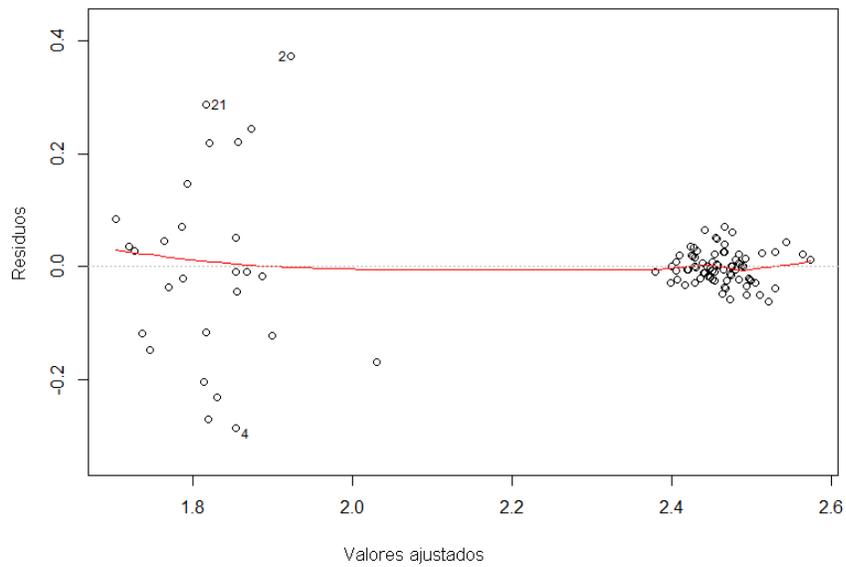
4.2.1.1. Transformaciones

Las transformaciones aplicadas a los datos de tiempo fueron aplicadas a los datos de precisión: potencia e , raíz cuadrada y logaritmo base 10.

En todos los casos, los supuestos fueron rechazados. Para cada transformación, potencia e , raíz cuadrada y logaritmo base 10, se muestran los gráficos Q-Q y RvAF, respectivamente en las figuras 4.16 4.17 4.18.



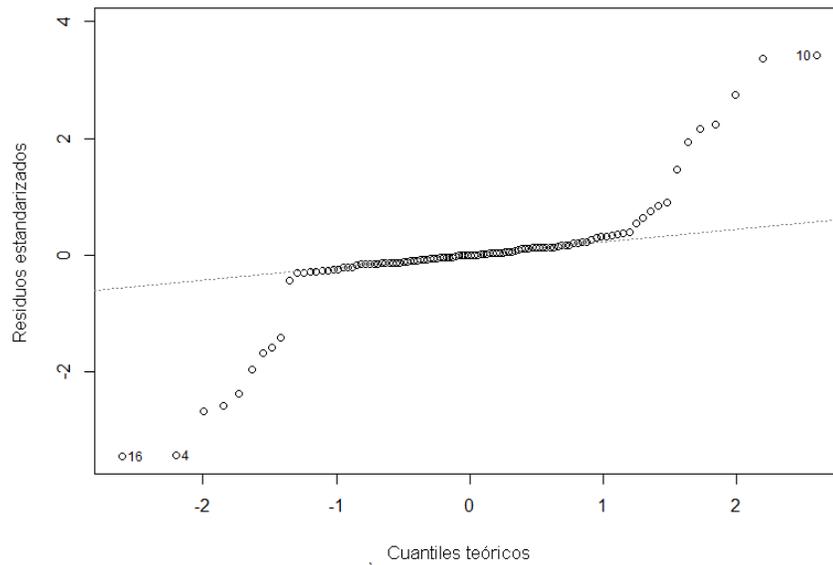
(a) Cuantiles contra Cuantiles de la Transformación con potencia e .



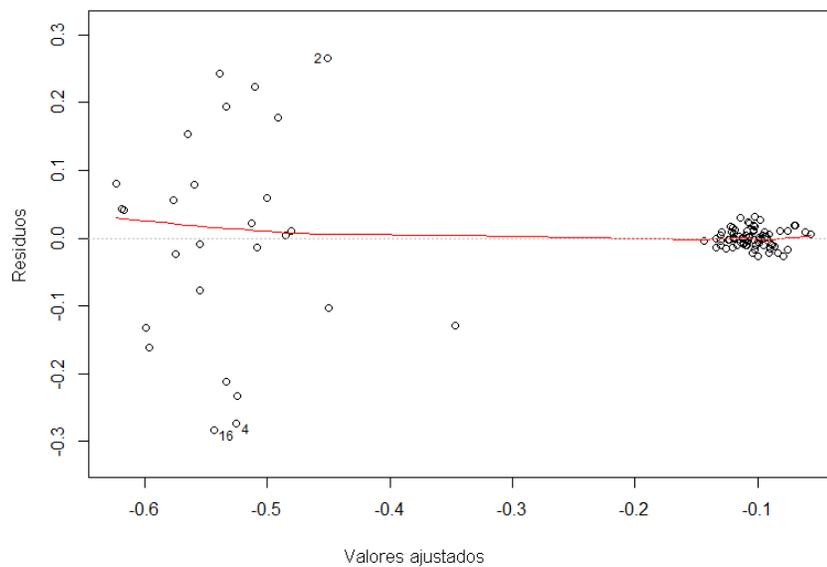
(b) Residuos contra Valores Ajustados de la Transformación con potencia e .

Figura 4.16: Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión transformados con una potencia e .

4.2. EXPERIMENTO B

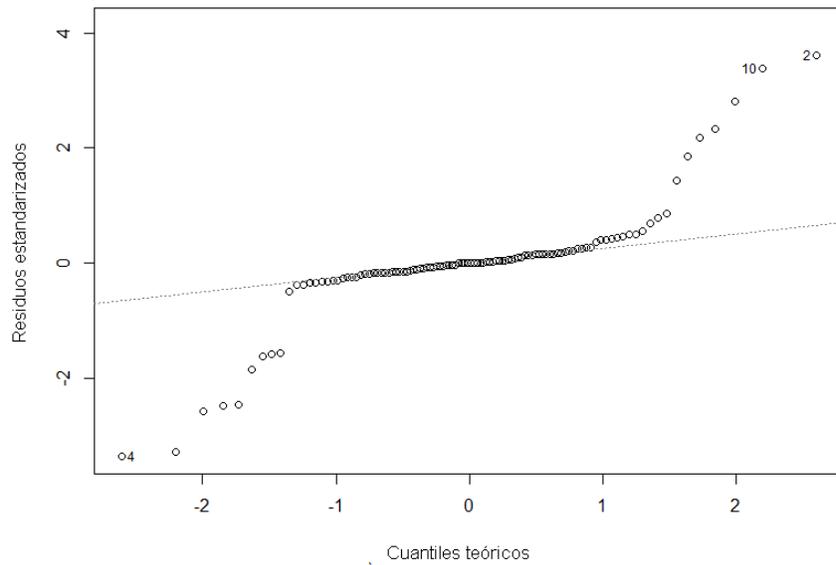


(a) Cuantiles contra Cuantiles de la Transformación con logaritmo base 10.

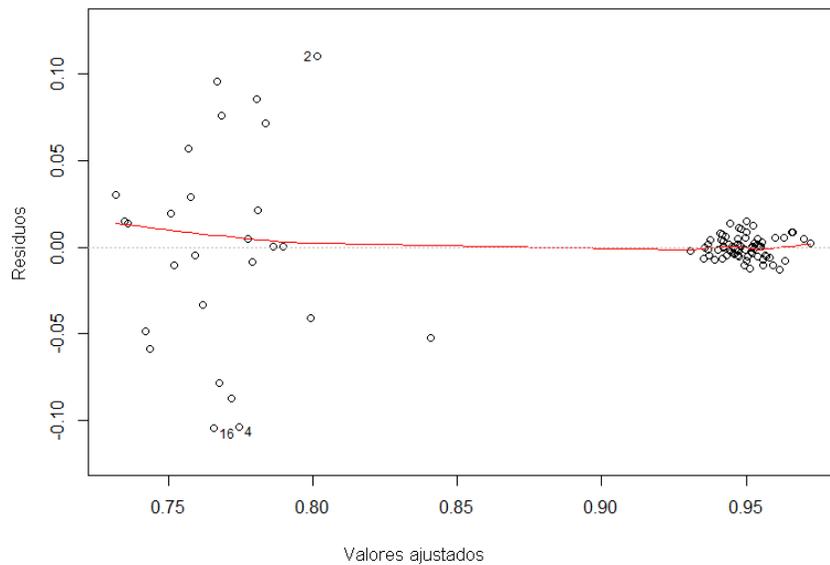


(b) Residuos contra Valores Ajustados de la Transformación con logaritmo base 10.

Figura 4.17: Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión transformados con una logaritmo base 10.



(a) Cuantiles contra Cuantiles de la Transformación con raíz cuadrada.



(b) Residuos contra Valores Ajustados de la Transformación con raíz cuadrada.

Figura 4.18: Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión transformados con una raíz cuadrada.

4.2.1.2. Modelo Lineal Generalizado

El GLM fue descartado dadas las mismas razones que se plantean en el experimento A: no se pueden afirmar supuestos sobre distribución de errores de los datos de precisión ni establecer una función de enlace sin que sea de forma arbitraria.

Al igual que en el experimento A, se procedió a aplicar el método Box-Cox con la expectativa de que dicha transformación confirme los supuestos.

4.2.1.3. Método Box-Cox

El método Box-Cox reporta un valor λ óptimo cerca de 2, como lo muestra la figura 4.19, por lo que se aplicó una transformación de potencia 2 a los datos.

Infortunadamente la transformación rechaza los supuestos nuevamente. La figura 4.20 muestra la exploración visual que descarta normalidad y homogeneidad de varianzas.

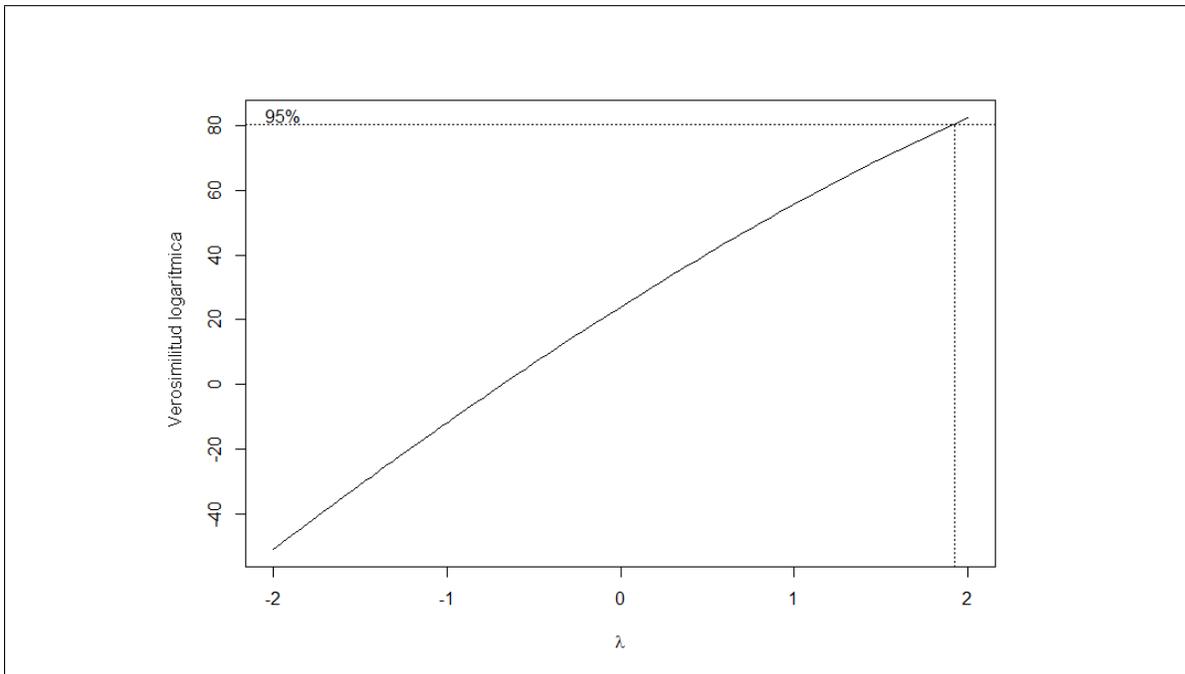
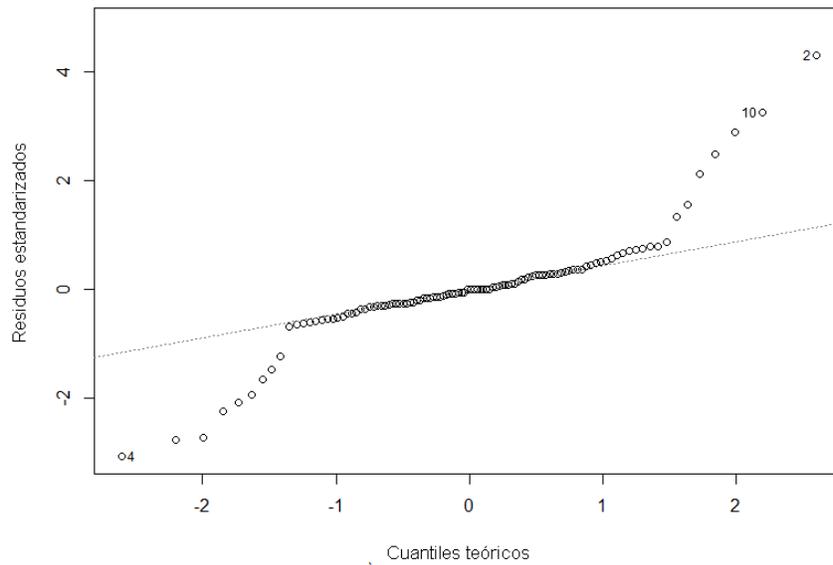
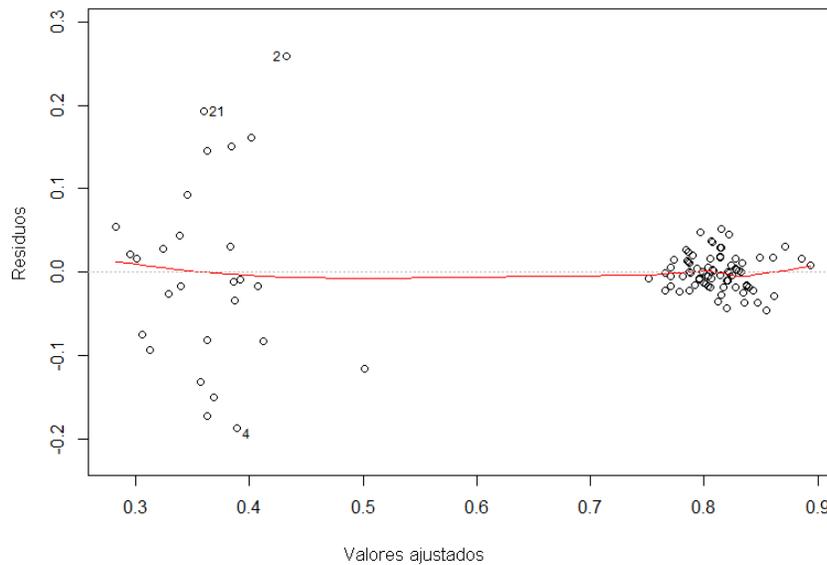


Figura 4.19: Método Box-Cox. El punto óptimo λ , con un 95 % de confianza se encuentra cerca de 2.



(a) Cuantiles contra Cuantiles de la Transformación con el método Box-Cox.



(b) Residuos contra Valores Ajustados de la Transformación con el método Box-Cox.

Figura 4.20: Exploración visual de la normalidad y homogeneidad de varianza en los datos de precisión transformados con el método Box-Cox.

4.2. EXPERIMENTO B

Dados estos fallos al verificar los supuestos, se optó por la misma solución que en el experimento A y aplicar una transformación por rangos.

4.2.1.4. Transformación por Rangos

Se aplicó en este caso una transformación por rangos donde un rango bajo significa una mala precisión, y un rango alto, una buena precisión (inverso al caso de los tiempos).

Así la peor precisión de 0,4375 tendrá el rango 1 y la mejor precisión 0,95 tendrá el rango más alto 212,5. Esta transformación se puede ver en la figura 4.21.

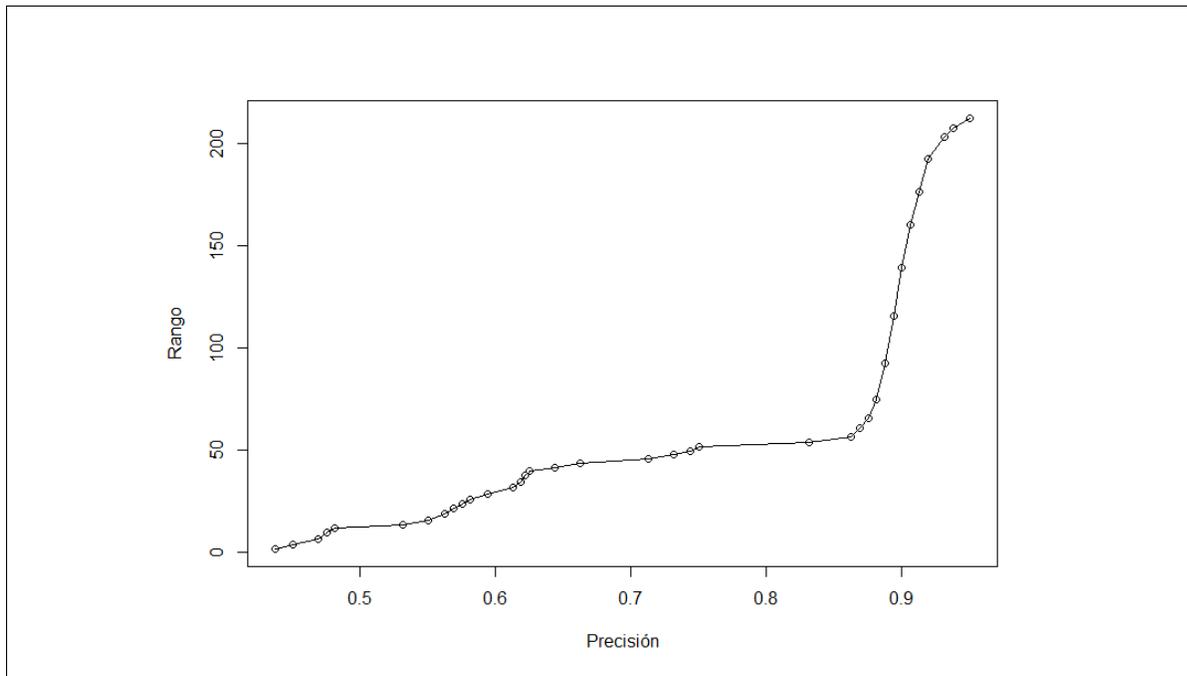


Figura 4.21: Transformación por rangos de los datos de precisión tomados en el experimento B.

4.2.2. Análisis de Resultados

La tabla ANOVA se muestra en la figura 4.4. Es preciso recordar que para este experimento, la transformación de los datos hizo que un rango bajo signifique una precisión alta (se puede ver también como el error en la clasificación).

Factor	Df	Sum Sq	Mean Sq	F-value	Pr(>F)	Sig.
dTau	1	1168	1168	4.622	0.034752	0.05
dPhi	1	89	89	0.353	0.553947	
dRho	1	133	133	0.528	0.469773	
clasificador	3	72454	24151	95.560	<2e-16	0
dTau:dPhi	1	58	58	0.230	0.632989	
dTau:dRho	1	248	248	0.979	0.325495	
dPhi:dRho	1	0	0	0.000	0.989759	
dTau:clasificador	3	5793	1931	7.641	0.000157	0
dPhi:clasificador	3	1384	461	1.825	0.149661	
dRho:clasificador	3	1970	657	2.598	0.058362	0.1
dTau:dPhi:dRho	1	462	462	1.827	0.180523	
dTau:dPhi:clasificador	3	664	221	0.876	0.457247	
dTau:dRho:clasificador	3	62	21	0.082	0.969620	
dPhi:dRho:clasificador	3	208	69	0.275	0.843459	
dTau:dPhi:dRho:clasificador	3	474	158	0.625	0.601146	
Residuals	76	19208	253			

Cuadro 4.4: Tabla ANOVA para la precisión en la clasificación.

Los factores e interacciones relevantes fueron: $\Delta\tau$, el clasificador y la interacción de segundo nivel entre ellos.

El factor $\Delta\tau$ presenta un efecto tenue sobre la variable de respuesta, sin importar su variación, la precisión se mantiene casi constante. La gráfica en la figura 4.22 (a) muestra una línea prácticamente horizontal de $\Delta\tau$ contra los rangos de la precisión. Por otro lado la subfigura (b) muestra el tipo de clasificador contra los rangos. Aquí las diferencias son muy marcadas entre ellos. En promedio LogitBoost y SMO superan a los otros clasificadores, con rangos promedios de 163,64 (precisión de 0,9134259) y 142,90 (precisión de 0,9023148) respectivamente, y Kmeans es el peor de los cuatro con un rango promedio de 27,5 (tan solo 0,5952315 de precisión).

4.2. EXPERIMENTO B

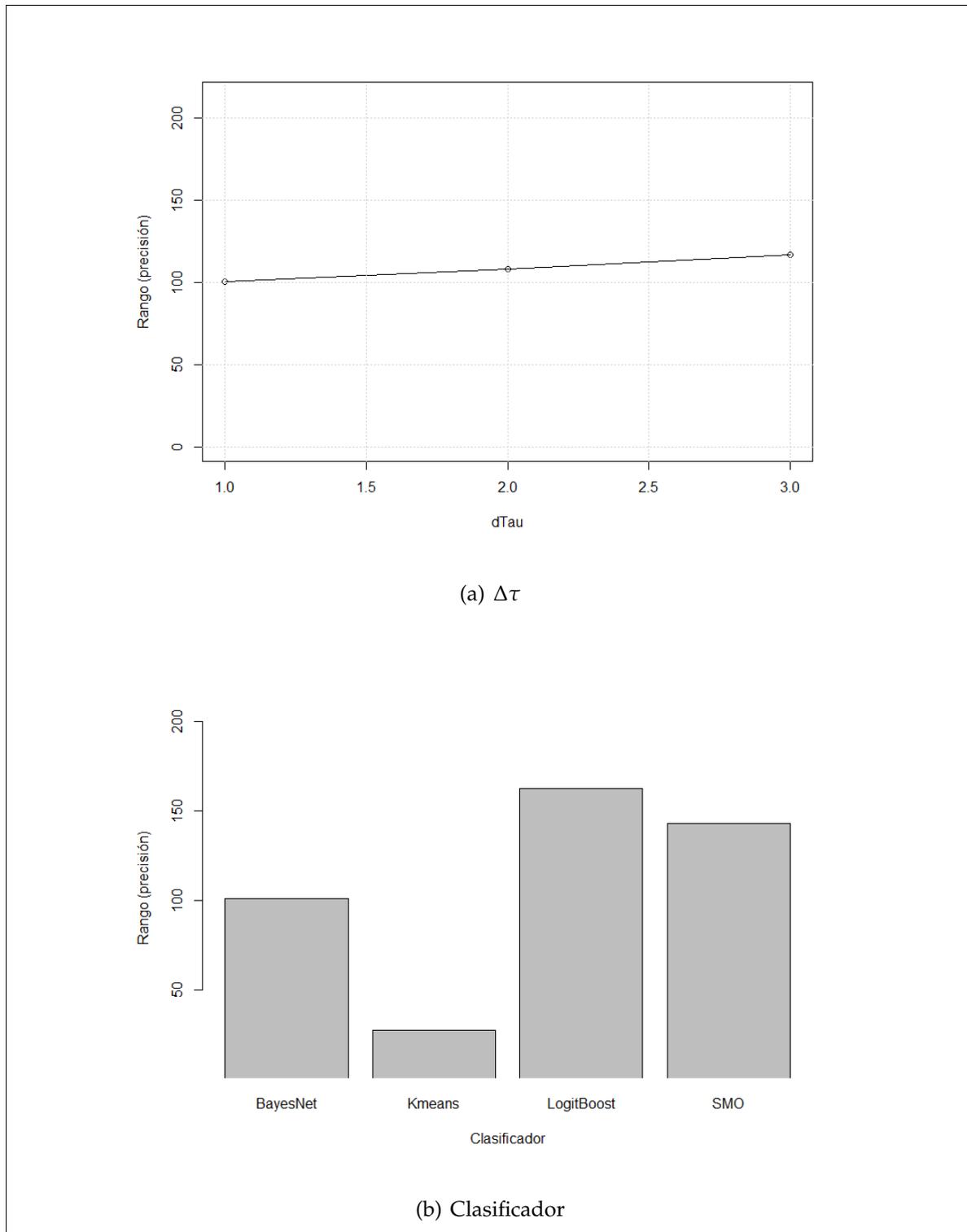


Figura 4.22: Promedio del efecto de los factores $\Delta\tau$ y clasificador sobre la precisión.

La interacción de segundo nivel entre $\Delta\tau$ y el clasificador se puede apreciar en la figura 4.23. Aquí se puede notar claramente el gran peso que tiene el clasifica-

dor sobre la precisión, donde se pueden notar dos picos muy marcados, uno bajo en Kmeans y uno alto en LogitBoost. La diferencia entre cada uno de los clasificadores en términos de $\Delta\tau$ no es muy fuerte, especialmente con Kmeans. Se da un comportamiento interesante, en todos los clasificadores menos en Kmeans, donde la diferencia entre los promedios varía mucho con respecto a $\Delta\tau$.

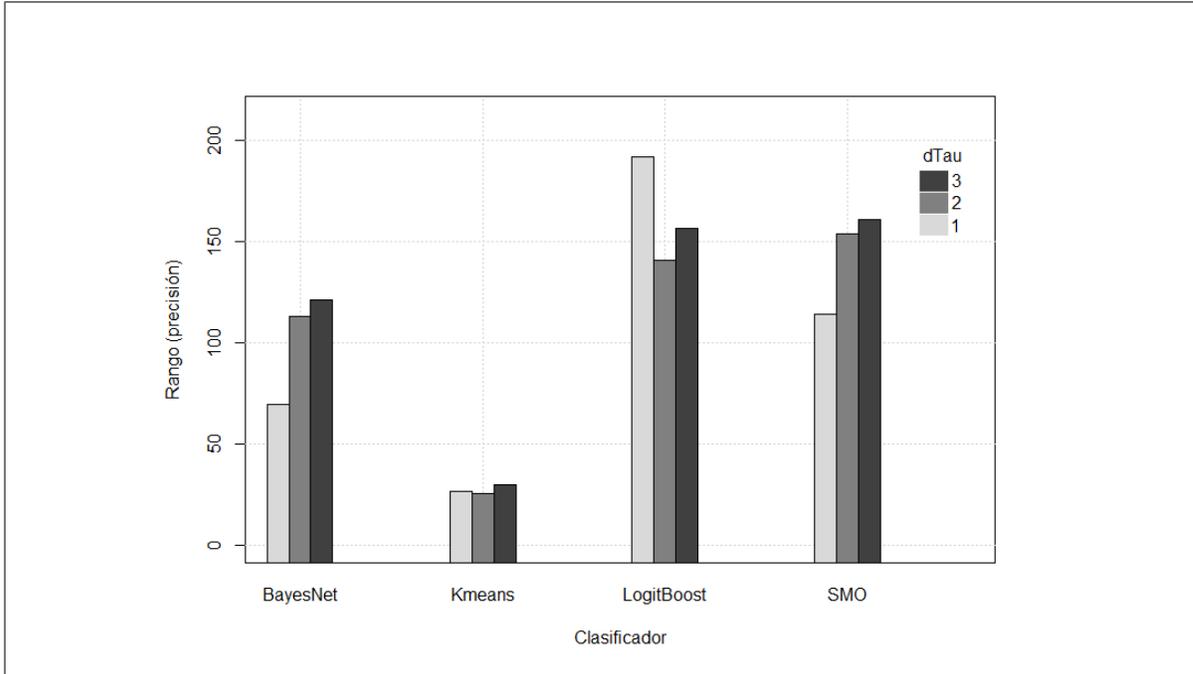


Figura 4.23: Promedio de la interacción de segundo nivel entre $\Delta\tau$ y el clasificador para la precisión en el experimento B.

La figura 4.24 grafica los clasificadores contra la cantidad de píxeles en promedio usados en cada configuración de parámetros de frecuencia. La cantidad promedio de píxeles utilizada se calcula con la siguiente fórmula:

$$\text{promedio píxeles} = \frac{\text{trazMax} + \text{trazMin}}{2} * \frac{\text{numTraz}}{\Delta\tau} \quad (4.2)$$

Donde:

- *trazMax* es el trazo más largo que se puede hacer sobre la imagen, en este caso 500.
- *trazMin* es el trazo más corto que se puede hacer sobre la imagen, en este caso

4.2. EXPERIMENTO B

1.

- *numTraz* es la cantidad de trazos que se hacen sobre la imagen según $\Delta\rho$ y $\Delta\phi$ y se calcula como se muestra a continuación:

$$\text{numTraz} = \frac{360}{\Delta\phi} * \frac{\text{trazMax}}{\Delta\rho} \quad (4.3)$$

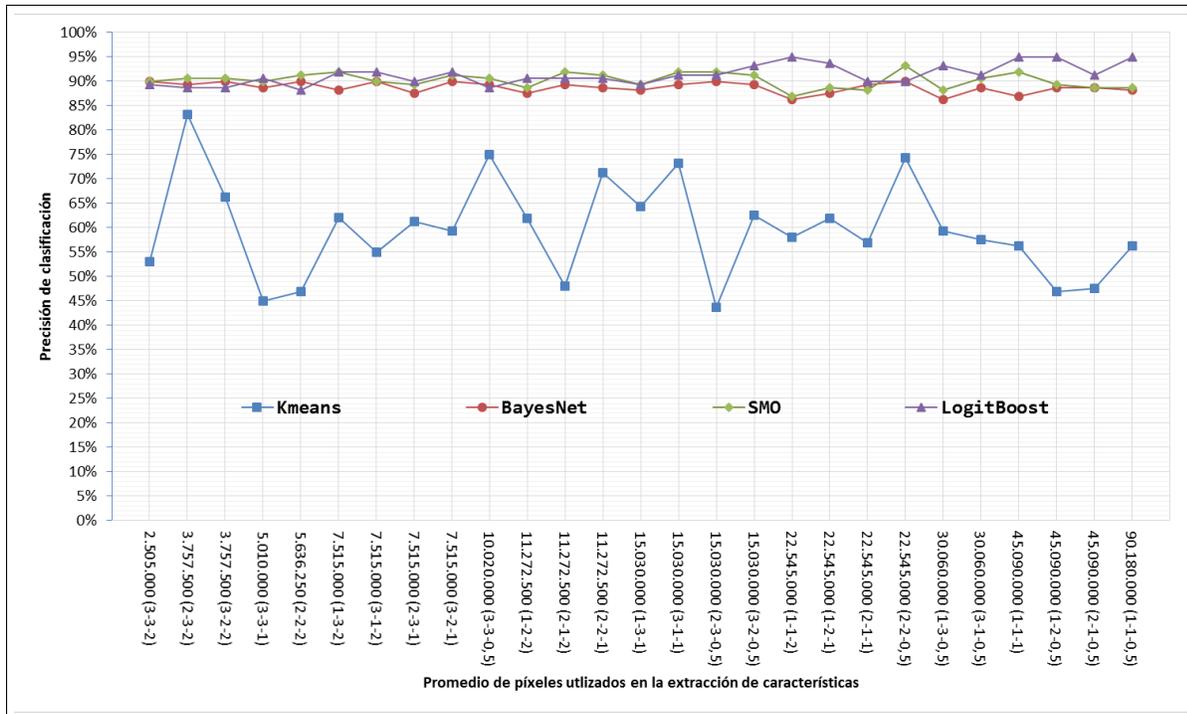


Figura 4.24: Promedio del efecto de los factores $\Delta\tau$ y clasificador sobre la precisión.

En dicha figura se puede notar que, independientemente de la cantidad de píxeles utilizados para la extracción (por ende los parámetros de frecuencia), los clasificadores se comportan de manera similar y con poca variabilidad, con la excepción de Kmeans. LogitBoost oscila entre 0,88125 y 0,95, BayesNet entre 0,8625 y 0,9 y SMO entre 0,86875 y 0,93125.

CAPÍTULO 5

CONCLUSIONES Y TRABAJO FUTURO

A lo largo del presente trabajo se ha evaluado el desempeño de la Transformada de Trazo en términos de tiempo de extracción de características y precisión de clasificación.

Los experimentos aquí realizados pretenden aportar evidencia estadística favor de la hipótesis de investigación: *La Transformada de Trazo presenta mejores resultados estadísticamente, en precisión de clasificación y en tiempo de extracción de características, al variar los parámetros de frecuencia, el tipo de clasificador y el tipo de cobertura.* Se demostró estadísticamente como los parámetros de frecuencia influyen el tiempo de extracción y no así el tipo de cobertura de terreno. También se demostró como el clasificador utilizado afecta la precisión pero no así, todos los parámetros de frecuencia.

A partir de los datos expuestos por los experimentos, se llega a las siguientes conclusiones:

1. A la luz del experimento A, **a parámetros de frecuencia más finos, mayor el tiempo de extracción.** La combinación de parámetros $\Delta\tau = 1$, $\Delta\rho = 1$ y $\Delta\phi =$

-
- 0,5 resultan en el mayor tiempo de extracción y la combinación $\Delta\tau = 3$, $\Delta\rho = 3$ y $\Delta\phi = 2$ en la menor.
2. De los parámetros de frecuencia, el que carga más peso para el tiempo de extracción es $\Delta\tau$: está presente en todas las interacciones reportadas como significativas. Esto hace que se necesite un particular cuidado al escoger este parámetro ya que los tiempos de respuesta pueden variar de 17,66167 a 52,60333 segundos en promedio.
 3. El tipo de **cobertura** de terreno, a pesar de ser significativo estadísticamente cuando se transformó a rangos, **presenta mínimo efecto sobre el tiempo de extracción.**
 4. El experimento B señala que **la precisión depende casi exclusivamente del clasificador utilizado**, donde el método aditivo LogitBoost presenta el mejor rendimiento.
 5. De los parámetros de frecuencia, $\Delta\tau$ nuevamente se presenta como un factor relevante luego de transformar los datos a rangos, aunque en términos de **porcentaje de precisión la diferencia que hace escoger un valor u otro de $\Delta\tau$ es mínima.**
 6. Según los datos experimentales, **no es útil hacer un barrido fino.** Los clasificadores se comportan de manera similar, a excepción de Kmeans, sea grueso o fino el barrido de la imagen. Las ganancias que pueden presentar los otros tres clasificadores, cerca del 7% **en precisión no justifica invertir 42 veces más tiempo en la extracción.** Aunque esta decisión tampoco debe tomarse a la ligera, pues depende del área de aplicación de la TT: tal vez en clasificación de imágenes áreas 7% no sea de gran importancia, pero si las imágenes a procesar fueran de una tomografía axial computarizada para detectar cáncer, ese pequeño porcentaje podría salvar vidas.
 7. Dados los tiempos y la precisión observada, es posible construir una versión

de la TT que utilice valores gruesos de frecuencia y un clasificador LogitBoost que corra en un equipo portátil para giras de campo (geografía).

5.1. Aportes

Dentro del alcance de esta propuesta se definen los siguientes entregables:

1. Prototipo de un programa para la extracción de características usando la TT. En este prototipo deben ser parametrizables, los parámetros de frecuencia y el clasificador.
2. Prototipo de los clasificadores.
3. Programas auxiliares para la ejecución y el control de los 2 anteriores.
4. Análisis estadístico para contrastar los resultados de los experimentos.
5. Artículo científico que se entregará al comité editorial de alguna revista o conferencia, con miras a su publicación.

El trabajo realizado en esta investigación cumplió con los entregables establecidos:

1. Una implementación de la TT parametrizada.
2. Una implementación de los clasificadores, utilizando el sistema Weka.
3. Los programas auxiliares para la ejecución y el control de los experimentos.
4. El análisis estadístico de los experimentos.
5. Un artículo científico, que fue aceptado para presentación y publicación en el IWOBI 2015¹.

5.2. Trabajo futuro

Luego de realizar esta investigación surgen varias preguntas de investigación surgen y quedan abiertos varios caminos para plantear nuevos proyectos. Se propo-

¹<http://iwobi.ulpgc.es/>

nen entonces las siguientes ideas para trabajos futuros:

1. Construir un generador de mapas de cobertura, utilizando la clasificación dada por el estudio aquí hecho y ajustado para maximizar la precisión y minimizar el tiempo de extracción de características.
2. Dadas las diferentes arquitecturas de procesadores y jerarquías de memoria, la medición y adaptación de la TT a otras plataformas de ejecución sería de interés para optimización.
3. La aplicación en imágenes más grandes y en otros campos, especialmente en las ciencias médicas, es de gran interés; existe un gran conjunto de datos de imágenes médicas que se puede explotar y que pueden tener un gran impacto en la calidad de vida.
4. El uso de una ventana de análisis en imágenes grandes se debe estudiar también, y contrastar con la técnica de segmentar la imagen en otras más pequeñas. Se puede a su vez medir el impacto que puede tener ésta ventana en la precisión.
5. La selección de funcionales y por lo tanto de TF es un tema que se debe abordar. Es posible que el aporte que brindan las TF no sea el mismo para la clasificación, y que un número mejor de funcionales sea posible de implementar sin perder precisión.
6. Se debe comparar el rendimiento de la TT contra otras técnicas automatizadas de clasificación de imágenes, particularmente en el área geográfica.

BIBLIOGRAFÍA

- [1] BIOMARCC-SINAC-GIZ, "Clasificación sistemas marino costeros costa pacífica de costa rica," 2012. http://www.biomarcc.org/download_PDF/SerieTecnica2_SistemasMarinos.pdf.
- [2] P. Zatelli, "Thematic maps," 2014. http://www.ing.unitn.it/~zatelli/environmental_data_management/Thematic_maps.pdf.
- [3] E. Chuvieco, *Teledetección ambiental: la observación de la tierra desde el espacio*. Editorial Ariel, 1era ed., 2010.
- [4] PRIAS/CeNAT, "Fotografías aéreas proyecto carta," *Distribuido por PRIAS-CeNAT*, 2005. 1.3 km al Norte de la embajada de U.S.A. Rohrmoser, Pavas 2519-5709.
- [5] SINAC-MINAE, "Redd+: Corrección atmosférica de imagenes rapid eye," *Adquirido por Gesellschaft für Internationale Zusammenarbeit/Propiedad del SINAC-MINAE*, 2012. 1.3 km al Norte de la embajada de U.S.A. Rohrmoser, Pavas 2519-5709.
- [6] A. Kadyrov and M. Petrou, "The trace transform and its applications," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 23, pp. 811–828, Agosto 2001.
- [7] B.-S. Shin, E.-Y. Cha, K.-B. Kim, K.-W. Cho, R. Klette, and Y. W. Woo, "Effective feature extraction by trace transform for insect footprint recognition," in *3rd International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 97 – 102, IEEE, sep 2008.
- [8] S. Srisuký, K. Ratanarangsanký, W. Kurutach, and S. Waraklang, "Face recognition using a new texture representation of face images," pp. 1097–1102,, Noviembre 2003.

- [9] M. Petrou, A. Talebpour, and A. Kadyrov, "Reverse engineering the way humans rank textures," *Pattern Analysis and Applications*, vol. 10, no. 2, pp. 101–114, 2007.
- [10] R. Garita and A. De la Ossa, "Paralelización del método de la transformada de trazo para el análisis de texturas en imágenes aéreas," in *6a Conferencia Latinoamericana de Computación de Alto Rendimiento*, 2013, En prensa.
- [11] M. Petrou and F. Wang, "A tutorial on the practical implementation of the trace transform," in *Handbook of Texture Analysis* (X. X. S. J. Mirmehdi, M., ed.), Oxford: Oxford University Press, 2008.
- [12] E. Alpaydin, *Introduction to Machine Learning*. MIT Press, 2da ed., 2010.
- [13] A. Sayeed, M. Petrou, and R. Maguire, "Diagnostic features of alzheimer's disease extracted from fdg pet images," 2001.
- [14] A. Kadyrov and M. Petrou, "Affine parameter estimation from the trace transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 1631–1645, Octubre 2006.
- [15] W. A. Albukhanajer, Y. Jin, J. A. Briffa, and G. Williams, "A comparative study of multi-objective evolutionary trace transform methods for robust feature extraction," in *Evolutionary Multi-Criterion Optimization* (R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, eds.), vol. 7811 of *Lecture Notes in Computer Science*, pp. 573–586, Springer Berlin Heidelberg, 2013.
- [16] A. Kadyrov and M. Petrou, "Object signatures invariant to affine distortions derived from the trace transform," *Image and Vision Computing*, vol. 21, pp. 1135–1143, 2003.
- [17] M. van Ginkel, C. Luengo, and L. van Vliet, "A short introduction to the radon and hough transforms and how they relate to each other," Tech. Rep. QI-2004-01, Delft University of Technology, 2004.
- [18] B.-S. Shin, E.-Y. Cha, and R. Klette, "Segmentation of scanned insect footprints using art2 for threshold selection," Tech. Rep. LNCS 4872, The University of Auckland, 2007.
- [19] J. Turan, Z. Bojkovic, P. Filo, and L. Ovsenik, "Invariant image recognition experiment with trace transform," in *7th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, 2005.
- [20] J. Zhang, Y. Yan, and M. Lades, "Face recognition: eigenface, elastic matching, and neural nets," *Proceedings of the IEEE*, vol. 85, pp. 1423–1435, Sep 1997.
- [21] D. Montgomery, *Design and Analysis of Experiments*. John Wiley & Sons, 2001.

- [22] R. Garita, *Paralelización del método de la Transformada de Trazo en dispositivos de memoria compartida para la identificación del tipo de cobertura de terreno en imágenes aéreas*. Universidad de Costa Rica, 2014.
- [23] ImageMagick Studio LLC, *ImageMagick: MagickWand, C API for ImageMagick: Image Methods*. ImageMagick Studio LLC, 2015. <http://www.imagemagick.org/api/magick-image.php#MagickQuantizeImage>.
- [24] ImageMagick Studio LLC, *ImageMagick: Color Reduction Utilizing Adaptive Spatial Subdivision*. ImageMagick Studio LLC, 2015. <http://www.imagemagick.org/script/quantize.php>.
- [25] I. H. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3ra ed., 2011.
- [26] A. Puurula, “Cumulative progress in language models for information retrieval,” in *Proc 11th Australasian Language Technology Workshop*, Brisbane, Australia, pp. 96–100, ACL, 2013.
- [27] J. Rajanayagam, E. Frank, R. W. Shepherd, and P. J. Lewindon, “Artificial neural network is highly predictive of outcome in paediatric acute liver failure,” *Pediatric Transplantation*, 2013.
- [28] O. Medelyan, S. Manion, J. Broekstra, A. Divoli, A.-L. Huang, and I. H. Witten, “Constructing a focused taxonomy from a document collection,” in *Proc 10th European Semantic Web Conference*, Montpellier, France, pp. 367–381, Springer, 2013.
- [29] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” 1967.
- [30] J. C. Platt, “Advances in kernel methods,” ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208, Cambridge, MA, USA: MIT Press, 1999.
- [31] R. T. J. Friedman, T. Hastie, “Additive logistic regression: A statistical view of boosting,” *The Annals of Statistics*, no. 2, pp. 337–407, 2000.
- [32] R. R. Bouckaert, “Bayesian network classifiers in weka for version 3-5-7,” 2008.
- [33] H. Thode Jr., *Testing for Normality*. Marcel Dekker, 2002.
- [34] H. Levene, “Robust tests for equality of variances,” in *Contributions to Probability and Statistics* (I. Olkin, ed.), Stanford University Press, 1960.
- [35] SAS Institute Inc., *SAS/STAT® 9.1 User’s Guide*. SAS Institute Inc., Cary, NC, USA, 2004. http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_91/stat_ug_7313.pdf.

BIBLIOGRAFÍA

- [36] R. M. Sakia, "The box-cox transformation technique: A review," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 41, no. 2, pp. pp. 169–178, 1992.
- [37] M. I. González, "Reunión de trabajo en la Unidad de Servicios en Estadística de la Universidad de Costa Rica." mariaisabel.gonzalezlutz@ucr.ac.cr, 2014. Comunicación personal.