

# Sallfus, Library for Satellite Images Fusion on Homogeneous and Heterogeneous Computing architectures

Rubén Javier Medina-Daza, Nelson Enrique Vera-Parra and Andrés Ovidio Restrepo-Rodríguez

**Abstract**— Fusion of satellite images consists of improving the quality of a multispectral image by combining data from a high spatial resolution panchromatic image with a high spectral resolution multispectral image. To carry out this, different techniques are available, which perform various operations at the pixel level, which leads to generating a dependency between the computational requirement and the image size. Currently, there are some libraries that implement these fusion methods, however, none of them allow this fusion process to be carried out on heterogeneous architectures, which enable the integration of acceleration platforms that reduce execution time through massive parallelization. For this reason, this document presents a library called Sallfus, which allows executing and evaluating the quality and performance of image fusion methods such as the Brovey transform, Multiplicative method, Principal Component Analysis (PCA) and Wavelet À trous, on homogeneous and heterogeneous architectures. Likewise, an evaluation of the library is made from an analysis of execution times and image quality using mathematical-statistical indices such as the correlation coefficient (CC), BIAS coefficient and Root of the Root Mean Square Error (RMSE). The results of the library evaluation showed that the merging process with images of 8192 pixels, presents a speed-up of approximately 591x for Brovey, 309x for Multiplicative, 18x for PCA and 6x for À trous. Additionally, it was observed that the methods that presented the best performance both computationally and in the quality of the merged image were Brovey and Wavelet À trous. Availability and implementation: <https://github.com/Parall-UD/sallfus>.

**Index Terms**— Brovey transform, Heterogeneous Computing, Multiplicative transform, Principal Component Analysis, Satellite-image fusion, Wavelet À trous.

## I. INTRODUCCIÓN

**D**urante los últimos años, el procesamiento de imágenes ha tomado importancia en el campo científico,

Este trabajo fue financiado por el CIDC (Centro de Investigaciones y Desarrollo Científico) de la Universidad Distrital Francisco José de Caldas e hizo uso del CECAD (Centro de Cómputo de Alto Desempeño) de la misma Universidad.

R. J. Medina-Daza, Grupo de investigación GEFEM - Universidad Distrital Francisco José de Caldas.

N. E. Vera-Parra, Grupo de investigación GICOGE - Universidad Distrital Francisco José de Caldas.

A. O. Restrepo-Rodríguez, Grupo de investigación en Multimedia Interactiva y Animación Digital - Universidad Distrital Francisco José de Caldas.

de acuerdo con [1] el propósito del procesamiento de imágenes es maximizar el uso de la información de una imagen para un contexto en particular. De acuerdo con lo anterior, uno de los principales temas en este campo, es la fusión de imágenes, la cual según [2] hace referencia a la combinación de información relevante obtenida a partir de dos imágenes, esto con el fin de producir una imagen que contenga una calidad superior a las originales. Dentro de este campo, se pueden realizar fusiones de imágenes satelitales, donde de acuerdo con [2], se debe proporcionar una imagen pancromática para realizar una inyección de riqueza espacial en la información espectral asociada a la imagen multiespectral.

Para llevar a cabo el proceso de fusión de imágenes satelitales, se presentan una amplia gama de métodos, como los identificados en un estudio realizado por [3], donde se hace referencia a métodos como Intensity Hue Saturation (IHS), Transformada Wavelet, Análisis de Componentes Principales (PCA por su sigla en inglés Principal Component Analysis) y GramSchmidt. No obstante, existen otras técnicas como Brovey Transform y Multiplicative relacionadas en [4] y Wavelet À trous Transform presentada en [5]. Estos métodos exponen características en común, una de ellas es la implementación de operaciones matriciales píxel a píxel y de acuerdo con [6], este tipo de operaciones son altamente paralelizables.

Por otro lado, de acuerdo con [7], cada vez es mayor el número de estudios relacionados con procesamiento de alto rendimiento que utilizan las capacidades de la computación heterogénea para enfrentar retos de alta exigencia computacional [8]. Sin embargo, son pocos los estudios que han relacionado el uso de unidades de procesamiento gráfico (GPU) con la fusión de imágenes satelitales; uno de estos pocos estudios es [9], donde se expone la evaluación de rendimiento de métodos de fusión como lo son Brovey, Smoothing Filter-Based Intensity Modulation (SFIM), Hue Saturation Value (HSV), Hyperspherical Color Space (HCS) y High Pass Filter (HPF), mediante GPU haciendo uso de CUDA (Compute Unified Device Architecture), evidenciando una disminución significativa en tiempos de ejecución frente al procesamiento en CPU.

De acuerdo con lo expuesto anteriormente, este documento tiene como propósito presentar una librería realizada en el lenguaje de programación Python, enfocada en facilitar la implementación de los principales métodos de fusión de imágenes satelitales tanto en una arquitectura homogénea

utilizando exclusivamente la Unidad de Procesamiento Central (CPU por sus siglas en inglés Central processing unit) como en una arquitectura heterogénea haciendo uso de CPU/GPU. Además, esta librería nombrada como Sallfus, permite realizar la medición de calidad de la imagen fusionada mediante índices matemático-estadísticos.

Este artículo está organizado de la siguiente manera. La sección 2 presenta algunos antecedentes y trabajos relacionados. La sección 3, la descripción general de la librería, los distintos módulos que la componen y la metodología de evaluación de la Librería. La siguiente sección, presentará los resultados y análisis de la evaluación realizada a partir de tiempo de ejecución e índices matemático-estadísticos. En la sección 5, se presentan las conclusiones y los trabajos futuros.

## II. TRABAJOS RELACIONADOS

Una de las finalidades del procesamiento de imágenes es mejorar la calidad de una imagen para maximizar la detección de características y patrones relevantes, y hacer más evidentes ciertos detalles [10][11][12]. Dentro de este campo, se encuentran aplicaciones como la clasificación de imágenes [13], [14], compresión de imágenes [15], [16], diagnóstico médico [17], [18], detección de objetos u zonas en imágenes satelitales [19], [20], entre otras. Para poder llevar a cabo esta última aplicación, se hace necesario contar con una imagen que presente alta resolución espacial y espectral, es allí donde la fusión de imágenes satelitales cobra importancia. Entiéndase esta fusión, como el proceso de inyectar correctamente información espacial a una imagen multiespectral, la imagen fusionada presenta pérdida espectral, esta riqueza espectral se degrada en menor o mayor valor según el algoritmo implementado. Los datos espaciales inyectados, provienen de una imagen pancromática, las cuales se caracterizan por presentar una alta riqueza espacial [21].

En [22] se realiza una revisión de literatura relacionada con técnicas de fusión de imágenes satelitales; a partir de esto, se destacan técnicas como PCA, IHS, la transformada de Brovey y la transformada de Wavelet, entre otras. Los autores de esta revisión concluyeron que los métodos de fusión tienen un impacto positivo y significativo en la calidad de las imágenes resultantes. Adicionalmente, en [23] se realiza un estudio comparativo entre técnicas como Brovey, Multiplicación, PCA, Pansharpening, IHS, HPF y la transformada de Wavelet, obteniendo que Wavelet es la técnica que expone la mejor calidad de imagen a partir de índices matemático-estadísticos.

Una vez se ha obtenido la imagen resultado del proceso de fusión, es necesario realizar un análisis espectral y espacial de esta imagen. Para esto, de acuerdo con [24] se pueden utilizar índices matemático-estadísticos como el coeficiente de correlación, Root Mean Squared Error (RMSE), Entropy, BIAS, Universal Q index, entre otros.

Por otro lado, de acuerdo con [25], el campo de computación heterogénea se ha centrado en el desarrollo de entornos de procesamiento que involucren dispositivos many-core (por ejemplo, las GPUs) que aceleren tareas paralelizables de forma masiva como es el caso de muchas

tareas con imágenes. En este orden de ideas, los autores de [26], realizan una adaptación iterativa de PCA bajo una arquitectura many-core, exponiendo que, para conjuntos de datos muy grandes, PCA se vuelve bastante costosa en términos de carga computacional, lo que ha causado una necesidad de acelerar su computación. Así mismo, en [27], se presenta un método de procesamiento de imágenes acelerado por CUDA en una arquitectura multicore/many-core, realizando un filtro a partir de un mapa de bits; de acuerdo con los resultados experimentales, el algoritmo sobre una arquitectura many-core expuso un factor de aceleración de hasta 365 veces para una imagen de 8192 x 8192 píxeles.

Al relacionar el campo de la fusión de imágenes satelitales y la computación heterogénea, se puede exponer que existen librerías como [28]–[30], escritas en el lenguaje de programación Python. Sin embargo, a pesar de brindar la posibilidad de realizar el proceso de fusión mediante técnicas como Brovey, IHS, Wavelet, entre otras, no proporcionan al usuario la posibilidad de ejecutar estas técnicas haciendo uso de unidades gráficas de procesamiento, lo cual como se ha expuesto anteriormente, puede generar una disminución de tiempo.

## III. MATERIALES Y MÉTODOS

Esta sección presenta una descripción general de la librería, sus diferentes funcionalidades, las arquitecturas heterogéneas planteadas para la implementación de las distintas técnicas de fusión y, por último, el entorno computacional y método de evaluación de la librería.

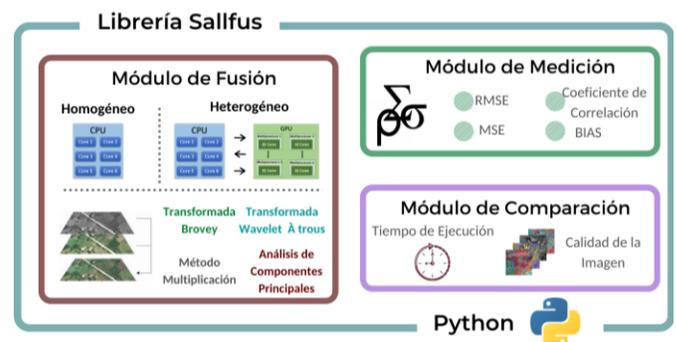


Fig. 1. Arquitectura general de Sallfus

### A. Descripción General

Sallfus tiene como propósito brindar herramientas y funcionalidades para implementar y evaluar la fusión de imágenes satelitales sobre arquitecturas homogéneas (CPU) y heterogéneas (CPU/GPU). Para esto, dispone de 3 módulos (ver Fig. 1): fusión (implementación homogénea y heterogénea de 4 métodos de fusión), medición (evaluación de calidad de riqueza espectral y espacial de la imagen fusionada frente a las originales utilizando 4 índices) y comparación (medición y comparación de tiempos de ejecución). Sallfus hace uso de librerías como Scipy [31] y del framework de paralelización CUDA [32], para realizar sus distintas rutinas de fusión, medición y comparación.

## B. Módulo de Fusión

La librería Sallfus, cuenta con un módulo cuya finalidad es realizar la fusión de una imagen multiespectral y una pancromática mediante técnicas como la transformada de Brovey, Multiplicación, À trous y Análisis por componentes principales. La imagen multiespectral, es una composición a color RGB (verdadero color), usando el mismo tamaño de píxel de la imagen pancromática (PAN). Adicionalmente, se presenta la posibilidad de hacer uso de cada uno de estos métodos tanto en una arquitectura homogénea (CPU) como en una arquitectura heterogénea (CPU/GPU).

Para la implementación en esta última arquitectura se hizo necesario plantear un modelo de procesamiento heterogéneo para cada uno de los métodos, que segmentara y distribuyera eficientemente las tareas entre los distintos dispositivos. Las tareas procesadas en GPU hacen uso de librerías eficientes como los son Scikit-cuda [33], CuPy [34] y Pycuda [35]; adicionalmente para PCA se hizo el aporte de dos kernels propios que paralelizan masivamente los procesos de obtención de la matriz de varianza-covarianza y la obtención de los componentes principales haciendo uso eficiente de la estructura jerárquica de la memoria de la GPU.



Fig. 2. Arquitectura heterogénea para la transformada de Brovey

### 1) Transformada de Brovey

Es un algoritmo basado en operaciones aritméticas, que se caracteriza por generar imágenes de buena calidad espacial, pero baja calidad espectral. La transformada de Brovey [36] es implementada para imágenes de  $N$  bandas según la expresión matemática:

$$ND_{FUS,bi} = \left( \frac{NB \times ND_{bi}}{ND_{bi} + ND_{bi} + \dots + BND_{bNB}} \right) * ND_{PAN} \quad (1)$$

Donde:

$NB$  es el número de bandas espectrales.

$ND_{FUS,bi}$  es el valor digital de la banda fusionada  $i$ .

$ND_{bi}$  es el valor digital de la banda multiespectral  $i$ .

$ND_{PAN}$  es el valor digital de la banda PAN.

La implementación realizada en Sallfus adiciona una etapa de ajuste de riqueza espectral la cual consiste, en restar el valor mínimo a cada elemento de una banda, multiplicarlos por un factor de 255 y este resultado dividirlo por la resta entre el valor máximo y mínimo. La Fig. 2 muestra el flujo de trabajo heterogéneo: los procesos que implican la operación de todos los píxeles se implementaron en la GPU en una sola línea de trabajo que reduce la transferencia de datos. La ecuación (1) se implementó mediante una normalización de bandas y una inyección de riqueza espacial utilizando Scikit-

cuda, el ajuste de riqueza espectral se llevó a cabo mediante los pasos 4 y 5 (ver Fig. 2) utilizando Scikit-cuda e implementando un kernel de tipo elementwise mediante Pycuda. Los procesos de los extremos correspondientes a la separación de bandas y a la compactación de la nueva imagen se implementaron en la CPU.

### 2) Método Multiplicativo

Este algoritmo permite incorporar el contenido de la imagen pancromática en la imagen multiespectral [37],[38], mediante la multiplicación mostrada en la ecuación (2).

$$R_{ijk} = M_{ijk} \times PAN \quad (2)$$

Donde

$R_{ijk}$  es la imagen fusionada.

$M_{ijk}$  es la imagen multiespectral banda  $k$ .

Para la implementación heterogénea de este algoritmo, la ecuación (2) se computó sobre la GPU mediante el módulo linealg de Scikit-cuda, adicionalmente, al igual que en Brovey, se introdujo un ajuste de riqueza espectral a partir de los máximos y mínimos; este ajuste se implementó mediante el módulo misc de Scikit-cuda (para hallar los máximos y mínimos) y mediante un kernel de tipo elementwise usando Pycuda (para aplicar el ajuste). La separación de bandas y la compactación de la nueva imagen se implementó en CPU (ver Fig. 3).

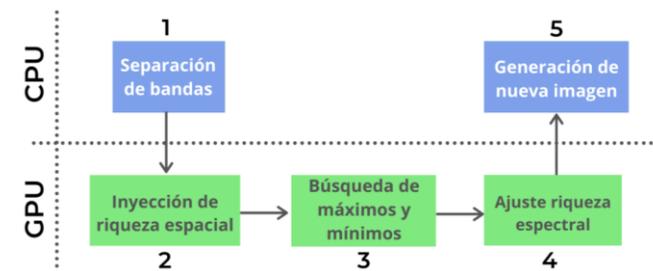


Fig. 3. Arquitectura heterogénea para el método Multiplicativo

### 3) Método PCA

La fusión PCA [39] se basa en la transformación de las bandas de la imagen multiespectral en una serie de componentes, para luego sustituir una de ellas por la imagen pancromática adaptada, buscando de esta manera añadir la información espacial a la espectral. El proceso consiste en obtener tantos CPs (Componentes Principales) como bandas tenga la imagen multiespectral. De este modo, el CP1 contiene información espacial, y los CPs restantes la información espectral. A continuación, se iguala el histograma de la imagen pancromática al del primer componente principal CP1, dado que CP1 retiene riqueza espacial y los demás CPs retiene riqueza espectral. La imagen pancromática modificada (una vez ajustado su histograma) sustituye el componente principal CP1. Aplicando a estos componentes la transformación inversa se obtienen la imagen fusionada.

Para la implementación heterogénea de este método se propuso el flujo de trabajo mostrado en la Fig 4. La obtención

de los CPs inversos se realiza netamente en GPU porque todos sus procesos son altamente paralelizables y operan sobre estructuras de datos del tamaño de las imágenes, contrario a los procesos que permiten la obtención de los CPs que no presentan esa regularidad, por tal motivo estos procesos se agruparon convenientemente en 3 bloques. 2 bloques (paso 2 y 4 de la Fig 4) de procesos altamente paralelizables que involucran todos los elementos de las imágenes de entrada: obtención de la matriz de varianza-covarianza y la obtención en sí de los CPs iniciales, y un bloque (paso 3 de la Fig. 4) de procesos que involucran arreglos de datos de muy pequeño tamaño que favorecen la ejecución secuencial eficiente y la baja transferencia de datos: el cálculo del coeficiente de diagonalización ortogonal, la obtención del polinomio característico, el cálculo de los vectores y valores propios y la obtención de la matriz ortogonalizada.

Para la obtención de la matriz de varianza-covarianza y de los CPs iniciales se diseñaron e implementaron dos kernels en CUDA aplicando una estrategia de “Tiles” jerárquicos de dos niveles que hace uso eficiente de la estructura de memoria de la GPU: carga la mayor cantidad de trabajo sobre los datos almacenados en los registros propios de cada hilo, carga poco trabajo (mediante operaciones atómicas) sobre los datos almacenados en memoria compartida y carga el mínimo trabajo posible sobre los datos almacenados en la memoria global. Los Tiles más pequeños están conformados por 3 elementos (un píxel en las 3 bandas) almacenados en un registro y operados por un mismo hilo; los tiles más grandes corresponden a un segmento de la imagen en sus 3 bandas almacenado en la memoria compartida y operados por los hilos de un mismo bloque.

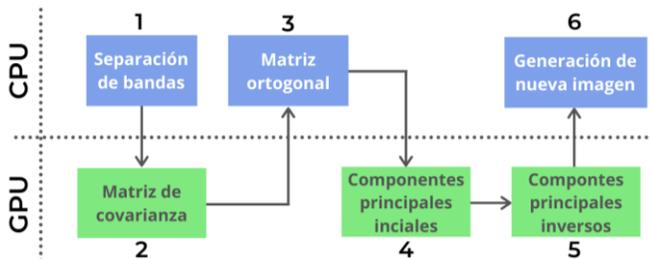


Fig. 4. Arquitectura heterogénea para el método PCA

4) Transformada Wavelet ‘À trous’

Las aproximaciones discretas de la transformación Wavelet algoritmo ‘À trous’ (con agujeros) [40], se implementan empleando un filtro bidimensional de paso bajo asociado a la función de escala, en este caso, una spline bi-cúbica. El algoritmo ‘À trous’, es no-ortogonal, lo que implica que un determinado plano wavelet para una escala, puede retener información de la escala vecina. Sintéticamente, los pasos para la implementación de la transformada Wavelet algoritmo de Á trous, son los siguientes: a) transformar la imagen RGB en componentes HSV (Value, Tono y Saturación); b) ajustar la PAN a la componente Value (Pan-V); c) ajustar los histogramas; d) aplicar el concepto de Transformada Wavelet algoritmo de Á trous al componente Pan-V; e) restar Pan-V con la imagen resultante para obtener el plano Wavelet w1

(donde se almacena la información espacial de Pan-V); f) aplicar la Transformada Wavelet algoritmo de Á trous a la imagen resultante y restarla con la anterior para obtener el segundo plano Wavelet w2; g) generar una nueva componente Value a partir de la suma de los planos Wavelet y la componente V, (N-VAL), el cual corresponde:

$$N - Val = V + w_1 + w_2 \tag{3}$$

Luego, h) generar una nueva composición HSV (N-HSV), concatenando la N-VAL junto con las componentes originales H y S. Por último, i) realizar la transformación HSV a RGB, usando la nueva composición N-HSV, para obtener la nueva imagen multispectral fusionada. Para la implementación heterogénea (ver Fig. 5), se agruparon los procesos de aplicación de filtros y de generación de planos wavelet para ejecutarlos en la GPU debido a su potencial para paralelización masiva y a su alta carga de trabajo por la complejidad de sus operaciones y por el tamaño de los datos a tratar.

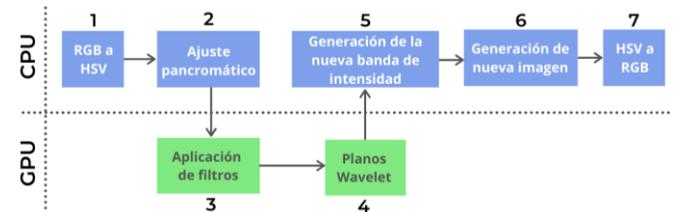


Fig. 5. Arquitectura heterogénea para la transformada Wavelet À trous

C. Módulo de Medición

Este módulo tiene como objetivo calcular la calidad de una imagen fusionada (Y), respecto a una imagen referencia (X). Esta imagen referencia puede ser la imagen multispectral inicial o la pancromática. Para llevar a cabo esto, este módulo dispone de 4 índices matemático-estadísticos, los cuales se presentan a continuación.

1) Coeficiente de Correlación

Este índice define el nivel de correlación entre las imágenes fusionadas y las imágenes originales. Este índice se considera confiable y, por lo tanto, ha sido ampliamente utilizado en la literatura. El valor del coeficiente de correlación se define en el intervalo [-1,1]; un valor de 0 indica que no hay correlación, mientras que un valor de 1 indica que las dos imágenes son iguales. Un valor de -1 indica que las imágenes son exactamente opuestas entre sí. La ecuación (4) presenta la descripción formal de este índice [24].

$$CC = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{4}$$

donde  $\sigma_{XY}$  es la covarianza,  $\sigma_X$ ,  $\sigma_Y$ , son las desviaciones estándar de las imágenes X y Y.

2) BIAS

Este índice se basa en la relación de valores promedio obtenidos de la imagen procesada y la imagen original. El

valor óptimo para este índice es 0, y los valores positivos o negativos más cercanos a cero representan una fuerte similitud entre las imágenes [24]. La ecuación (5) describe este índice.

$$BIAS = 1 - \frac{\bar{X}}{\bar{Y}} \quad (5)$$

$\bar{Y}$ ,  $\bar{X}$  valores medios de la imagen procesada  $Y$  y la original  $X$

### 3) MSE

El índice Error Cuadrático Medio (MSE por sus siglas en inglés Mean Square Error), describe la diferencia al cuadrado, que existe entre dos conjuntos de datos. A su vez, esta diferencia se divide entre el número total de elementos de los datos, dado que deben tener las mismas dimensiones, para el caso de imágenes, corresponde al número total de píxeles. Lo anterior, es presentado en (6).

$$MSE = \frac{\sum_{i=1}^n (X_i - Y_i)^2}{n} \quad (6)$$

### 4) RMSE

De acuerdo con [24], la Raíz del Error Cuadrático Medio (RMSE por sus siglas en inglés Root Mean Square Error) representa uno de los índices más populares y utilizados para este tipo de tareas. Representa la raíz de la diferencia al cuadrado, de dos conjuntos de datos, dividido por el número de elementos o píxeles. La ecuación (7), expone este índice.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (X_i - Y_i)^2}{n}} \quad (7)$$

## D. Módulo de Comparación

Este módulo tiene como objetivo brindar funcionalidades de comparación tanto a nivel de tiempos de ejecución como a nivel de calidad. Los tiempos de ejecución pueden corresponder a una o varias técnicas bajo un entorno homogéneo (CPU) o uno heterogéneo (CPU/GPU). Los niveles de calidad permiten comparar una imagen fusionada respecto a una imagen de referencia (normalmente las de entrada: pancromática o multispectral) haciendo uso de los índices matemático-estadísticos expuestos anteriormente.

## IV. RESULTADOS Y ANÁLISIS

En esta sección se presenta la metodología optada para la evaluación de esta librería. Adicionalmente, a partir del proceso de evaluación propuesto, se exponen los resultados y su respectivo análisis. Por lo tanto, se mostrarán los tiempos de ejecución de las técnicas de fusión implementadas en CPU y en CPU/GPU. Adicionalmente, se exhibirán los índices matemático-estadísticos como Coeficiente de Correlación, BIAS, MSE, RMSE, y, calculados a partir de las imágenes resultado de las técnicas de fusión, con los distintos pares de imágenes. Estos índices se presentarán a nivel espectral y espacial.

### A. Proceso de evaluación de la librería

A continuación, se presentan algunos de los aspectos de

evaluación a considerar, como los son, el entorno de computación, imágenes utilizadas y criterios.

### 1) Entorno computacional

TABLA I  
ENTORNO COMPUTACIONAL

SO	Procesador	Núcleos CPU	GPU	Memoria
CentOS7	Intel (R) Xeon (R) CPU E-52697 v3 @ 2.60GHZ	56	NVIDIA Tesla k80	128GB

La tabla 1, presenta las características del entorno de computación utilizado para llevar a cabo la evaluación de la librería Sallfus. Este entorno de computación fue acondicionado con la instalación de Sallfus, Python versión 3.6 y el Kit de desarrollo de software (SDK) de CUDA en su versión 8.0. Además de esto, se instalaron otros paquetes, como Scipy, Numpy, Pycuda y Cupy, necesarios para el correcto funcionamiento de la librería.



Fig. 6. Imagen de prueba con tamaño 1024x1024 píxeles. A) Multiespectral, B) Pancromática.

### 2) Imágenes de prueba

La evaluación de la librería se realizó con: a) una sub-imagen Landsat OLI TIRS y una imagen pancromática que corresponde a la ciudad de Bogotá - Colombia, y b) una sub-imagen Ikonos y una imagen pancromática que corresponde al aeropuerto El Dorado de la ciudad de Bogotá - Colombia. Se generaron un total de cuatro pares de imágenes, es decir cada par es compuesto por su respectiva imagen multispectral y pancromática. Estos pares de imágenes tenían distintos tamaños, los cuales eran 1024x1024, 2048x2048, 4096x4096 y 8192x8192 píxeles, respectivamente. Además, las imágenes de 1024 y 2048 píxeles son sub-escenas de una imagen Ikonos RGB verdadero color (Azul 1 visible: (0.45 - 0.52  $\mu\text{m}$ ) 4m, Verde: 2 visible (0.52 - 0.60  $\mu\text{m}$ ) 4m, Roja 3 visible: (0.63 - 0.69  $\mu\text{m}$ ) 4 m y la banda pancromática Ikonos (0.45 - 0.90  $\mu\text{m}$ ) 1m y las otras dos imágenes. Landsat 8 OLI TIRS, RGB verdadero color, Azul 2 Visible: (0.450 - 0.51  $\mu\text{m}$ ) 30 m, Band 3 Visible (0.53 - 0.59  $\mu\text{m}$ ) 30 m, Band 4 Red (0.64 - 0.67  $\mu\text{m}$ ) 30 m. y la Band 8 Pancromática (PAN) (0.50 - 0.68  $\mu\text{m}$ ) 15 m, es importante señalar que la longitud de onda de la imagen pancromática tanto de la imagen Landsat como Ikonos

corresponde parcial o en su totalidad a la longitud de onda de las bandas del visible. La Fig. 6(A) presenta la imagen multiespectral de 1024x1024 píxeles y la Fig. 6(B) la imagen pancromática. El conjunto de imágenes de prueba utilizadas, se pueden encontrar en el siguiente enlace: [https://github.com/Parall-UD/sallfus/blob/master/documentation/images\\_test.txt](https://github.com/Parall-UD/sallfus/blob/master/documentation/images_test.txt)

### 3) Métricas de evaluación

La evaluación se realizó en dos vías: a) medir y comparar los tiempos de ejecución de cada uno de los métodos para obtener el speed-up (CPU vs. CPU/GPU) de cada implementación, y su comportamiento en función del tamaño de la imagen; b) determinar la calidad espectral y espacial de las imágenes fusionadas, a partir de los índices matemático-estadísticos expuesto en el módulo de medidas.

TABLA II  
TIEMPO DE EJECUCIÓN

Método	Arq.	Tiempo por tamaño			
		1024x 1024px	2048x 2048px	4096x 4096px	8192x 8192px
Brovey	CPU	25.39s	76.76s	311.85s	1437.85s
	CPU/ GPU	1.95s	2.15s	2.26s	2.43s
Multipli cative	CPU	9.27s	36.66s	136.37s	534.57s
	CPU/ GPU	1.55s	1.57s	1.66s	1.73s
PCA	CPU	23.29s	86.32s	342.03s	1360.40s
	CPU/ GPU	3.05s	7.53s	24.36s	74.35s
À trous	CPU	1.94s	7.40s	30.86s	142.62s
	CPU/ GPU	1.08s	2.11s	5.84s	22.93s

### B. Resultados - Tiempo de ejecución

La tabla 2, presenta el tiempo de ejecución para cada una de las técnicas de fusión dispuestas en Sallfus. Se realiza una discriminación, por tamaño y tipo de arquitectura implementada. Con base en la tabla 2, se puede observar que para cada una de las técnicas implementadas tanto secuencial como paralelamente, a medida que incrementa el tamaño de la imagen incrementa su tiempo de ejecución, al igual que se expone en [8]. Sin embargo, para las técnicas de fusión que utilizan exclusivamente la CPU, se presentan incrementos de tiempo mucho más significativos que los presentados en las

TABLA III  
TASA DE CRECIMIENTO DEL TIEMPO DE EJECUCIÓN POR PÍXEL

Método	Tasa por arquitectura	
	CPU	CPU/GPU
Brovey	$2.13 \times 10^{-5}$ s/píxel	$7.26 \times 10^{-9}$ s/píxel
Multiplicative	$7.95 \times 10^{-6}$ s/píxel	$2.72 \times 10^{-9}$ s/píxel
PCA	$2.02 \times 10^{-5}$ s/píxel	$1.08 \times 10^{-6}$ s/píxel
À trous	$2.12 \times 10^{-6}$ s/píxel	$3.30 \times 10^{-7}$ s/píxel

implementaciones en CPU/GPU.

La tabla 3 presenta la tasa de crecimiento en segundos por píxel de cada una de las técnicas. Esto se realizó, mediante una linealización para obtener la pendiente que representa la tasa

TABLA IV  
SPEED -UP

Método	Speed- up por tamaño			
	1024 x 1024px	2048 x 2048px	4096 x 4096px	8192 x 8192px
Brovey	13,02x	35,70x	137,98x	591,71x
Multiplicative	5,98x	23,35x	82,15x	309,00x
PCA	7,63x	11,47x	14,04x	18,30x
À trous	1,79x	3,51x	5,29x	6,22x

de crecimiento del tiempo de ejecución en función de los píxeles. Lo anterior, dado que algunas de las técnicas presentan un comportamiento exponencial y otras aproximadamente lineal. Analizando esta tabla se evidencia que las tasas de crecimiento disminuyen sustancialmente al utilizar CPU/GPU (mucho más significativo para Brovey y Multiplicación), lo que indica que a mayor tamaño de las imágenes se sacará mayor beneficio de la plataforma heterogénea y se obtendrá mayor aceleración.

A partir de la tabla 4 y teniendo en cuenta el tamaño más alto de imagen, que en este caso corresponde a 8192x8192 píxeles, se puede observar que, la técnica que presenta la mayor aceleración en CPU/GPU respecto a CPU, es Brovey con un total de 591.71x. Después se ubica Multiplicación con un total de 309.00x, posteriormente, se encuentra PCA con 18.30x y, por último, está À trous, evidenciando solo 6.22x.

Adicionalmente se evaluó una implementación externa para Brovey denominada Pysharpening [28], a pesar de que esta implementación no involucra una etapa de ajuste de riqueza espectral, su tiempo de ejecución fue mayor que Sallfus. Esta diferencia fue más notable en la imagen de mayor tamaño: Pysharpening: 5.39s vs. Sallfus (heterogénea): 2.43s; por lo tanto, se evidenció un speed-up de 2.22x.

### C. Resultados - Calidad de la imagen

Al analizar los índices matemático-estadísticos a nivel espectral (ver tabla 5, columnas de riqueza espectral) se puede observar que los métodos que generan las imágenes

TABLA V  
ÍNDICES

Método	Índice	Riqueza espectral		Riqueza espacial	
		Ikonos 2048x 2048px	Landsat 8192x 8192px	Ikonos 2048x 2048px	Landsat 8192x 8192px
Brovey	CC	0.530	0.978	0.970	0.999
	RMSE	32.277	17.504	28.357	11.631
	BIAS	0.432	0.338	0.423	0.259
Multiplicative	CC	0.848	0.951	0.847	0.951
	RMSE	48.955	32.547	48.955	32.547
	BIAS	0.745	0.819	0.745	0.819
PCA	CC	0.558	0.979	0.938	0.998
	RMSE	31.725	17.240	28.543	20.100
	BIAS	0.421	0.506	0.420	0.439
À trous	CC	0.558	0.978	0.965	0.999
	RMSE	20.149	19.610	10.062	3.778
	BIAS	0.128	0.189	0.124	0.088

fusionadas con un nivel de similitud fuerte frente a la imagen multispectral de referencia son À trous y Brovey. Además, se presenta alta correlación espectral al utilizar imágenes tomadas del satélite Landsat. Es decir, se podría inferir que, a nivel espectral, las técnicas presentadas obtienen distintos valores asociados a la calidad de la imagen al utilizar imágenes tomadas de distintas fuentes. En cuanto a los índices a nivel espacial (ver tabla 5, columnas de riqueza espacial) se puede observar que las imágenes generadas por À trous exponen un sobresaliente nivel de similitud frente a la imagen pancromática. No obstante, Brovey también se aproxima a buen nivel de similitud. En este caso, en el contexto espacial, no se presentan comportamientos divergentes al utilizar distintas fuentes de imágenes.

## V. CONCLUSIONES

A partir de la comparación de tiempos de ejecución realizado, se demuestra que todos los métodos implementados presentan una disminución significativa en su tiempo de ejecución. Sin embargo, Brovey es el método que expone el mejor esquema de paralelización, dado que llega a ser aproximadamente 532 veces más rápido que en CPU. Adicionalmente, analizando la tasa de crecimiento del tiempo de ejecución por pixel, se evidencia que el método PCA presenta un comportamiento atípico frente a los otros métodos sobre una arquitectura heterogénea, esto podría significar que el costo de transferencia entre unidades de procesamiento es más alto que en las otras técnicas y que si de igual manera se presenta una mejora significativa en el tiempo de ejecución, las operaciones realizadas en PCA siguen representando un alto costo computacional en dispositivos many-core.

De acuerdo con los resultados y análisis expuestos anteriormente, se puede concluir que Salfus, brinda un conjunto de herramientas y funcionalidades, que le proporcionan a un usuario la facilidad de realizar y de evaluar (a nivel de tiempo y de calidad) un proceso de fusión de imágenes satelitales tanto en una plataforma homogénea como en una heterogénea, de acuerdo con la exigencia computacional de los métodos de fusión y de las imágenes utilizadas.

Este proyecto comprobó que la paralelización masiva usando plataformas heterogéneas es una solución real a la alta exigencia computacional de las técnicas de fusión de imágenes satelitales, estas imágenes fusionadas degradan en menor valor su riqueza espectral con una ganancia espacial significativa, la calidad de los métodos de fusión solo depende de los algoritmos utilizados y no de la arquitectura utilizada. Es por esto, por lo que se propone como trabajos futuros diseñar algoritmos de paralelización que permitan implementar modelos de procesamiento heterogéneo para técnicas de fusión no tenidas en cuenta en esta librería, tales como HPF [41], Gram-Schmidt [42] y algoritmo de Mallat [43].

## REFERENCES

- [1] N. Jindal and K. Singh, "Applicability of fractional transforms in image processing - review, technical challenges and future trends," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 10673–10700, Apr. 2019, doi: 10.1007/s11042-018-6594-0.
- [2] R. Swathika and T. S. Sharmila, "Image fusion for MODIS and Landsat images using top hat based moving technique with FIS," *Cluster Computing*, vol. 22, pp. 12939–12947, Sep. 2019, doi: 10.1007/s10586-018-1802-2.
- [3] A. Mookambiga and V. Gomathi, "Comprehensive review on fusion techniques for spatial information enhancement in hyperspectral imagery," *Multidimensional Systems and Signal Processing*, vol. 27, no. 4, pp. 863–889, 2016, doi: 10.1007/s11045-016-0415-2.
- [4] X. Zhu and W. Bao, "Investigation of Remote Sensing Image Fusion Strategy Applying PCA to Wavelet Packet Analysis Based on IHS Transform," *Journal of the Indian Society of Remote Sensing*, vol. 47, no. 3, pp. 413–425, Mar. 2019, doi: 10.1007/s12524-018-0930-8.
- [5] J. Medina, O. Ardila, and E. Upegui, "Implementación y evaluación de la transformada Wavelet 2-D A trous usando Phyton y Matlab para la fusión de Imágenes Satelitales WorldView-2." *Revista Ibérica de Sistemas e Tecnologías de Informação*; Lousada N.º E27, pp. 704-717, Mar 2020.
- [6] W. Sun, B. Chen, and D. W. Messinger, "Nearest-neighbor diffusion-based pan-sharpening algorithm for spectral images," *Optical Engineering*, vol. 53, no. 1, p. 013107, 2014, doi: 10.1117/1.oe.53.1.013107.
- [7] C. L. Zhang, Y. P. Xu, Z. J. Xu, J. He, J. Wang, and J. H. Adu, "A Fuzzy Neural Network Based Dynamic Data Allocation Model on Heterogeneous Multi-GPUs for Large-scale Computations," *International Journal of Automation and Computing*, vol. 15, no. 2, pp. 181–193, Apr. 2018, doi: 10.1007/s11633-018-1120-4.
- [8] F. Ye, X. Li, and X. Zhang, "FusionCNN: a remote sensing image fusion algorithm based on deep convolutional neural networks," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14683–14703, Jun. 2019, doi: 10.1007/s11042-018-6850-3.
- [9] I. S. Acikgoz, M. Teke, U. Kutbay, and F. Hardalac, "Performance evaluation of pansharpening methods on GPU for RASAT images," *RAST 2015 - Proceedings of 7th International Conference on Recent Advances in Space Technologies*, pp. 283–288, 2015, doi: 10.1109/RAST.2015.7208356.
- [10] P. Erbao and Z. Guotong, "Image Processing Technology Research of On-Line Thread Processing," *Energy Procedia*, vol. 17, pp. 1408–1415, 2012, doi: 10.1016/j.egypro.2012.02.260.
- [11] Alparone, L. Wald, L. Chanussoat, J. Thomas, C. Gamba, P. Bruce, L. (2007). Comparison of Pansharpening Algorithms: Outcome of the 2006 GRS-S Data Fusion Contest, *IEEE Transactions on Geoscience and Remote Sensing*. Vol. 45 no. 10. pp 3012-3021.
- [12] Medina, J. Pinilla, C. Joyanes, L. (2013). Two-Dimensional Fast Haar Wavelet Transform for Satellite-Image Fusion, *Journal of Applied Remote Sensing*. vol. 7, Issue 1, <https://doi.org/10.1117/1.JRS.7.073698>
- [13] A. O. R. Rodríguez, D. E. C. Mateus, P. A. G. García, C. E. M. Marín, and R. G. Crespo, "Hyperparameter optimization for image recognition over an AR-Sandbox based on convolutional neural networks applying a previous phase of segmentation by color-space," *Symmetry*, vol. 10, no. 12, Dec. 2018, doi: 10.3390/sym10120743.
- [14] A. O. Restrepo Rodríguez, D. E. Casas Mateus, P. A. Gaona Garcia, A. Gomez Acosta, and C. E. Montenegro Marín, *Segmentation Methods for Image Classification Using a Convolutional Neural Network on AR-Sandbox*, vol. 559. 2019.
- [15] Y. qiong Cai, H. xia Zou, and F. Yuan, "Adaptive compression method for underwater images based on perceived quality estimation," *Frontiers of Information Technology and Electronic Engineering*, vol. 20, no. 5, pp. 716–730, May 2019, doi: 10.1631/FITEE.1700737.
- [16] K. M. Hosny, A. M. Khalid, and E. R. Mohamed, "Efficient compression of volumetric medical images using Legendre moments and differential evolution," *Soft Computing*, vol. 24, no. 1, pp. 409–427, Jan. 2020, doi: 10.1007/s00500-019-03922-7.
- [17] A. Jenitta and R. Samson Ravindran, "Image Retrieval Based on Local Mesh Vector Co-occurrence Pattern for Medical Diagnosis from MRI Brain Images," *Journal of Medical Systems*, vol. 41, no. 10, Oct. 2017, doi: 10.1007/s10916-017-0799-z.
- [18] A. Yasar, I. Saritas, and H. Korkmaz, "Computer-Aided Diagnosis System for Detection of Stomach Cancer with Image Processing Techniques," *Journal of Medical Systems*, vol. 43, no. 4, Apr. 2019, doi: 10.1007/s10916-019-1203-y.
- [19] Y. Ya, H. Pan, Z. Jing, X. Ren, and L. Qiao, "Fusion object detection of satellite imagery with arbitrary-oriented region convolutional neural

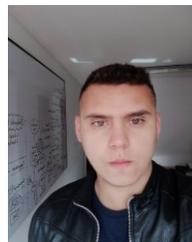
- network,” *Aerospace Systems*, vol. 2, no. 2, pp. 163–174, Dec. 2019, doi: 10.1007/s42401-019-00033-x.
- [20] F. Zhu, W. Shen, J. Diao, M. Li, and G. Zheng, “Integrating cross-sensor high spatial resolution satellite images to detect subtle forest vegetation change in the Purple Mountains, a national scenic spot in Nanjing, China,” *Journal of Forestry Research*, 2019, doi: 10.1007/s11676-019-00978-x.
- [21] Lillo-Saavedra, M. Gonzalo, C. Arquero, A. and Martinez, E. (2005). Fusion of multispectral and panchromatic satellite sensor imagery based on tailored filtering in the Fourier domain. *International Journal of Remote Sensing*. Vol. 26, pp. 1263-1268.
- [22] V. Solanky and S. K. Katiyar, “Pixel-level image fusion techniques in remote sensing: a review,” *Spatial Information Research*, vol. 24, no. 4, pp. 475–483, Aug. 2016, doi: 10.1007/s41324-016-0046-6.
- [23] A. Gupta and V. Dey, “A Comparative Investigation of Image Fusion in the Context of Classification,” *Journal of the Indian Society of Remote Sensing*, vol. 40, no. 2, pp. 167–178, Jun. 2012, doi: 10.1007/s12524-011-0138-7.
- [24] A. D. Vaiopoulos, “Developing Matlab scripts for image analysis and quality assessment,” in *Earth Resources and Environmental Remote Sensing/GIS Applications II*, 2011, doi: 10.1117/12.897806.
- [25] N. Vera, C. Rojas, and J. Pérez, *OPENCL PRÁCTICO COMPUTACIÓN HETEROGÉNEA PARALELA*. Bogotá: Colección Doctorado en Ingeniería, Universidad Distrital Francisco José de Caldas, 2019.
- [26] R. Lazcano et al., “Adaptation of an Iterative PCA to a Manycore Architecture for Hyperspectral Image Processing,” *Journal of Signal Processing Systems*, vol. 91, no. 7, pp. 759–771, 2019, doi: 10.1007/s11265-018-1380-9.
- [27] A. Asaduzzaman, A. Martinez, and A. Sepehri, “A time-efficient image processing algorithm for multicore/manycore parallel computing,” in *Conference Proceedings - IEEE SOUTHEASTCON*, 2015, vol. 2015-June, no. June, doi: 10.1109/SECON.2015.7132924.
- [28] GitHub. 2016. *Mapbox/Rio-Pansharpen*. [online] Available at: <<https://github.com/mapbox/rio-pansharpen>> [Accessed 13 April 2020].
- [29] GitHub. 2019. *Alexeytrekin/Pansharpen*. [online] Available at: <<https://github.com/AlexeyTrekin/pansharpen>> [Accessed 13 April 2020].
- [30] GitHub. 2014. *Developmentseed/Landsat-Util*. [online] Available at: <<https://github.com/developmentseed/landsat-util>> [Accessed 13 April 2020].
- [31] P. Virtanen et al., “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, doi: 10.1038/s41592-019-0686-2.
- [32] J. Nickolls, I. Buck, M. Garland, and K. Skadron, “Scalable Parallel PROGRAMMING.”
- [33] Givon, Lev E., et al. *scikit-cuda 0.5. 1: a Python interface to GPU-powered libraries*. 2015.
- [34] Okuta, Ryosuke, et al. “Cupy: A numpy-compatible library for nvidia gpu calculations.” *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*. 2017.
- [35] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih, “PyCUDA and PyOpenCL: A scripting-based approach to GPU runtime code generation,” *Parallel Computing*, vol. 38, no. 3, pp. 157–174, 2012.
- [36] Hallada, Wayne A., and Scott Cox. “Image sharpening for mixed spatial and spectral resolution satellite systems.”. 1983.
- [37] C. Pohl and J. L. V. Genderen, “Review article Multisensor image fusion in remote sensing: Concepts, methods and applications,” *International Journal of Remote Sensing*, vol. 19, no. 5, pp. 823–854, 1998.
- [38] Javier Medina Daza Rubén, Alberto Lizarazo Salcedo Iván, A. Vivas, Calderón Carlos, and A. Munar, *Fusión de imágenes satelitales usando la transformada de wavelet*. Bogotá (Colombia): Universidad Distrital Francisco José de Caldas. Centro de investigaciones y Desarrollo Científico, 2004.
- [39] K. Shettigara, “A Linear Transformation Technique For Spatial Enhancement Of Multispectral Images Using A Higher Resolution Data Set,” *12th Canadian Symposium on Remote Sensing Geoscience and Remote Sensing Symposium*.
- [40] J.-L. Starck, F. D. Murtagh, and A. Bijaoui, “,” *Image Reconstruction and Restoration*, 1994.
- [41] Schowengerdt, R., “Reconstruction of Multispatial, Multispectral Image Data Using Spatial Frequency Content.” *Photogrammetric Engineering & Remote Sensing*, vol. 46, no. 10, pp. 1325-1334, October. 1980.
- [42] B. Aiazzi, S. Baronti, M. Selva and L. Alparone, "Enhanced Gram-Schmidt Spectral Sharpening Based on Multivariate Regression of MS and Pan Data," *IEEE International Symposium on Geoscience and Remote Sensing*, Denver, CO, 2006, pp. 3806-3809, doi: 10.1109/IGARSS.2006.975.
- [43] J. Nunez, X. Otazu, O. Fors, A. Prades, V. Pala and R. Arbiol, "Multiresolution-based image fusion with additive wavelet decomposition," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 3, pp. 1204-1211, May 1999, doi: 10.1109/36.763274.



**Rubén Javier Medina Daza** es Doctor en Informática, énfasis: Sistemas de Información Geográfica, Universidad Pontificia de Salamanca. Magíster en Teleinformática, Licenciado en Matemáticas de la Universidad Distrital Francisco José de Caldas. Profesor Titular de Ingeniería Catastral y Geodesia, de la Maestría en Ciencias de la Información y las Comunicaciones y del Doctorado de Ingeniería.



**Nelson Enrique Vera Parra** es Ingeniero Electrónico de la Universidad Surcolombiana, Magíster en Ciencias de la Información y las Comunicaciones y Doctor en Ingeniería de la Universidad Distrital Francisco José de Caldas. Profesor Titular de la misma Universidad. Investigador en HPC y Ciencia de datos.



**Andrés Ovidio Restrepo Rodríguez** es Ingeniero de sistemas y actual estudiante de la Maestría en Ciencias de la Información y las Comunicaciones en la Universidad Distrital Francisco José de Caldas. Sus campos de investigación incluyen Inteligencia Artificial, Learning Analytics, Entornos Inmersivos y Computación Heterogénea.