



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ciencias Matemáticas

Escuela Profesional de Computación Científica

**Ajuste del modelo de funciones racionales para la
ortorectificación de imágenes satelitales**

TESIS

Para optar el Título Profesional de Licenciado en Computación
Científica

AUTOR

Clinton Teobaldo SAMANIEGO LOPEZ

ASESOR

Mg. Luis Javier VÁSQUEZ SERPA

Lima, Perú

2023



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Samaniego, C. (2023). *Ajuste del modelo de funciones racionales para la ortorectificación de imágenes satelitales*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Escuela Profesional de Computación Científica]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Clinton Teobaldo Samaniego Lopez
Tipo de documento de identidad	DNI
Número de documento de identidad	48034491
URL de ORCID	No aplica
Datos de asesor	
Nombres y apellidos	Luis Javier Vásquez Serpa
Tipo de documento de identidad	DNI
Número de documento de identidad	43389380
URL de ORCID	https://orcid.org/0000-0002-5414-6764
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Frank Duberlee Alvarez Huertas
Tipo de documento	DNI
Número de documento de identidad	45173557
Miembro del jurado 1	
Nombres y apellidos	Henry Zorrilla Masías
Tipo de documento	DNI
Número de documento de identidad	41190698
Miembro del jurado 2	
Nombres y apellidos	Luis Javier Vásquez Serpa
Tipo de documento	DNI
Número de documento de identidad	43389380
Datos de investigación	
Línea de investigación	A.3.4.6. Visualización gráfica y procedimiento de imágenes
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento

Ubicación geográfica de la investigación	Edificio: Facultad de Ciencias Matemáticas, Universidad Nacional Mayor de San Marcos. País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Av. Venezuela s/n cuadra 34. Lima 1-Perú Ciudad Universitaria. UNMSM. Latitud: -12.060186431587528 Longitud: -77.08213204783175
Año o rango de años en que se realizó la investigación	Mayo 2023 - octubre 2023
URL de disciplinas OCDE	Ciencias de la computación https://purl.org/pe-repo/ocde/ford#1.02.01 Ciencias de la información https://purl.org/pe-repo/ocde/ford#1.02.02



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú. Decana de América
FACULTAD DE CIENCIAS MATEMÁTICAS
ESCUELA PROFESIONAL DE COMPUTACIÓN CIENTÍFICA

**ACTA DE SUSTENTACIÓN DE TESIS PARA LA OBTENCIÓN DEL TÍTULO
PROFESIONAL DE LICENCIADO(A) EN COMPUTACIÓN CIENTÍFICA
(PROGRAMA DE TITULACIÓN PROFESIONAL 2023)**

En la UNMSM – Ciudad Universitaria – Facultad de Ciencias Matemáticas, siendo las 10:15 horas del sábado 28 de octubre del 2023, se reunieron los docentes designados como Miembros del Jurado Evaluador (PROGRAMA DE TITULACIÓN PROFESIONAL 2023): Dr. Frank Duberlee Alvarez Huertas (PRESIDENTE), Mg. Henry Zorrilla Masias (MIEMBRO) y el Mg. Luis Javier Vásquez Serpa (MIEMBRO ASESOR), para la sustentación de la Tesis titulada: **“AJUSTE DEL MODELO DE FUNCIONES RACIONALES PARA LA ORTORECTIFICACIÓN DE IMÁGENES SATELITALES”**, presentado por el señor **Bachiller CLINTON TEOBALDO SAMANIEGO LOPEZ**, para optar el Título Profesional de Licenciado en Computación Científica.

Luego de la exposición de la Tesis, el Presidente invitó al expositor a dar respuesta a las preguntas formuladas.

Realizada la evaluación correspondiente por los Miembros del Jurado Evaluador, el expositor mereció la aprobación *.....sobresaliente.....*, con un calificativo promedio de *.....dieciocho (18).....*

A continuación, los Miembros del Jurado Evaluador dan manifiesto que el participante **Bachiller CLINTON TEOBALDO SAMANIEGO LOPEZ**, en vista de haber aprobado la sustentación de su Tesis, será propuesto para que se le otorgue el Título Profesional de Licenciado en Computación Científica.

Siendo las 11:00 horas se levantó la sesión firmando para constancia la presente Acta.

Dr. Frank Duberlee Alvarez Huertas
PRESIDENTE

Mg. Henry Zorrilla Masias
MIEMBRO

Mg. Luis Javier Vásquez Serpa
MIEMBRO ASESOR



CERTIFICADO DE SIMILITUD

Yo LUIS JAVIER VÁSQUEZ SERPA en mi condición de asesor acreditado con la

Resolución Decanal N° 001668-2023-D-FCM/UNMSM de la

tesis, cuyo título es: "AJUSTE DEL MODELO DE FUNCIONES RACIONALES PARA LA ORTORECTIFICACIÓN DE IMÁGENES SATELITALES"

presentado por el bachiller CLINTON TEOBALDO SAMANIEGO LOPEZ

para optar el Título Profesional de Licenciado en Computación Científica de la Facultad de Ciencias Matemáticas,

CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de

Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la

revisión, análisis y evaluación mediante el software de similitud textual, el documento

evaluado cuenta con el porcentaje de ..6% de similitud, nivel **PERMITIDO** para continuar

con los trámites correspondientes y para su **publicación en el repositorio**

institucional.

Se emite el presente certificado en cumplimiento de lo establecido en las normas

vigentes, como uno de los requisitos para la obtención del grado/ título/ especialidad

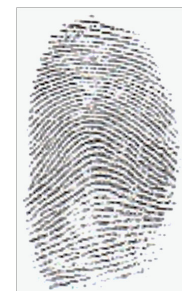
correspondiente.

Firma del Asesor _____

DNI: 43389380

Nombres y apellidos del asesor:

Luis Javier Vásquez Serpa



Resumen

El modelo RFM (Rational Functions Model) es muy usado en el área de la fotogrametría como herramienta de corrección geométrica (Ortorectificación) a imágenes satelitales. El modelo RPC (Rational polynomial coefficients) es un caso específico del modelo RFM, este modelo consiste en relacionar el espacio imagen y el espacio objeto mediante parámetros denominados coeficientes RPCs de ahí el nombre del modelo RPC.

La presente investigación toma como elemento particular las imágenes del satélite PeruSAT1. El centro de investigación CONIDA (Comisión Nacional de Investigación y Desarrollo Aeroespacial) cuenta con software licenciado de paga que le permite realizar la corrección geométrica de sus imágenes satelitales. Como antecedente se tuvo que al aplicar directamente el modelo RPCs no se obtuvo resultados a los del software. Es por ello que se hizo ajuste al modelo RPC para conseguir resultados similares a los del software.

Palabras clave: Modelo RFM, Modelo RPC, Ortorectificación, Python, rasterio

Abstract

The RFM model (Rational Functions Model) is widely used in the area of photogrammetry as a tool for geometric correction (Orthorectification) to satellite images. The RPC model (Rational polynomial coefficients) is a specific case of the RFM model. This model consists of relating the image space and the object space by means of parameters called RPC coefficients, hence the name of the RPC model.

The present investigation takes as a particular element the images of the PeruSAT1 satellite. The CONIDA research center (National Commission for Aerospace Research and Development) has paid licensed software that allows you to perform the geometric correction of your satellite images. As antecedent, it was had that when directly applying the RPCs model, no results were obtained from those of the software. That is why the RPC model was adjusted to achieve results similar to those of the software.

Keywords: RFM Model, RPC Model, Orthorectification, PeruSAT1, Python, rasterio

Índice general

Índice de figuras	VI
Índice de cuadros	VII
1 Introducción	1
2 Marco Teórico	3
2.1. Antecedentes	3
2.2. Preliminares Matemáticos	8
2.2.1. Transformaciones afines	8
2.2.2. Formula de Haversine	10
2.2.3. Modelo de Funciones Racionales	14
2.3. Preliminares Geomática	16
2.3.1. Modelo de Elevación Digital (DEM)	16
2.3.2. Modelo gravitacional de la tierra (EGM)	19
2.3.3. Evaluación de la precisión vertical de los modelos digitales de elevación	19
3 Metodología	21
3.1. Descripción de datos	21

3.1.1.	Imágenes Satelitales PeruSAT-1	21
3.1.2.	Datos del Satelitales PeruSAT-1	22
3.2.	Proceso de ortorectificación	24
3.2.1.	Calculo de matriz de coordenadas geográficas de la imagen satelital	24
3.2.2.	Calculo de matriz de filas y columnas	28
3.2.3.	Aplicación del Modelo RFM	29
3.2.4.	Establecer la dimensión del pixel de la imagen	29
4	Implementación computacional	32
4.1.	Diagrama de Flujo	32
4.1.1.	Desarrollo del programa	33
5	Conclusiones y Recomendaciones	46
5.1.	Conclusiones	46
5.2.	Recomendaciones	47
	Bibliografía	48

Índice de figuras

2.1. Traslación de un punto	8
2.2. Escala de un punto	9
2.3. Rotación de un punto	9
2.4. Escala de un punto	10
2.5. Coordenadas esféricas	11
2.6. Longitud de arco	12
2.7. Relación espacio objeto vs espacio imagen	14
2.8. Modelo Digital de Elevación como mapa de calor	17
2.9. Modelo Digital de Elevación como raster	18
2.10. Modelo Digitales de Elevación	20
3.1. Información de la imagen PeruSAT-1	25
3.2. Extracción de alturas	26
3.3. Imagen resultado sobre puesta a la imagen corregida por el software	31
4.1. Diagrama de flujo	33
4.2. Imagen vista desde Python	35
4.3. Imagen corregida vista desde Python	45

Índice de cuadros

3.1. Requerimientos del sistema	23
---	----

Introducción

Una imagen capturada por un satélite presenta dos tipos de distorsiones; distorsiones radiométricas y geométricas. La distorsión radiométrica se debe a factores externos como las condiciones atmosféricas o la fuente de iluminación, la corrección radiométrica generalmente se realiza en el procesamiento de la imagen. La distorsión geométrica se debe a anomalías en la plataforma como la orientación orbital del satélite, el traslado de datos de una superficie esférica a un plano, etc. es necesario que el usuario realice la corrección geométrica de la imagen.

El modelo RFM es muy popular en estos tiempos para la corrección de las distorsiones geométricas. Con esta investigación proponemos hacer ajuste al modelo RFM para lograr estas correcciones. Los datos disponibles son un Modelo de Elevación Digital (DEM) y la metadata de una imagen del satélite PerúSAT-1. La organización de esta investigación es como sigue:

El capítulo 2, contiene el marco teórico que cubren los antecedentes del tema principal de la investigación en la sección de antecedentes, las formulas y métodos matemáticos en la sección de Preliminares Matemáticos y los conceptos de geomática en la sección de Preliminares Geomática.

El capítulo 3, contiene la metodología que se empleara para el ajuste del modelo RFM, para ello se describira los datos que se tienen del satélite PeruSAT1 en la sección de Descripción de datos y el los pasos de ortorectificación en la sección del Proceso de ortorectificación.

El capítulo 4, contiene la implementación computacional para el ajuste al Modelo RFM, se presentara el Diagrama de Flujo así como las funciones programadas en el lenguaje de programación Python.

Finalmente en el capítulo 5, se mencionan las conclusiones del trabajo de investigación así como las recomendaciones para la posible mejora de este trabajo.

CAPÍTULO 2

Marco Teórico

2.1. Antecedentes

En esta sección se presenta una serie de estudios realizados (libros, tesis de grado, artículos y trabajos presentados) que tiene como base del presente tema de investigación.

En C. Vincent Tao y Yong Hu (2001) se hace resaltar las características del RFM, comenta que Los polinomios son una forma común de aproximación matemática, pero tienen una desventaja en cuanto a su comportamiento. Esta desventaja radica en su tendencia a la oscilación, es decir, que en ciertas situaciones pueden tener fluctuaciones o variaciones rápidas y abruptas en su gráfica. Esto puede llevar a que los límites de error en la aproximación polinomial sean más grandes que el error de aproximación promedio. Pero el RFM es alternativa con mejores propiedades de interpolación, tiende a

ser más suave en su comportamiento, lo que significa que presenta menos fluctuaciones abruptas en su gráfica. Además, puede distribuir el error de aproximación de manera más uniforme entre los puntos en los que se ajusta exactamente a los valores dados.

En Arie Croitoru, C. Vincent Tao y Yong Hu (2004) analiza el concepto de modelos de funciones racionales (RFM), en el contexto de relacionar las coordenadas de puntos de objetos (X, Y, Z) con las coordenadas de píxeles de imágenes (l, s) , o viceversa, en forma de funciones racionales que son cocientes de polinomios. Varios modelos de sensores, incluidos los modelos polinómicos 2D y 3D, el modelo de transformación proyectiva y el modelo de transformación lineal directa, se incluyen en el marco RFM. Para la transformación del suelo a imagen, las proporciones se definen en forma directa utilizando índices de muestra y línea normalizados (l_n, s_n) para píxeles en el espacio de la imagen, y valores de coordenadas normalizados (X_n, Y_n, Z_n) para puntos de objeto en espacio de tierra.

$$\begin{aligned} l_n &= \frac{P_1(X_n, Y_n, Z_n)}{P_2(X_n, Y_n, Z_n)} \\ s_n &= \frac{P_3(X_n, Y_n, Z_n)}{P_4(X_n, Y_n, Z_n)} \end{aligned} \quad (2.1)$$

Donde (l_n, s_n) son línea normalizada (fila) y el índice de muestra (columna) de píxeles en el espacio de la imagen; (X_n, Y_n, Z_n) son valores de coordenadas normalizados de puntos de objeto en el espacio terrestre. La normalización (es decir, compensación y escala) minimiza la introducción de errores durante el cálculo. Los coeficientes polinómicos también se denominan RPC, es decir, capacidad de posicionamiento rápido, coeficientes polinómicos racionales o datos de cámara polinómicos racionales. El término

modelo RPC a menudo se refiere a un caso específico del RFM que está en proceso de formación.

En Ian Dowman* and John T. Dolloff* (2000) enfoca al modelo RFM como un método tradicional para transformar espacio objetos a espacio imágenes utilizando ecuaciones colineales. Posteriormente, se realiza un examen crítico de la literatura pertinente con el propósito de responder a interrogantes fundamentales en relación a la exactitud y estabilidad de estos métodos. Se procede a analizar los pasos necesarios para calcular las funciones racionales y se evalúa la elección de parámetros, señalando posibles deficiencias en el proceso. Asimismo, se destaca la formación de un grupo de trabajo de la ISPRS bajo la dirección del autor principal, el cual se encuentra investigando múltiples facetas de las funciones racionales, y los resultados de dicha investigación se incorporan en el artículo.

En Jacek Grodecki y Gene Dial (2003) se centra en el modelo de cámara de Ikonos. Explica que los coeficientes usados en el RFM son una secuencia de 78 coeficientes determinados por un ajuste del modelo físico de cámara y si se tiene estos coeficientes, el cálculo de las coordenadas de píxeles de imágenes (l, s) es rápido, fácil y preciso. El modelo RPC relaciona el espacio objeto, con su representación de 3 coordenadas (ϕ, λ, h) , y las coordenadas del espacio imagen $(Line, Sample)$. Para mejorar la precisión numérica del modelo RPC, las coordenadas del espacio objeto y espacio imagen deben ser normalizados es decir deben estar en el rango de $< -1, +1 >$. Para ello se usa los valores de compensación y factores de escala $(LON_{SCALE}, LAT_{SCALE}, HEIGHT_{SCALE})$ de las coordenadas de longitud, latitud y altura correspondiente al objeto espacio.

$$\begin{aligned}
P &= \frac{\phi - LAT_{OFF}}{LAT_{SCALE}} \\
L &= \frac{\lambda - LON_{OFF}}{LON_{SCALE}} \\
H &= \frac{h - HEIGHT_{OFF}}{HEIGHT_{SCALE}}
\end{aligned} \tag{2.2}$$

En (Jacek Grodecki (2019)) al igual que los autores anteriores enfoca al modelo RFM como, la relación entre el espacio objeto (latitud, longitud, altura) y el espacio imagen (línea, columna). Este modelo funcional RPC adopta la forma de una relación compuesta por dos funciones cúbicas que vinculan los espacios mencionados. Para lograr esto, se emplean la normalización del los valores de cada variable. el espacio imagen del modelo RPC esta dado por:

$$\begin{aligned}
l &= \frac{Num_L(U, V, W)}{Den_L(U, V, W)} \\
s &= \frac{Num_S(U, V, W)}{Den_S(U, V, W)}
\end{aligned} \tag{2.3}$$

Donde U, V y W son las coordenadas normalizadas del espacio objeto . Sea ϕ, λ y h latitud, longitud y altura respectivamente entonces:

$$\begin{aligned}
U &= \frac{(\phi - O_\phi)}{SF_\phi} \\
V &= \frac{(\lambda - O_\lambda)}{SF_\lambda} \\
W &= \frac{(h - O_h)}{SF_h}
\end{aligned} \tag{2.4}$$

las variables l y s vienen a ser las coordenadas normalizadas del espacio imagen, Sea L y S fila y columna respectivamente, tenemos:

$$\begin{aligned} l &= \frac{(L - O_L)}{SF_L} \\ s &= \frac{(S - O_S)}{SF_S} \end{aligned} \quad (2.5)$$

Donde $O_\phi, O_\lambda, O_h, O_L$ y O_S valores promedios:

$$\begin{aligned} O_\phi &= \frac{1}{n} \sum \phi \\ O_\lambda &= \frac{1}{n} \sum \lambda \\ O_h &= \frac{1}{n} \sum h \\ O_L &= \frac{1}{n} \sum L \\ O_S &= \frac{1}{n} \sum S \end{aligned} \quad (2.6)$$

Y $SF_\phi, SF_\lambda, SF_h, SF_L$ y SF_S son los factores escalares:

$$\begin{aligned} SF_\phi &= \text{máx}(|\phi_{max} - O_\phi|, |\phi_{min} - O_\phi|) \\ SF_\lambda &= \text{máx}(|\lambda_{max} - O_\lambda|, |\lambda_{min} - O_\lambda|) \\ SF_h &= \text{máx}(|h_{max} - O_h|, |h_{min} - O_h|) \\ SF_L &= \text{máx}(|L_{max} - O_L|, |L_{min} - O_L|) \\ SF_S &= \text{máx}(|S_{max} - O_S|, |S_{min} - O_S|) \end{aligned} \quad (2.7)$$

2.2. Preliminares Matemáticos

2.2.1. Transformaciones afines

Para representar transformaciones afines con matrices, podemos usar coordenadas homogéneas. Esto significa representar un vector de 2D (x, y) como un vector de 3D $(x, y, 1)$

Consideremos un punto $X = (x, y)$, transformación afín de X son todas las transformaciones que puedan ser escrita.

$$x' = \begin{bmatrix} ax + by + c \\ dx + ey + f \end{bmatrix}$$

Donde a, b, c, d, e y f son escalares. Por ejemplo si tenemos que $a, e = 1$ y $b, d = 0$, entonces se tiene una traslación del punto X .

$$x' = \begin{bmatrix} x + c \\ y + f \end{bmatrix}$$

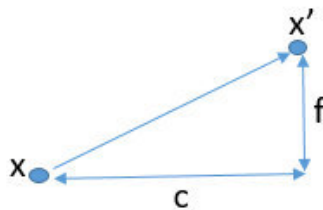


FIGURA 2.1: Traslación de un punto

Si consideramos que $b, d = 0$ y $c, f = 0$, entonces se tiene una escala del

punto X .

$$x' = \begin{bmatrix} ax \\ ey \end{bmatrix}$$

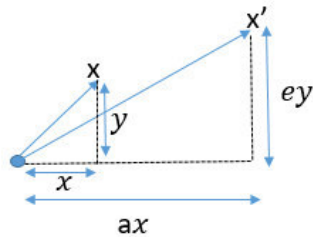


FIGURA 2.2: Escala de un punto

Si consideramos que $a, e = \cos(\theta)$, $b = -\sin(\theta)$, $d = \sin(\theta)$ y $c, f = 0$ entonces se tiene una rotación del punto X .

$$x' = \begin{bmatrix} x \cos(\theta) - y \sin(\theta) \\ x \sin(\theta) + y \cos(\theta) \end{bmatrix}$$

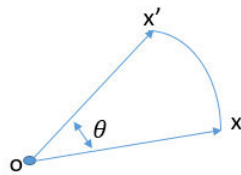


FIGURA 2.3: Rotación de un punto

Si consideramos que $a, e = 1$ y $c, f = 0$, entonces se tiene una transformación de corte del punto X .

$$x' = \begin{bmatrix} x + by \\ y + dx \end{bmatrix}$$

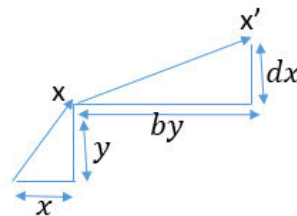


FIGURA 2.4: Escala de un punto

2.2.2. Formula de Haversine

La fórmula haversine, también denominada ecuación del gran círculo. Se usa comúnmente en la navegación puesto que permite calcular la distancia entre dos puntos de la superficie de la tierra, para el calculo de esta distancia se realiza dos etapas (ver Ariel Ivan Diaz (2012)):

1. Calcular la distancia entre dos puntos.
2. Convertir la distancia a una medida de la superficie de la tierra.

Para el punto 1 consideremos que tenemos 2 puntos $P1$ y $P2$ determinados por sus respectivos pares de longitud y latitud $P1(\theta_1, \phi_1)$ y $P2(\theta_2, \phi_2)$. En coordenadas cartesianas tenemos $P1 = P1(x_1, y_1, z_1)$ y $P2 = P2(x_2, y_2, z_2)$, donde x, y y z son determinadas por las coordenadas esféricas.

$$x = R \cos(\theta) \cos(\phi) \quad (2.8)$$

$$y = R \cos(\theta) \sin(\phi) \quad (2.9)$$

$$z = R \sin(\theta) \quad (2.10)$$

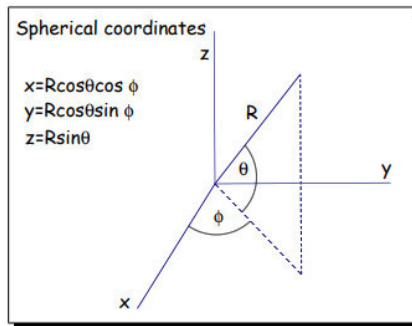


FIGURA 2.5: Coordenadas esféricas

2.2.2.1. Distancia Euclidiana

La distancia d entre los puntos P_1 y P_2 esta dada por el teorema de Pitagoras tridimensional.

$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \quad (2.11)$$

Usando las variables (z, y, z) expresadas en sus coordenadas esféricas

reemplazamos en la ecuación (2.11), tenemos:

$$\begin{aligned} \frac{d^2}{R^2} &= (\cos(\theta_1) \cos(\phi_1) - \cos(\theta_2) \cos(\phi_2))^2 \\ &+ (\cos(\theta_1) \sin(\phi_1) - \cos(\theta_2) \sin(\phi_2))^2 \\ &+ (\sin(\theta_1) - \sin(\theta_2))^2 \end{aligned}$$

$$\frac{d^2}{R^2} = 2 - 2 \cos(\theta_1) \cos(\theta_2) \cos(\phi_1 - \phi_2) - 2 \sin(\theta_1) \sin(\theta_2) \quad (2.12)$$

2.2.2.2. Distancia de un Arco

La distancia de la superficie de la tierra podemos calcular usando la formula de la longitud de Arco. Vemos la siguiente figura:

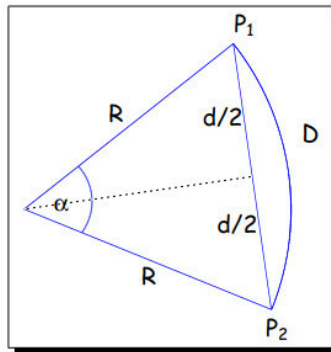


FIGURA 2.6: Longitud de arco

De la figura (2.6) obtenemos la siguiente ecuación:

$$\sin\left(\frac{\alpha}{2}\right) = \frac{d}{2R} \quad (2.13)$$

Resolviendo la ecuación (2.13) tenemos:

$$\begin{aligned}\sin(\alpha) &= 2 \sin\left(\frac{\alpha}{2}\right) \cos\left(\frac{\alpha}{2}\right) \\ &= \frac{d}{R} \sqrt{1 - \left(\frac{d}{2R}\right)^2} \\ &= \frac{d}{2R^2} \sqrt{4R^2 - d^2}\end{aligned}$$

Sin embargo sabemos la distancia D esta dado por:

$$D = R\alpha = R \sin^{-1} \left(\frac{d}{2R^2} \sqrt{4R^2 - d^2} \right) \quad (2.14)$$

La distancia se suele representar con la llamada función de *haversine*, el cual esta definido por:

$$\text{haversine}(A) = \sin^2\left(\frac{A}{2}\right) = \frac{1 - \cos(A)}{2} \quad \cos(A) = 1 - 2\text{haversine}(A)$$

Regresando a la ecuación(2.12) tenemos:

$$\begin{aligned}\frac{d^2}{R^2} &= 2 - 2 \cos(\theta_1) \cos(\theta_2) \cos(\phi_1 - \phi_2) - 2 \sin(\theta_1) \sin(\theta_2) \\ &= 2 - 2 \cos(\theta_1) \cos(\theta_2) [1 - 2\text{haversine}(\phi_1 - \phi_2)] - 2 \sin(\theta_1) \sin(\theta_2) \\ &= 4\text{haversine}(\theta_1 - \theta_2) + 4 \cos(\theta_1) \cos(\theta_2) \text{haversine}(\phi_1 - \phi_2)\end{aligned}$$

Tambien podemos expresar de la siguiente manera

$$\left(\frac{d}{2R}\right)^2 = \text{haversine}(\theta_1 - \theta_2) + \cos(\theta_1) \cos(\theta_2) \text{haversine}(\phi_1 - \phi_2)$$

para Finalmente obtener la siguiente expresión, conocida como la función de *haversine*.

$$\text{haversine}(\alpha) = \text{haversine}(\theta_1 - \theta_2) + \cos(\theta_1) \cos(\theta_2) \cdot \text{haversine}(\Delta\phi) \quad (2.15)$$

2.2.3. Modelo de Funciones Racionales

2.2.3.1. Método directo

El Modelo de Funciones Racionales es la división de dos polinomios cúbicos como la ecuación 2.16, en el cual se pretende encontrar nuevas coordenadas del espacio imagen (fila, columna) a partir del espacio objeto (Latitud, Longitud, Altitud).

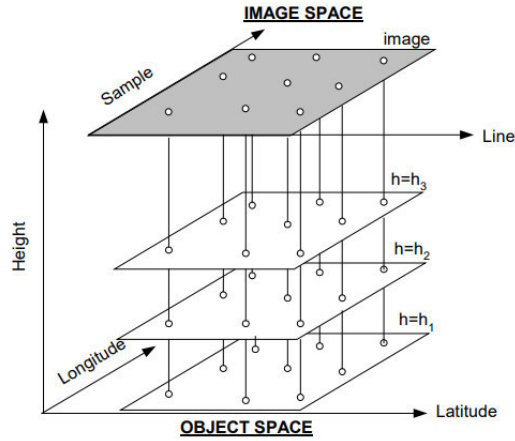


FIGURA 2.7: Relación espacio objeto vs espacio imagen

$$\begin{aligned}
 l_n &= \frac{p_1(X_n, Y_n, Z_n)}{p_2(X_n, Y_n, Z_n)} \\
 s_n &= \frac{p_3(X_n, Y_n, Z_n)}{p_4(X_n, Y_n, Z_n)}
 \end{aligned}
 \tag{2.16}$$

La ecuación 2.16 se puede escribir de la siguiente manera

$$\begin{aligned} l_n &= \frac{p_1(X_n, Y_n, Z_n)(a_1, a_2, \dots, a_{20})^T}{p_2(X_n, Y_n, Z_n)(b_1, b_2, \dots, b_{20})^T} \\ s_n &= \frac{p_3(X_n, Y_n, Z_n)(c_1, c_2, \dots, c_{20})^T}{p_4(X_n, Y_n, Z_n)(d_1, d_2, \dots, d_{20})^T} \end{aligned} \quad (2.17)$$

Donde:

- l_n y s_n son x (fila) y y (columna) normalizados de la figura 2.5 respectivamente.
- X_n, Y_n, Z_n son valores de coordenadas(latitud, longitud, altitud) normalizados del espacio objeto.
- $a_i, i = 1, 2, 3, \dots, 20$ vendrían a ser los coeficientes del polinomio p_1
- $b_i, i = 1, 2, 3, \dots, 20$ vendrían a ser los coeficientes del polinomio p_2 además $b_1 = 1$.
- $c_i, i = 1, 2, 3, \dots, 20$ vendrían a ser los coeficientes del polinomio p_3
- $d_i, i = 1, 2, 3, \dots, 20$ vendrían a ser los coeficientes del polinomio p_4 además $d_1 = 1$

2.2.3.2. Método inverso

El método inverso es la relación de coordenadas geográficas (tierra) a coordenadas imagen ?.

$$col_n = \frac{P_1(Y_n, X_n, Z_n)}{P_2(Y_n, X_n, Z_n)} \quad (2.18)$$

$$row_n = \frac{P_3(Y_n, X_n, Z_n)}{P_4(Y_n, X_n, Z_n)} \quad (2.19)$$

Donde:

- X_n es la longitud normalizada.
- Y_n es la latitud normalizada.
- Z_n es la altura normalizada
- row_n es la coordenada fila de la imagen normalizada
- col_n es la coordenada columna de la imagen normalizada

El polinomio P_1 tiene la siguiente forma:

$$\begin{aligned}
 P_1 = & A_1 + A_2X_n + A_3Y_n + A_4Z_n + A_5X_nY_n + \\
 & A_6X_nZ_n + A_7Y_nZ_n + A_8X_n^2 + A_9Y_n^2 + A_{10}Z_n^2 + \\
 & A_{11}Y_nX_nZ_n + A_{12}X_n^3 + A_{13}X_nY_n^2 + A_{14}X_nZ_n^2 + A_{15}X_n^2Y_n + \\
 & A_{16}Y_n^3 + A_{17}Y_nZ_n^2 + A_{18}X_n^2Z_n + A_{19}Y_n^2Z_n + A_{20}Z_n^3
 \end{aligned}$$

Donde A_i son los coeficientes del polinomio P_1 de manera similar se tiene B_i , C_i y D_i que son coeficientes de los polinomios P_2 , P_3 y P_4 respectivamente (para $i = 1, 2, 3, \dots, 20$).

2.3. Preliminares Geomática

2.3.1. Modelo de Elevación Digital (DEM)

Los Modelos Digitales de Elevación (DEM) son representaciones digitales que muestran la altitud de un terreno con respecto al nivel del mar. Estos DEM se crean mediante la conversión de datos de elevación recopilados de mapas topográficos o a través de técnicas avanzadas como imágenes estéreo, radar de apertura sintética o LiDAR. Lo que hacen los DEM es proporcionarnos una representación en tres dimensiones (3D) de un área

geográfica particular. Esto significa que podemos observar esa área desde diferentes perspectivas y a distintas alturas, lo que facilita el análisis y la visualización detallada del terreno.

Los modelos DEM se pueden superponer con fotografías aéreas, mapas topográficos, mapas geológicos y geomorfológicos para fines de visualización. Además, los datos morfométricos, como los gradientes de la pendiente, el aspecto de la pendiente, los cinturones altimétricos, la rugosidad de la superficie, el grano y los parámetros de la red hidrográfica se pueden extraer automáticamente de los DEM.

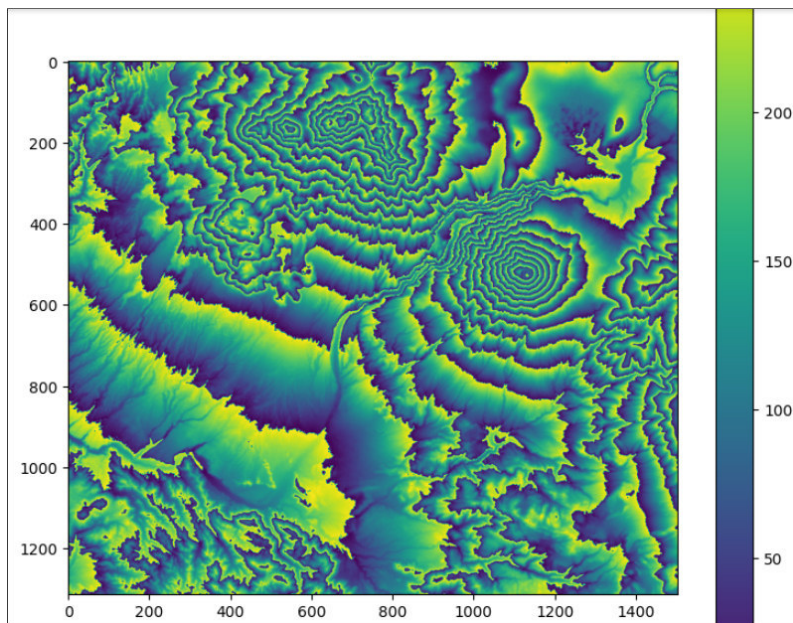


FIGURA 2.8: Modelo Digital de Elevación como mapa de calor

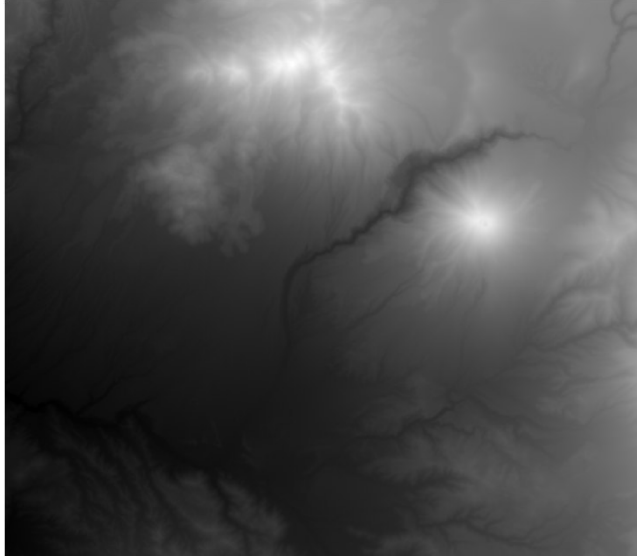


FIGURA 2.9: Modelo Digital de Elevación como raster

La disponibilidad de DEM detallados permite el análisis de la morfología y los procesos del paisaje en términos de morfometría topográfica o geomorfometría. Este enfoque es particularmente valioso en la geomorfología tectónica, cuyo objetivo es extraer información sobre las tasas y patrones de deformación activa de la topografía del paisaje. El estudio de los canales de lecho rocoso, en particular, juega un papel crucial en la comprensión de la relación entre las tasas de relieve, elevación y denudación. Los largos perfiles de los lechos rocosos de los ríos pueden proporcionar información valiosa sobre la distribución de la deformación reciente en la región subyacente.

2.3.2. Modelo gravitacional de la tierra (EGM)

Para hacer cálculos simples y aproximados, solemos imaginar la Tierra como una esfera. Sin embargo, la realidad es más compleja: la Tierra no es completamente esférica; está achatada en los polos, el hemisferio sur es un poco más grande que el norte y presenta irregularidades en su superficie debido a las montañas y valles. Por esta razón, la geodesia, una disciplina científica que se dedica al estudio de la forma y dimensiones de la Tierra, utiliza una representación aproximada llamada *elipsoide*.

El elipsoide es una figura geométrica matemática que se adapta de manera más precisa a la verdadera forma de la Tierra y puede ser definido mediante ecuaciones matemáticas altamente precisas. Esta característica facilita la ejecución de cálculos sumamente precisos en el campo de la geodesia, que es la disciplina científica encargada de estudiar la forma y dimensiones de la Tierra. En geodesia, se emplean varios modelos de elipsoides, y las diferencias entre ellos radican en los valores asignados a ciertos parámetros clave. En resumen, el elipsoide desempeña un papel crucial en la geodesia, permitiendo realizar cálculos extremadamente precisos que tienen en cuenta la verdadera forma de nuestro planeta.

2.3.3. Evaluación de la precisión vertical de los modelos digitales de elevación

El modelo EGM también conocida como la altura geoidal viene a ser la altura (N) entre la superficie elipsoidal y la superficie geoidal. Mientras que el SRTM que usamos como el DEM para el modelo RFM viene a ser la altura ortométrica (H), es por ello que para tener una altura mas precisa se

le tiene que agregar la altura geoidal como se muestra en la ecuación Luis A. Velo, Juan S. Flores (2018).

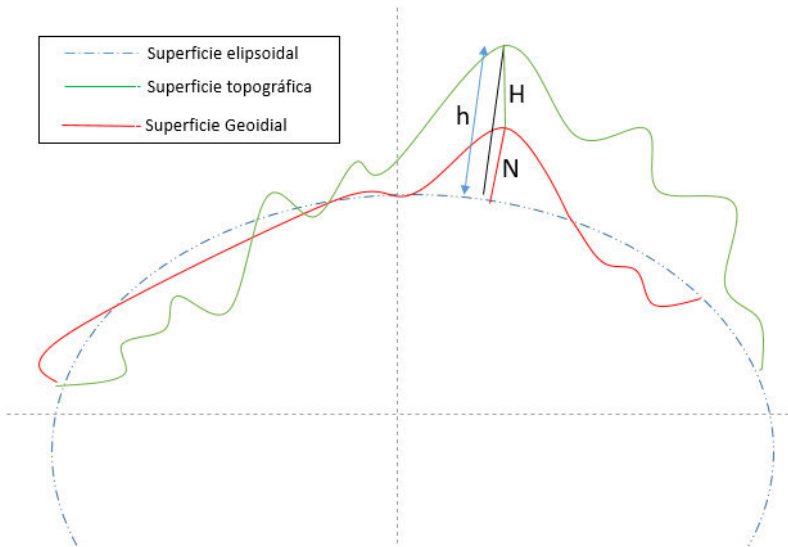


FIGURA 2.10: Modelo Digitales de Elevación

$$h = H + N \quad (2.20)$$

Donde:

- h es la altura elipsoidal
- H es la altura ortométrica
- N es la altura geoidal

CAPÍTULO 3

Metodología

3.1. Descripción de datos

3.1.1. Imágenes Satelitales PeruSAT-1

PeruSAT-1 es un satélite peruano especialmente diseñado para observar la Tierra y tomar imágenes de alta resolución en una variedad de rangos de colores. Este satélite fue desarrollado en colaboración con la empresa francesa Airbus Defence and Space a petición del gobierno peruano. Su finalidad es servir tanto para aplicaciones militares como civiles en el territorio de Perú.

El gobierno peruano encargó la construcción del satélite en abril de 2014, y el desarrollo del sistema de captura de información del satélite se llevó a cabo de manera exitosa en menos de dos años, cumpliendo con el plazo establecido como objetivo. Su misión principal es cumplir con los requisitos gubernamentales en diversas áreas, incluyendo el seguimiento de actividades

agroforestales, la planificación urbana, la detección de cultivos ilegales, la vigilancia de desastres naturales y la supervisión de infraestructuras de transporte.

3.1.1.1. Características del Satélite PerúSAT-1

En la tabla se ven mas detalles de las características del Satélite Jorge Roberto Pacheco Linares (2016).

3.1.2. Datos del Satelitaes PeruSAT-1

- **DIMAP.XML** se encuentra los vértices del imagen así como las el numero de filas columnas.

```
1 <Vertex>
2 <LON> -71.61813043</LON>
3 <LAT> -16.22138171</LAT>
4 <COL>0</COL>
5 <ROW>0</ROW>
6 </Vertex>
7 <Vertex>
8 <LON> -71.46383342</LON>
9 <LAT> -16.26024494</LAT>
10 <COL>5139</COL>
11 <ROW>0</ROW>
12 </Vertex>
13 <Vertex>
14 <LON> -71.5173381</LON>
15 <LAT> -16.47818336</LAT>
16 <COL>5139</COL>
17 <ROW>8159</ROW>
18 </Vertex>
19 <Vertex>
```

Parámetros	Valor
Resolución espacial	0.7m en pancromático(blanco y negro) y 2.8m en multispectral(colores)
Ancho de barrido	14.5 km
Bandas espectrales	5 bandas: 1 banda pancromático y 4 bandas en las regiones del rojo, verde, azul e infrarrojo cercano.
Área geográfica de cobertura	Todo el territorio peruano incluyendo el dominio marítimo, así como las áreas de interés fuera del territorio peruano
Resolución radiométrica	12 bits
Programación	12 horas o 3 horas en caso de emergencia
Seguridad de la transmisión	Encriptado
Memoria a bordo	1 Tera bits
Masa de la cámara	60 kg
Fila 10	Contenido 10
Tecnología	Carburo de silicio
Masa	400 Kg de lanzamiento
Vida útil	10 años

Cuadro 3.1: Requerimientos del sistema

```

20 <LON> -71.67273535</LON>
21 <LAT> -16.43910984</LAT>
22 <COL>0</COL>
23 <ROW>8159</ROW>
24 </Vertex>
25

```

- **RPC.XML** se encuentran los coeficientes racionales (RPCs) que se

emplean en el modelo RFM y también se encuentra los parámetros de normalización.

```
1 <LON_SCALE>0.1093861281</LON_SCALE>
2 <LON_OFF>-71.56438353</LON_OFF>
3 <LAT_SCALE>0.1331610664</LAT_SCALE>
4 <LAT_OFF>-16.35156792</LAT_OFF>
5 <HEIGHT_SCALE>2104.083743</HEIGHT_SCALE>
6 <HEIGHT_OFF>3954.046761</HEIGHT_OFF>
7 <COL_SCALE>2570</COL_SCALE>
8 <COL_OFF>2570</COL_OFF>
9 <ROW_SCALE>4080</ROW_SCALE>
10 <ROW_OFF>4080</ROW_OFF>
11
```

- **IMAGEN.TIF** es la imagen sin procesar a la cual se le aplicara el proceso de ortorectificación.

3.2. Proceso de ortorectificación

3.2.1. Calculo de matriz de coordenadas geográficas de la imagen satelital

3.2.1.1. Calculo de la matriz de Longitud y Latitud

Para obtener la matriz de longitudes y latitudes se usaron los vértices que nos proporciono el archivo DIMAP.XML. Se tiene 4 vértices por lo cual forma un cuadrilátero como se muestra en la figura 3.2. Una vez ubicado los 4 vértices, usamos la ecuación general de la recta, ver en la ecuación 3.1.

La estrategia consistió en extraer todos los puntos que pasan por los segmentos de color rojo. Una vez obtenido los puntos del segmento rojo, estos puntos se usaran para extraer los puntos de los segmentos de color

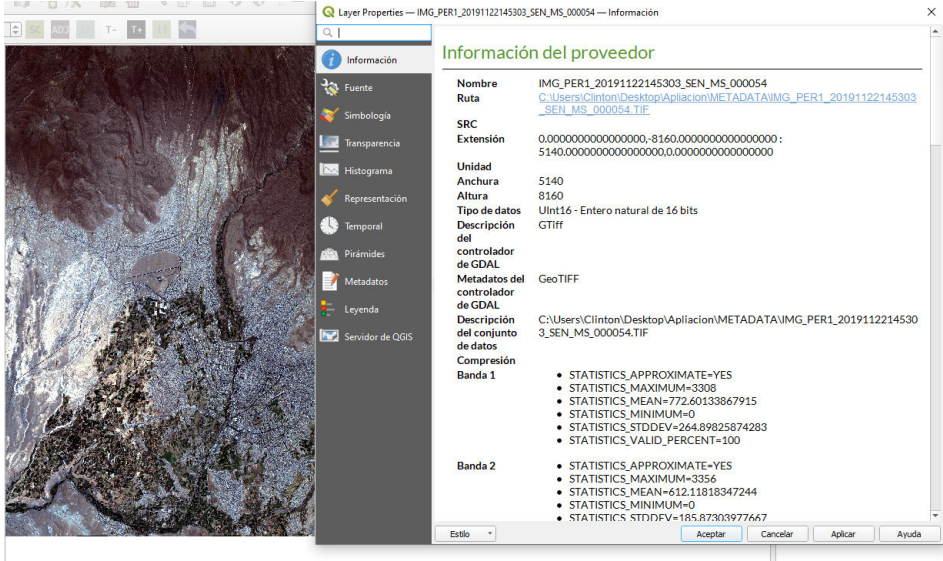


FIGURA 3.1: Información de la imagen PeruSAT-1

verde. El numero de segmentos de color verde deben ser igual a numero de filas de la imagen.

$$y - y_0 = m(x - x_0) \quad (3.1)$$

Donde x_0 , y_0 son los puntos iniciales, $m = (y_f - y_0)/(x_f - x_0)$ es la pendiente y y_f , x_f son los puntos finales. A continuación, se muestra las ecuaciones iterativas para el calculo de los puntos.

$$x_i = x_0 + \frac{(x_f - x_0) \cdot i}{N} \quad (3.2)$$

$$y_i = m(x_i - x_0) + y_0 \quad (3.3)$$

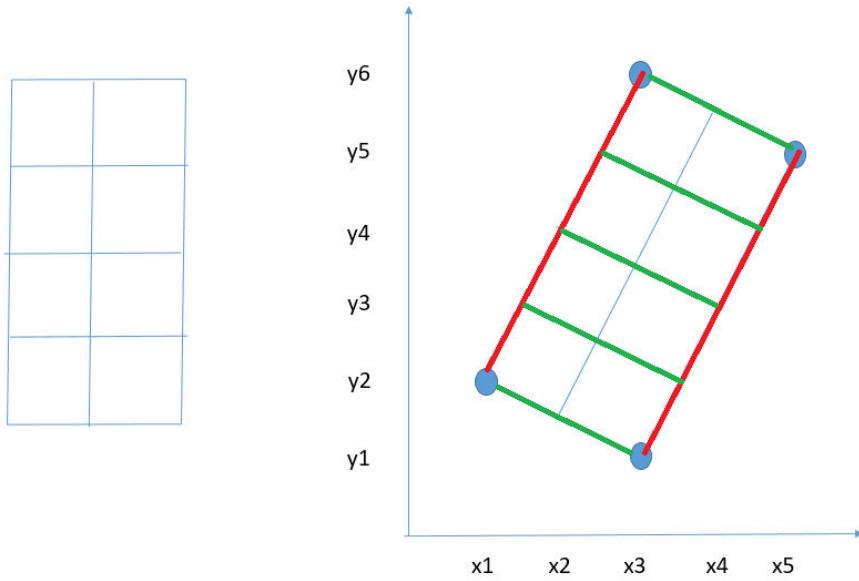


FIGURA 3.2: Extracción de alturas

Donde N es el número de columnas de cada píxel de la imagen e $i = 0, 1, 2, 3, \dots, N$. Con la ecuación (3.2) se ha calculado las longitudes que pasan por una de las rectas y con la ecuación (3.3) se ha calculado las latitudes que pasan por la misma recta entonces tenemos:

$$Lon_i = x_i \quad (3.4)$$

$$Lat_i = y_i \quad (3.5)$$

este proceso se debe repetir para todas las rectas que cubren el área de la imagen, podemos suponer que se tiene j rectas que cubren dicha área,

entonces tenemos al matriz de longitudes de la siguiente forma:

$$Lon = \begin{bmatrix} Lon_{0,0} & Lon_{0,1} & Lon_{0,2} & \cdots & Lon_{0,i} \\ Lon_{1,0} & Lon_{1,1} & Lon_{1,2} & \cdots & Lon_{1,i} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ Lon_{j,0} & Lon_{j,1} & Lon_{j,2} & \cdots & Lon_{j,i} \end{bmatrix}$$

Mientras que la matriz de latitudes tiene la siguiente forma:

$$lat = \begin{bmatrix} lat_{0,0} & lat_{0,1} & lat_{0,2} & \cdots & lat_{0,i} \\ lat_{1,0} & lat_{1,1} & lat_{1,2} & \cdots & lat_{1,i} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ lat_{j,0} & lat_{j,1} & lat_{j,2} & \cdots & lat_{j,i} \end{bmatrix}$$

Notamos que las matrices de longitudes y latitudes son de orden $j \times i$, donde j es el numero de filas y i es el numero de columnas. Ahora para obtener la matriz de altura, se deben usar las transformaciones afín, para convertir las coordenadas de longitud y latitud a coordenadas de pixel filas y columnas. Se le tiene que pasar la coordenadas de filas y columnas al DEM par obtener cada altura de cada pixel.

$$H = \begin{bmatrix} H_{0,0} & H_{0,1} & H_{0,2} & \cdots & H_{0,i} \\ H_{1,0} & H_{1,1} & H_{1,2} & \cdots & H_{1,i} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ H_{j,0} & H_{j,1} & H_{j,2} & \cdots & H_{j,i} \end{bmatrix}$$

De la misma forma se aplicaría para EGM obteniendo la altura geoidal

$$N = \begin{bmatrix} N_{0,0} & N_{0,1} & N_{0,2} & \cdots & N_{0,i} \\ N_{1,0} & N_{1,1} & N_{1,2} & \cdots & N_{1,i} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ N_{j,0} & N_{j,1} & N_{j,2} & \cdots & N_{j,i} \end{bmatrix}$$

Por lo enunciado en la ecuación (2.20) la altura que se aproxima mejor sería la suma de la altura ortométrica más la altura geoidal, entonces tenemos:

$$High = \begin{bmatrix} H_{0,0} + N_{0,0} & H_{0,1} + N_{0,1} & H_{0,2} + N_{0,2} & \cdots & H_{0,i} + N_{0,i} \\ H_{1,0} + N_{1,0} & H_{1,1} + N_{1,1} & H_{1,2} + N_{1,2} & \cdots & H_{1,i} + N_{1,i} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ H_{j,0} + N_{j,0} & H_{j,1} + N_{j,1} & H_{j,2} + N_{j,2} & \cdots & H_{j,i} + N_{j,i} \end{bmatrix}$$

3.2.2. Cálculo de matriz de filas y columnas

Una vez obtenida la matriz de longitudes, latitudes y alturas, procedemos a calcular la matriz de filas y de columnas. Las matrices de fila y columna se basa en la posición de cada pixel, posición que podemos expresar como $P(i, j)$, donde $i = 0, 1, 2, \dots, 8160$ y $j = 0, 1, 2, \dots, 5140$. Entonces las matrices tendrán la siguiente forma:

$$columna = \begin{bmatrix} 0 & 1 & 2 & \cdots & 5140 \\ 0 & 1 & 2 & \cdots & 5140 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & 2 & \cdots & 5140 \end{bmatrix}$$

Mientras que la matriz de filas tiene la siguiente forma:

$$filas = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ 2 & 2 & 2 & \cdots & 2 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 8160 & 8160 & 8160 & \cdots & 8160 \end{bmatrix}$$

Usando la ecuación 2.2 para normalizar los valores de las coordenadas y estén en el rango de $< -1, 1 >$ tenemos los resultados siguientes:

$$\begin{aligned} columna_n &= \frac{columna - columna_{off}}{columna_{scale}} \\ fila_n &= \frac{fila - fila_{off}}{fila_{scale}} \\ High_n &= \frac{High - High_{off}}{High_{scale}} \end{aligned} \quad (3.6)$$

3.2.3. Aplicación del Modelo RFM

Se procede a aplicar el meto inverso del modelo RFM.

$$Lon_n = \frac{P_1(fila_n, columna_n, High_n)}{P_2(fila_n, columna_n, High_n)} \quad (3.7)$$

$$Lat_n = \frac{P_3(fila_n, columna_n, High_n)}{P_4(fila_n, columna_n, High_n)} \quad (3.8)$$

Con esto hemos obtenido las nuevas coordenadas de longitud y latitud corregidas, procedemos a hacer una normalizacion inversa, entonces tenemos:

$$Lon = (Lon_n)(lon_{scale}) + lon_{off} \quad (3.9)$$

$$Lat = (Lat_n)(lat_{scale}) + lat_{off} \quad (3.10)$$

3.2.4. Establecer la dimensión del pixel de la imagen

Usamos la Formula de Haversine para establecer al dimensión que tendrá el pixel al reproyectar la imagen con sus nuevos valores. De la ecuación 2.15 tenemos:

$$haversine\left(\frac{D}{R}\right) = haversine(\phi_1 - \phi_2) + \cos(\phi_1)\cos(\phi_2).haversine(\Delta\lambda) \quad (3.11)$$

- $haversine(\theta) = \sin(\frac{\theta}{2})$.
- D = distancia entre dos puntos.
- R es radio de la esfera
- ϕ es latitud
- λ es longitud
- $\Delta\phi$ diferencia de latitud
- $\Delta\lambda$ diferencia de longitud

Luego se tiene las siguientes igualdades.

$$\begin{aligned}\Delta\phi &= \phi_1 - \phi_2 \\ \Delta\lambda &= \lambda_2 - \lambda_1 \\ A &= \sin(\frac{\Delta\phi}{2})^2 \cos(\phi_1) \cdot \sin(\frac{\Delta\lambda}{2}) \\ c &= 2 \cdot \text{Atan}(\frac{\sqrt{A}}{\sqrt{1-A}}) \\ d &= cR\end{aligned}$$

Finalmente sustituyendo las igualdades y convirtiendo los ángulos a radianes obtenemos lo siguiente valores de la dimensión del pixel en grados.

$$\begin{aligned}\Delta\phi &= \frac{180}{\pi R} dy \\ \Delta\lambda &= 2 \frac{180}{\pi R} \text{Arcsen}(\frac{\sin(\frac{dx}{2R})}{\cos(\phi)})\end{aligned}$$

Donde:

- dx es el ancho del pixel en metros
- dy es el alto del pixel en metros

Con esto se tendría todos los parámetros que se requieren para re proyectar le imagen ortorectificada. A continuación vemos los resultado obtenidos

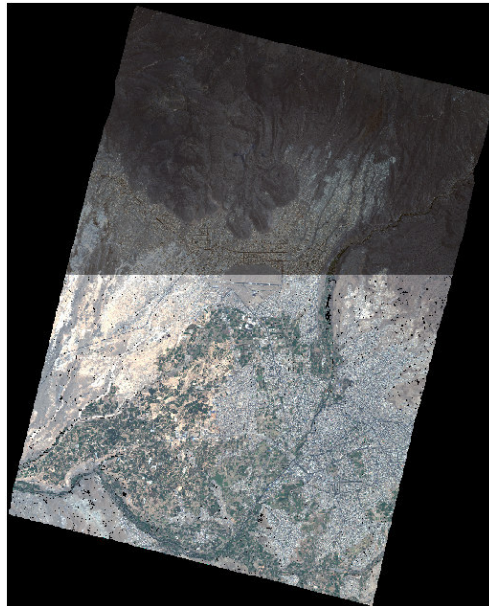


FIGURA 3.3: Imagen resultado sobre puesta a la imagen corregida por el software

Implementación computacional

La Implementación computacional de esta investigación se realizo con el lenguaje de programación Python, se eligio este lenguaje por su eficiencia y facilidad de aprendizaje, además de su capacidad para ejecutarse en múltiples plataformas. Python está disponible de forma gratuita, se integra de manera versátil con diversos sistemas y acelera el proceso de desarrollo. En particular se uso una de bastas librerías **resterio** y la mas conocida que es **numpy**.

4.1. Diagrama de Flujo

Primero se realizara un diagrama de flujo que se utlizo para el desarrollo del programa, el cual se puede visualizar en la figura 4.1, luego se pasara a codificar.

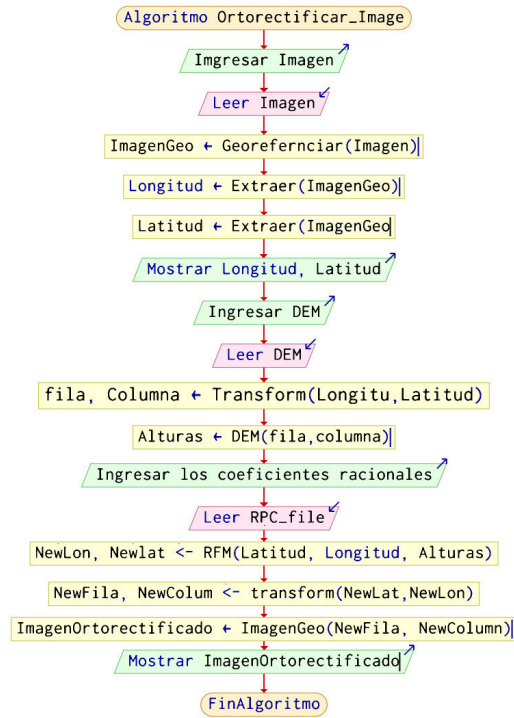


FIGURA 4.1: Diagrama de flujo

4.1.1. Desarrollo del programa

Antes de esta empezar con la explicación del desarrollo, pude considerar ver el código fuente ingresando a la siguiente ruta: https://colab.research.google.com/drive/1lDXxaim-0BHpfnQDprjAadWiufR_kpE4?usp=chrome_ntp#scrollTo=cpAsAdmnlQlu

4.1.1.1. Imports

Importar la librerías a usar:

```
1 # Librerías
2 import rasterio as rio # Manejo de rasters
3 import numpy as np # Manejo de arrays
4 import matplotlib.pyplot as plt # Manejo de Graficos
5 import xml.etree.ElementTree as ET # Manejo de archivos .
   xml
6 from rasterio.enums import Resampling
7 from rasterio.transform import Affine
8 from rasterio.plot import reshape_as_raster
9
```

4.1.1.2. Ingresar datos

Aquí se debe colocar la imagen en el formato .TIF, el archivo RPC en el formato .XML, el archivo DIMAP en el formato .XML, el modelo de elevación digital (DEM) en el formato .tif y el modelo gravitacional (EGM) en el formato .tif.

```
1 # Leer Imagen
2 imageR = rio.open('/content/drive/My Drive/PostGrado/
   AREQUIPA_DATA/IMG_PER1_20191122145303_SEN_MS_000054.TIF')
3
4 # Leer RPC_file
5 tree = ET.parse('/content/drive/My Drive/PostGrado/
   AREQUIPA_DATA/RPC_PER1_20191122145303_SEN_MS_000054.XML')
6 root = tree.getroot()
7
8 # Leer DIMAP_file
9 treeD = ET.parse('/content/drive/My Drive/PostGrado/
   AREQUIPA_DATA/DIM_PER1_20191122145303_SEN_MS_000054.XML')
```

```
10 rootD= treeD.getroot()
11
12 # Leer DEM
13 DEM=rio.open('/content/drive/My Drive/PostGrado/
    AREQUIPA_DATA/DEM_CUT.tif')
14
15 # Leer EGM
16 GEOD=rio.open('/content/drive/My Drive/PostGrado/
    AREQUIPA_DATA/geoide_96.tif')
```

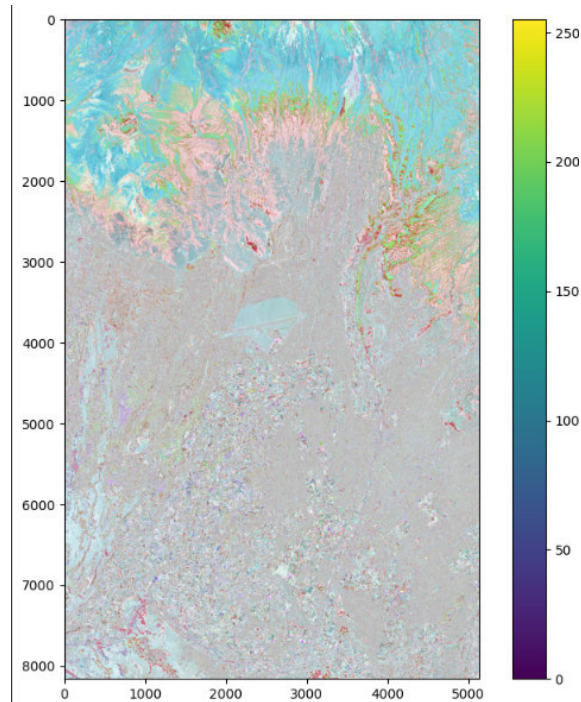


FIGURA 4.2: Imagen vista desde Python

4.1.1.3. Extracción de coordenadas

Primero estableceremos la dimensión de la imagen original para ello pasaremos la variable del archivo DIMA.xml y con ello obtenemos el número de filas y columnas de píxel que tiene la imagen.

```

1  def DIMENSION(root):
2      D = []
3      for element in root.iter('NROWS'):
4          NROWS = element.text
5          D.append(int(NROWS))
6      for element in root.iter('NCOLS'):
7          NCOLS = element.text
8          D.append(int(NCOLS))
9
10     Nfilas = D[0] # Numero de Filas
11     Ncolum = D[1] # Numero de Columnas
12
13     return Nfilas, Ncolum
14
15 #Obtener el numero de filas y Columnas
16 NROWS, NCOLS = DIMENSION(rootD)

```

la siguiente función permite extraer la matrices de longitudes y latitudes usando la ecuación de la recta.

```

1  def coor_extrac_rect(X1,Y1,X2,Y2,N): # Ecuacion que extrea
2      LAT y LON
3      m = (Y2-Y1)/(X2-X1)
4      P = []
5      L = []
6      for i in np.arange(0,N+1,1):
7          x1 = X1 + (X2-X1)*i/N
8          y1 = m*(x1-X1) + Y1
9          P.append(y1)

```

```

9     L.append(x1)
10    return P,L
11
12    def extrac_lon_lat(P1,L1,P2,L2,NROWS,NCOLS):
13        P = np.empty((NROWS,NCOLS)) # Matriz de Latitudes
14        L = np.empty((NROWS,NCOLS)) # Matriz de Longitudes
15        for i in np.arange(0,NROWS,1):
16            P[i][:],L[i][:] = coor_extrac_rect(L1[i],P1[i],L2[i],P2[i]
17            ],NCOLS-1)
18        return P,L
19
20        #Llamar a la funcion
21        P1,L1 = coor_extrac_rect(LON[0],LAT[0],LON[3],LAT[3],NROWS
22        -1)
23        P2,L2 = coor_extrac_rect(LON[1],LAT[1],LON[2],LAT[2],NROWS
24        -1)
25        P,L = extrac_lon_lat(P1,L1,P2,L2,NROWS,NCOLS) #Devuelve
26        la mtriz de latitud y longitud

```

la siguiente función permite extraer la matrices de alturas, para ello se le tiene que pasar la Matriz de latitud, longitud calculado en el paso anterior y el Modelo de Elevación Digital.

```

1     def extracc_alturas(DEM,P,L):
2
3         transformD = DEM.transform
4         px,py = ~transformD*(L,P)
5         px,py=px+0.5,py+0.5
6         px,py=px.astype(int), py.astype(int)
7         data_array=DEM.read(1)
8         H=data_array[py,px] # Matriz de Alturas
9
10        return H
11    H1 = extracc_alturas(DEM, P, L)
12    H2 = extracc_alturas(GEOD, P, L)

```

```
13 H = H1 + H2
```

la función Normalizacion nos permite normalizar las matrices calculadas en el en el paso anterior. Como resultado nos devuelve las matrices normalizadas.

```
1 def Normalizacion(root,P,L,H):
2     element_norm=[]
3     for element in root.iter('LON_SCALE'): # 0
4         long_scale = element.text
5         element_norm.append(float(long_scale))
6
7     for element in root.iter('LON_OFF'): # 1
8         long_off = element.text
9         element_norm.append(float(long_off))
10
11    for element in root.iter('LAT_SCALE'): # 2
12        lat_scale = element.text
13        element_norm.append(float(lat_scale))
14
15    for element in root.iter('LAT_OFF'): # 3
16        lat_off = element.text
17        element_norm.append(float(lat_off))
18
19    for element in root.iter('HEIGHT_SCALE'): # 4
20        height_scale = element.text
21        element_norm.append(float(height_scale))
22
23    for element in root.iter('HEIGHT_OFF'): # 5
24        height_off = element.text
25        element_norm.append(float(height_off))
26
27    P_n = (P - element_norm[3])/element_norm[2] # Matriz de
28    L_n = (L - element_norm[1])/element_norm[0] # Matriz de
        longitudes normalizadas
        latitudes normalizadas
```

```

29     H_n = (H - element_norm[5])/element_norm[4] # Matriz de
        altitud normalizadas
30
31     return P_n, L_n, H_n

```

De la misma forma se tiene que normalizar las filas y columnas el cual se usara en el modelo RFM.

```

1  def coorgeo2rowcol(P,root):
2      Alto, Ancho = P.shape
3
4      a = np.linspace(0,Ancho,Ancho)
5
6      # np.tile(matriz_entrada,repeticiones)
7      matriz_COL = (np.tile(a,(Alto,1))) # Matriz de Columnas
8
9      b = np.linspace(0,Alto,Alto)
10
11     # np.tile(matriz_entrada,repeticiones)
12     matriz_ROW = (np.tile(b,(Ancho,1))).T # matriz de filas
13
14     coef_norm=[]
15     for element in root.iter('COL_SCALE'): # 0
16         samp_scale = element.text
17         coef_norm.append(float(samp_scale))
18
19     for element in root.iter('COL_OFF'): # 1
20         samp_off = element.text
21         coef_norm.append(float(samp_off))
22
23     for element in root.iter('ROW_SCALE'): # 2
24         line_scale = element.text
25         coef_norm.append(float(line_scale))
26
27     for element in root.iter('ROW_OFF'): # 3

```

```
28     line_off = element.text
29     coef_norm.append(float(line_off))
30
31     COL_SCALE= coef_norm[0]
32     COL_OFF  = coef_norm[1]
33     ROW_SCALE= coef_norm[2]
34     ROW_OFF  = coef_norm[3]
35
36     col_n = (matriz_COL - COL_OFF)/COL_SCALE
37     row_n = (matriz_ROW - ROW_OFF)/ROW_SCALE
38
39     return col_n, row_n
```

4.1.1.4. Aplicación del modelo RFM

Con los valores ya calculados procedemos a aplicar el modelo de funciones racionales RFM para la corrección geométrica de la imagen.

```
1     def coeficientes(root):
2         P1 = [] # coeficientes del numerador de sample
3         P2 = [] # coeficientes del denominador de sample
4         P3 = [] # coeficientes del numerador de line
5         P4 = [] # coeficientes del denominador de line
6         for i in range(21):
7
8             xNUM = 'LON_NUM_COEFF_'+str(i)
9             xDEN = 'LON_DEN_COEFF_'+str(i)
10
11            yNUM = 'LAT_NUM_COEFF_'+str(i)
12            yDEN = 'LAT_DEN_COEFF_'+str(i)
13
14
15            for element in root.iter(xNUM):
16                value = element.text
```

```
17     P1.append(float(value))
18
19     for element in root.iter(xDEN):
20         value = element.text
21         P2.append(float(value))
22
23     for element in root.iter(yNUM):
24         value = element.text
25         P3.append(float(value))
26
27     for element in root.iter(yDEN):
28         value = element.text
29         P4.append(float(value))
30
31     return P1,P2,P3,P4
32 def RPCfuncion(C,row_n,col_n,H_n):
33
34     # Input    C: coeficientes racional
35     #          row_n: row_n: Matriz de filas normalizadas
36     #          col_n: Matriz de columnas normalizadas
37     #          H_n: Matriz de alturas normalizadas
38
39     xsize, ysize = col_n.shape
40     U = np.ones((xsize, ysize))
41     U = U.astype('uint16')
42     a = C[0]*1*U + C[1]*col_n + C[2]*row_n + C[3]*H_n
43     a = a + C[4]*col_n*row_n + C[5]*col_n*H_n + C[6]*row_n*H_n
44     a = a + C[7]*col_n*col_n + C[8]*row_n*row_n + C[9]*H_n*H_n
45     a = a + C[10]*row_n*col_n*H_n + C[11]*col_n*col_n*col_n
46     a = a + C[12]*col_n*row_n*row_n + C[13]*col_n*H_n*H_n
47     a = a + C[14]*col_n*col_n*row_n + C[15]*row_n*row_n*row_n
48     a = a + C[16]*row_n*H_n*H_n + C[17]*col_n*col_n*H_n
49     a = a + C[18]*row_n*row_n*H_n + C[19]*H_n*H_n*H_n
50
```

```

51     # Output Matrices para la evaluacion del Polinomo del
    modelo RFM
52     return a
53     numLon = RPCfuncion(C1,row_n,col_n,H_n)
54     denLon = RPCfuncion(C2,row_n,col_n,H_n)
55     numLat = RPCfuncion(C3,row_n,col_n,H_n)
56     denLat = RPCfuncion(C4,row_n,col_n,H_n)
57     NewLon = numLon/denLon
58     NewLat = numLat/denLat

```

Construimos el parámetro de transformación a la imagen con la siguiente función, para ello se le tiene que dar como valor las coordenadas nuevas.

```

1  def transf(P_new,L_new, NROWS, NCOLS): # Funcion que permite
    convertir
2
3  x0= np.min(L_new)# Longitud inicial
4  x1= np.max(L_new) # Longitud final
5  y0= np.min(P_new) # Latitud inicial
6  y1= np.max(P_new) # Latitud final
7  res1 = (x1 - x0)/NCOLS
8  res2 = (y0 - y1)/NROWS
9  t=Affine.translation(x0,y1)*Affine.scale(res1, res2)
10
11  return t
12
13  def Ortorectificaion(imageR,row,col):
14  band1 = imageR.read(1)
15  band2 = imageR.read(2)
16  band3 = imageR.read(3)
17  band4 = imageR.read(4)
18
19  B1=np.empty((np.max(row)+1,np.max(col)+1), dtype = 'uint16
    ')

```

```

20     B2=np.empty((np.max(row)+1,np.max(col)+1), dtype = 'uint16
    ')
21     B3=np.empty((np.max(row)+1,np.max(col)+1), dtype = 'uint16
    ')
22     B4=np.empty((np.max(row)+1,np.max(col)+1), dtype = 'uint16
    ')
23
24     B1[row,col] = band1[:]
25     B2[row,col] = band2[:]
26     B3[row,col] = band3[:]
27     B4[row,col] = band4[:]
28
29     newima = np.dstack((B1,B2,B3,B4))
30
31     return newima

```

Definimos el tamaño del pixel que tendrá como salida en la imagen, con la siguiente función

```

1     def PIXEL(root,VC):
2
3         GSD = []
4         for element in root.iter('GSD_ACROSS_TRACK'):
5             ACROSS = element.text
6             GSD.append(float(ACROSS))
7         for element in root.iter('GSD_ALONG_TRACK'):
8             ALONG = element.text
9             GSD.append(float(ALONG))
10        GSD_ACROSS = GSD[0]
11        GSD_ALONG = GSD[1]
12        R = 6378137
13        k = 180/math.pi
14        height_pixel = k*GSD_ALONG/R
15

```



```

16     widht_pixel = 2*k*math.asin(math.sin(GSD_ACROSS/(2*R))/
17     math.cos(VC/k))
18
19     return height_pixel, widht_pixel
20
21     def reproyet(newima, P_new, L_new, rootD, VC):
22
23         x0= np.min(L_new)# Longitud inicial
24         x1= np.max(L_new) # Longitud final
25         y0= np.min(P_new) # Latitud inicial
26         y1= np.max(P_new) # Latitud final
27
28         Alto,Ancho, bands = newima.shape
29         res2,res1 = PIXEL(rootD,VC)
30         Ancho = int((x1 - x0)/res1)
31         Alto = int((y1 - y0)/res2)
32
33         t=Affine.translation(x0,y1)*Affine.scale(res1, -res2)
34
35         return t, Alto, Ancho

```

Ahora configuramos la imagen de salida con la siguiente linea de comando. En el cual especificamos la ruta donde dejara la imagen corregida.

```

1  new_dataset = rio.open(
2      '/content/drive/My Drive/PostGrado/AREQUIPA_DATA/
3      Resultado2.tif',
4      'w',
5      bigtiff = 'IF_NEEDED',
6      driver = 'GTiff',
7      height = height,
8      width = width,
9      count = int(newimag.shape[2]), # numero de bandas
10     dtype = newimag.dtype,
11     crs = '+proj=latlong +datum=WGS84', # Sistema de
12     referencia de coordenadas

```

```
11     transform = Geotransf ,  
12     resampling = Resampling.cubic ,  
13     )
```

Finalmente aplicamos los cambios

```
1  newdata= reshape_as_raster(newimag)  
2  newdata.shape  
3  new_dataset.write(newdata)  
4  new_dataset.close()
```

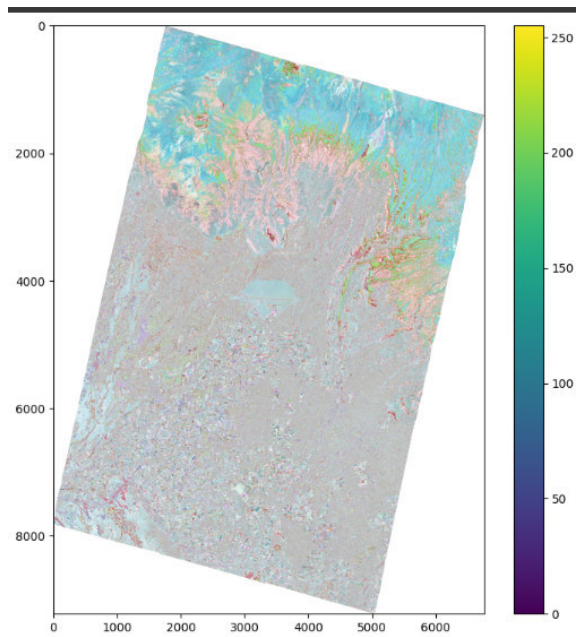


FIGURA 4.3: Imagen corregida vista desde Python

Conclusiones y Recomendaciones

5.1. Conclusiones

En el proceso de investigación, se pudo corroborar que aplicar el modelo RFM directamente a la imagen, no da un resultado óptimo. Pero con el ajuste que se aplicó a la corrección de las alturas y a la especificación del píxel de salida, se logró mejorar la precisión de la corrección geométrica. Pero una de las desventajas que se pudo apreciar en el desarrollo de esta investigación es que el proceso consume muchos recursos computacionales.

5.2. Recomendaciones

Esta investigación deja abierto muchos temas que se pueden aplicar para mejorar la precisión de la corrección geométrica de las imágenes satelitales, como por ejemplo optimizar los coeficientes que se emplean en el modelo RFM. Otra recomendación es que se implemente una interfaz de usuario para la facilidad de ejecución del programa.

Bibliografía

Arie Croitoru, C. Vincent Tao y Yong Hu. *UNDERSTANDING THE RATIONAL FUNCTION MODEL: METHODS AND APPLICATIONS*. GeoICT Lab, York University, 4700 Keele Street, Toronto M3J 1P3, 2004.

Ariel Ivan Diaz. *Calculadora Geoespacial*. Facultad de Informática de la Universidad de la Plata, 2012.

C. Vincent Tao y Yong Hu. *A Comprehensive Study of the Rational Function Model for Photogrammetric Processing*. American Society for Photogrammetry and Remote Sensing, 2001.

Ian Dowman* and John T. Dolloff*. *AN EVALUATION OF RATIONAL FUNCTIONS FOR PHOTOGRAMMETRIC RESTITUTION*. Department of Geomatic Engineering, University College London, UK, 2000.

Jacek Grodecki. *IKONOS STEREO FEATURE EXTRACTION - RPC APPROACH*. 12076 Grant Street, Thornton, CO 80241, 2019.

Jacek Grodecki y Gene Dial. *Block Adjustment of High-Resolution Satellite Images Described by Rational Polynomials*. American Society for Photogrammetry and Remote Sensing, 2003.

Jorge Roberto Pacheco Linares. *ACCESO A LAS IMÁGENES POR PARTE DE LAS INSTITUCIONES DEL ESTADO PERUANO*. Centro Nacional de Operaciones de Imágenes Satelitales, 2016.

Luis A. Velo, Juan S. Flores. *MODELO GEOIDAL CON RECEPTORES GNSS EN METODO CINEMATICO Y NIVELACION GEOMETRICA PARA LA ESTIMACION DE ALTURAS ORTOMETRICAS EN LA CIUDAD UNIVERSITARIA UNA-PUNO*. Universidad Nacional del Altiplano, 2018.