

# Filtrado de imágenes correlacionadas en telescopios de gran envergadura (Septiembre de 2020)

Huerfano Z. Benjamin, Pontificia universidad católica de Chile

**Resumen** – El filtrado de imágenes es el eslabón básico para desarrollar actividades sobre el procesamiento de imágenes, solucionando problemas de presentación de la información y ayudando a eliminar los errores generados por los métodos de adquisición de datos y/o ambientes abiertos no controlables en la captura de la imágenes, un área sobre la cual las imágenes juegan un rol importante en la astronomía, donde literalmente la visualización de los astros tiene cabida directa sobre el procesamiento de imágenes, basado en ello el actual paper presenta el análisis e implementación de filtrado de imágenes correlacionadas, en el caso de estudio son obtenidas de telescopios de gran envergadura, generadas por pares de sensores [1]. En el presente documento se estudian modelos y métodos para la eliminación de ruido de imágenes y en secuencias de imágenes, con el fin de brindar una base para el posterior procesamiento de las imágenes obtenidas de la sensorica de telescopios. Se Realizo la implementación de 5 métodos de filtrado para imágenes estáticas y 3 para secuencias de imágenes, con los cuales se brinda una comparación de los resultados.

**Índice de Términos** - Filtrado, denoising, Image processing, video

## I. INTRODUCCION

El análisis de imágenes espaciales adquiridas por telescopios, tienen dinámicas inherentes a los modelos de los lentes usados, tanto sus características intrínsecas (Física óptica, curvaturas, superficies...) como extrínseca (Ambientes, movimiento, vibraciones, métodos de adquisición...) [2], específicamente hablando los telescopios de gran envergadura acentúan problemas de dichas características intrínsecas y extrínsecas, debido a que pequeños cambios [1-3] afecta las suposiciones que a menor escala son depreciables, por ejemplo su superficie y los cual afecta la forma en que presenta las imágenes a los astrónomos. Por otra parte, las condiciones ambientales y ámbito en el que se toman la imagen afectan de diferentes maneras las imágenes observadas, por tanto, características indeseables aparecen sobre las adquisiciones.

La adquisición de imágenes y el procesamiento de la mismas a nivel computacional ha tenido gran acogida a nivel académico e investigativo en los últimos años como lo relaciono en [4]. Finalmente el desarrollo tecnológico actual presenta una constante en la evolución, en cuanto al desarrollo de software se refiere, como lo señala Gonzales[5], trayendo consigo numerosas ventajas, entre las cuales cabe destacar el desarrollo

de tecnologías más eficientes, rápidas y baratas [6]; como resultado de lo anterior es plausible abordar ciertas temáticas que tiempo atrás eran ignoradas por la exigencia de grandes recursos computacionales. Esto se debe a que las nuevas tecnologías han conseguido mejorar las velocidades de procesamiento de la información, abriendo así un amplio camino, como lo es el estudio en el desarrollo de algoritmos de PDI.

Con miras a esto, en el presente documento recopila métodos y sistemas de filtrado tanto de imágenes estáticas como secuencias de imágenes correlacionadas, con el fin de abordar la problemática inicial de denoising en las imágenes y prepararlas para un posterior trabajo adicional, buscando eliminar objetos y ruidos indeseables sobre las imágenes, en este caso se busca mitigar los efectos de ópticas adaptativas relacionadas con la atmosfera y el frente de onda. “Recientemente se ha diseñado un dispositivo capaz de medir esta interacción” (Guesalaga et al, 2017)[1] sobre el cual se permitió obtener un pequeño set de imágenes de la correlación de pares de sensores con un total de 480 imágenes divididas en set de 30 imágenes, las cuales representan 6000 muestras de los sensores para lograr un set de imágenes. [1]

## II. ANÁLISIS DE REQUISITOS

En este apartado se encuentran consignado los elementos usados para la adquisición e identificación de los requisitos que demandó el proyecto, con el fin de identificar beneficios, desventajas y características básicas de las tecnologías implementadas; brindando de esta manera un contexto más claro y sustentable para el desarrollo del mismo y sus fases posteriores.

### A. Revisión Bibliográfica

Se desarrolló una revisión bibliográfica en aras de identificar el contexto científico relacionado con el filtrado de imágenes y secuencias de imágenes (Videos), con el fin de definir de manera directa los modelos, estrategias, técnicas y tecnologías que se han visto implicadas en el contexto del control, a través del PDI a nivel mundial, permitiendo tener un sustento para captar y aplicar de forma beneficiosa los valores identificados al desarrollo del trabajo.

Para efectos del buen planteamiento del ejercicio bibliográfico, en caso de continuar se definirían lineamientos y características tenidas en cuenta para la selección de la información como:

Ventana cronológica de búsqueda, Número mínimo referencia, Marco geográfico, tipo de contenido (**Científico – EJ: Filtrado de imagen y videos de tipo astronómico mediante el PDI**) se obtuvieron 5 artículos [7-11] de los cuales se seleccionaron los 3 métodos a implementar en el proceso de denoising de los 16 sets de imágenes.

### B. Metodología

Debido a las mediciones obtenidas las cuales cuentan con características similares, sobre los 16 sets de imágenes se realizó la ejecución de este modelo, solo en 3 sets iniciales sobre los cuales se desarrollaron múltiples cambios en las variables inherentes de los métodos de filtrado, posteriormente se realiza un análisis cualitativo y la propuesta de métricas para su evaluación posterior (cuantitativo).

Basado en el conocimiento de filtros de imágenes estáticas se partió por la aplicación de 5 métodos [12] de filtrado variando sus parámetros, buscando así un punto de referencia para el posterior filtrado de video.

Finalizando con la aplicación de métodos de filtrado relacionados con video (VIDOSAT, VBM3D, SALT y proxTV), sobre los cuales se generaron los resultados dispuestos en Repositorio GitHub (<https://github.com/ANDRESHZ/FiltradoTarea1>).

## III. FILTRADO DE IMÁGENES

Este proceso se dividió en dos fases, el filtrado de imágenes estáticas usando lo indicado en [12] con el fin de aplicar diferentes métodos y el filtrado de secuencias de imágenes, los resultados de las mismas están dispuestas en la carpeta Evidencias/.

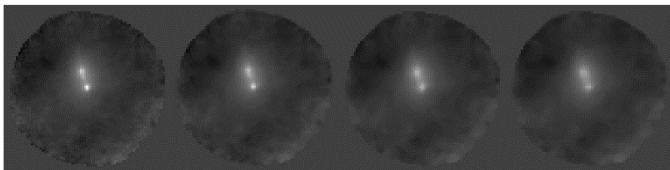
La idea base de las imágenes a filtrar es mantener los objetos blancos de alta intensidad móviles y el punto de referencia central, con los cuales posteriormente se pretende dar solución a problemas de convolución (PSF) y tracking

### A. Filtros estáticos

Se realizó el filtrado de imágenes con diferentes filtros estudiados en [12], sobre los cuales se aplicaban cambios en las variables de cada uno de ellos con el fin de obtener los mayores beneficios en la eliminación de ruido.

#### 1) Filtro de media

Se implementó el filtro obteniendo la media de los píxeles vecinos y así obtener un cambio más suave en 2 dimensiones, el cual tiene como parámetro de variación el tamaño de la vecindad ( $Nb$ )  $n \times m$ , sobre la cual se promedian los valores, debido a la naturaleza y al tamaño de las imágenes del dataset ( $79 \times 79$  píxeles) solo se implementaron en tres tamaños  $n, m \in [2,3,4]$ . Dando como resultado:



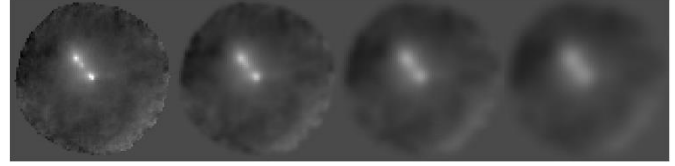
**Fig1.** Resultado filtro media. a) Original b)  $tam(Nb) = 2 \times 2$  c)  $tam(Nb) = 3 \times 3$  d)  $tam(Nb) = 4 \times 4$

$3 \times 3$  d)  $tam(Nb) = 4 \times 4$

El uso de vecindades de diferentes tamaños afecta la calidad de la imagen perdiendo demasiada información, como es evidenciable en el cambio de la imagen a) a la d) de la figura 1.

#### 2) Filtro Gaussiano

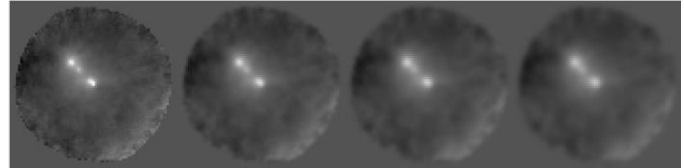
Filtro basado en un kernel teniendo en cuenta una distribución gaussiana con tamaño  $n \times m$ , el cual recorre la imagen a nivel global, aplicándose en toda la matriz



**Fig2.** Filtro Gaussiano. a) Original b)  $Kernel=2 \times 2$  c)  $Kernel=3 \times 3$  d)  $Kernel=4 \times 4$

#### 3) Filtro anisotrópico para difusiones

En este modelo se filtran mejor los bordes, y es de tipo iterativo así puede ser aplicado múltiples veces, además de tener poder para cambiar el tipo de conducción, pese a esto se mantuvieron los resultados dispuestos en la figura



**Fig3.** Resultado filtro Anisotrópico. a) Original b)  $iteración=2$  c)  $iteración=3$  d)  $iteración=4$

#### 4) Variación Total

Implementando la función de variación total en Matlab descrita por [13], dada por la función a nivel discreto:

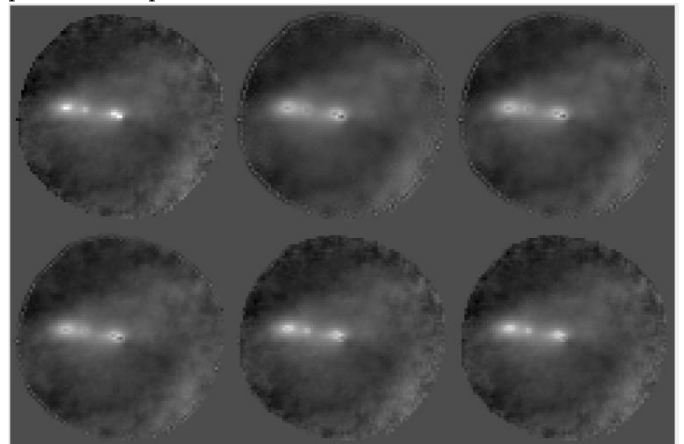
$$TV_{L_1}(f) = \sum_{i,j} |\nabla f_x| + |\nabla f_y| = \sum_{i,j} |f_{i+1,j} - f_{i,j}| + |f_{i,j+1} - f_{i,j}|$$

$$\min_f \|Af - g\|_2^2 + \lambda^2 TV(f)$$

Donde  $f$  es la imagen,  $A$  el modelo y  $g$  imagen con ruido. Lo que al final usando **SVD** sería [14]:

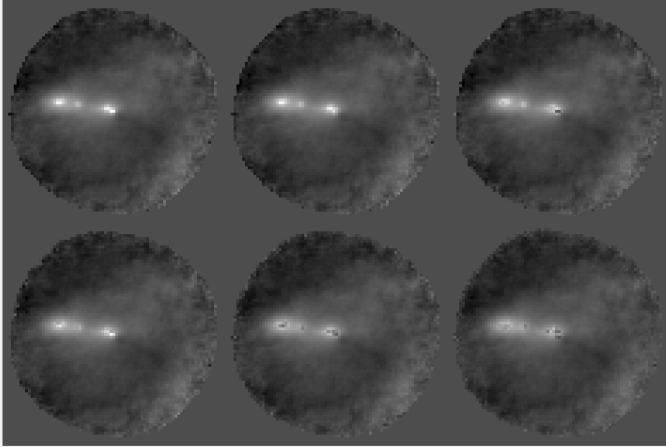
$$\hat{f}_k = \sum_{i=1}^n TV(f) \frac{u_i^T b}{\sigma_i} v_i$$

Con este algoritmo iterativo se tienen factores como  $\lambda$  y el número de iteraciones. Así se procedió modificar ambos parámetros para obtener



**Fig4.** Resultado variación Total con norma Gradiente. Iteraciones=2 a) Original b)  $\lambda=0.005$  c)  $\lambda=0.009$  d)  $\lambda=0.02$  e)  $\lambda=0.07$  f)  $\lambda=0.09$

Donde se puede observar que el regularizador al crecer disminuye la fuerza de la aplicación del filtro. Y la última variación desarrollada es respecto al número de iteraciones, donde estas dos tienen una relación de aplicación, por ello se debió aumentar el  $\lambda = 0.3$ , para que no afectara fuertemente las imágenes.

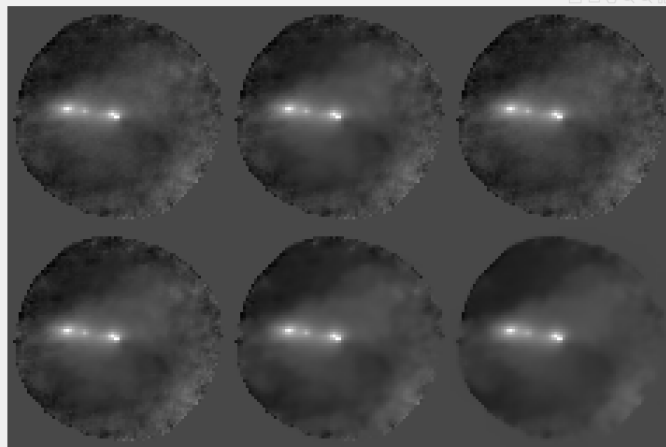


**Fig5.** Resultado variación Total con norma Gradiente.  $\lambda = 0.3$  a) Original b)  $iter = 3$  c)  $iter = 5$  d)  $iter = 9$  e)  $iter = 13$  f)  $iter = 20$

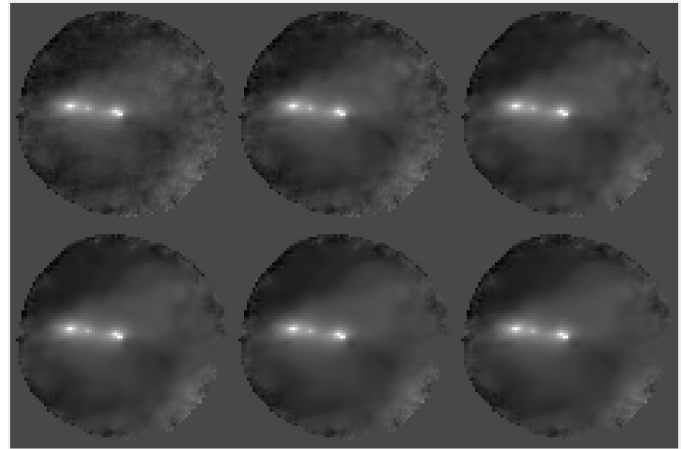
Obteniendo un mejor resultado que los anteriores filtros. Además de tratarse de un problema convolucional y por ende ayuda a otras características como la reducción del efecto de la PSF.

#### 5) Non-Local means

Sobre este filtro aprovecharemos la generación de ventanas de acción para predecir los valores y evitar cambios abruptos sobre la imagen, pero a su vez permite mantener los bordes si la ventana es lo suficientemente grande para realizar el ajuste de los mismos. Basado en lo dispuesto en [10] los valores de ventana de búsqueda (SW) no deberían superar más de 30 veces los detalles deseados y las ventanas de comparación (CW) deben ajustarse a las áreas con espacios planos a reconstruir



**Fig6.** Resultado non-local means. CW=3 SW=15 a) Original b) Suavizado = default c) Suavizado =  $1E^{-4}$  d) Suavizado =  $2E^{-4}$  e) Suavizado =  $5E^{-4}$  f) Suavizado =  $1E^{-3}$



**Fig7.** Resultado non-local means. SW = 15 Suavizado =  $4E^{-4}$  a) Original b) CW = default c) CW = 3 d) CW = 5 e) CW = 7 f) CW = 11

Siguiendo la recomendación se obtiene que una mezcla de las variables SW=15 es suficiente para captar los objetos, CW=3 para mantener las áreas homogéneas. En caso de necesitar detalle en zonas específicas mantener un suavizado bajo es lo esencial, en nuestro caso favorece a los puntos blancos (objetos de estudio) tener un valor 0.0004.

#### B. Filtros sobre Video:

Se identifico 4 filtros sobre video conocidos como (VIDOSAT, VBM3D, SALT y proxTV) de los cuales escogí los 3 primeros para aplicar a las secuencias de imágenes, primero debemos conocer el funcionamiento de los algoritmos. Cabe aclarar los 3 se centran en el desarrollo de parches o ventanas, para posteriormente por similitud reconocer cual es información real de la imagen.

##### 1) VIDOSAT (High-dimensional Sparsifying Transform Learning for Online Video Denoising):[8]

A grandes rasgos el algoritmo realiza un aprendizaje de transformación en línea a la eliminación de ruido por video, aprovechándose de las diferencias entre las imágenes y ventanas de comparación. Frente a la eliminación de ruido de video en línea basado en el aprendizaje de dispersión de alta dimensión para parches temporales [8]. Los parches se construyen a partir de marcos 2 dimensiones correspondientes en cuadros sucesivos o mediante una técnica de coincidencia de bloques en línea, creando así la base de su trabajo, siendo esta la pila tipo FIFO (First In First Out) siguiendo el raciocinio de un video normal, considerando esto crean una pila espacio temporal sobre una ventana, sobre la cual miran la redundancia de los datos, en las ventanas seleccionadas, y usa  $n$  vecinos de imágenes, con las cuales entrenan pequeños batch de información, que se encargan de reconocer el ruido (pues el mismo no suele ser estático y no coincide con grandes cantidades de píxeles aglomerados).

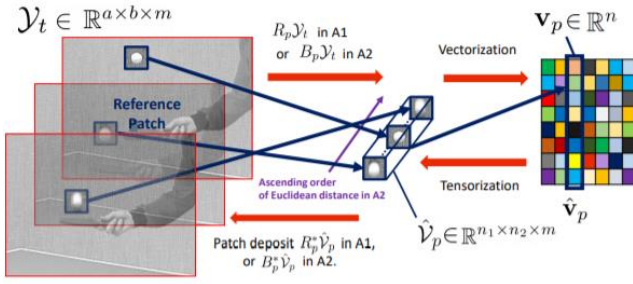


Fig8. Modelo de ventanas de VIDOSAT para entrenamiento de vectores [8]

## 2) VBM3D (Implementation of the VBM3D Video Denoising Method and Some Variants) [10]

Es un algoritmo basado en BM3D (Blocking matching an 3D Filtering) el cual está ampliamente implementado en matlab e incluso tiene librerías de instalación [www.cs.tut.fi/~foi/GCF-BM3D](http://www.cs.tut.fi/~foi/GCF-BM3D), este modelo aprovecha la creación de ventanas para obtener características relevantes de la información real de la imagen y discernir el ruido. Debido a que estos bloques se superponen, para cada píxel obtienen muchas estimaciones diferentes que deben combinarse [14].

La extrapolación a 3D es basada en la generación de cluster de imágenes así el algoritmo ofrece una buena compensación entre el rendimiento de eliminación de ruido y la complejidad computacional [15], es divisible en los siguientes pseudo-pasos:

- Buscar ventanas similares en la secuencia, agrupándolos en pilas 3D.
- Aplicar una transformación de dominio lineal 3D al bloque 3D (vectores de características)
- Reducir los coeficientes transformados (normalizar características)
- Aplicar la transformación inversa (con un pre filtrado de Wiener).
- Agregar los parches resultantes en el video (ahora todo lo que pase por allí y no sea reconocido subestimado)
- Volver a repetir...

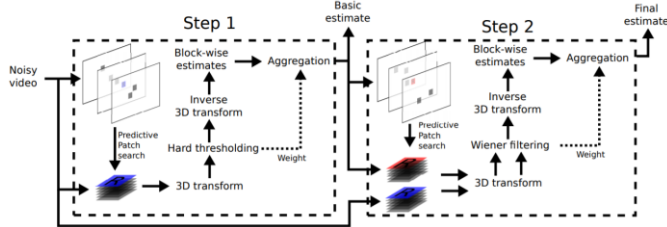


Fig9. Modelo VBM3D, estimación Básica y Final [8]

## 3) SALT (joint Adaptive Sparsity and Low-Rankness on the Fly: An Online Tensor Reconstruction Scheme for Video Denoising) [9]

Utilizando el modelo de VIDOSAT ellos reconstruyen con una pila tipo FIFO y generan los procesos previos (clústeres de Imágenes 3D separadas por tiempo), la ventaja es que realizan un entrenamiento en línea de un algoritmo para la eliminación de información aleatoria no relevante y de vectores característicos no utilizados. Por tanto, la mitigación de ruido de video basada en SALT demuestra baja latencia y potencialmente puede manejar videos en tiempo real [9].

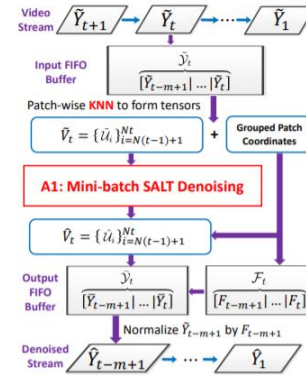


Fig9. Modelo de SALT Denosign [9]

La arquitectura es prácticamente idéntica a VIDOSAT, pero agrega antes del Batch de vectores característicos, unos modelos de filtrado colaborativo, basado en la similitud (kNN K-nearest neighbor). Lo cual permite ir eliminando vectores característicos no utilizados y agregando valor a los que si son relevantes.

## IV. RESULTADOS

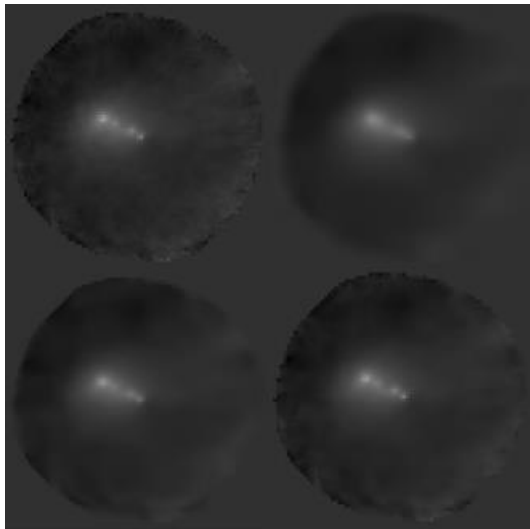
La información de las pruebas está dispuesta en link GitHub indicado previamente para obtener los parámetros adecuados de los videos y debido a la cantidad de parámetros dispuestos para la ejecución de pruebas, se hizo necesario la ejecución de pruebas sobre un único set de videos, múltiples veces con los métodos dispuestos, incluido como lo expresaban en los paper las mezclas entre VIDOSAT y SALT, se ejecutaron 207 casos con una media de tiempo de procesamiento de 10 mins para un total de 34.5 horas continuas de procesamiento. Lo primero fue preparar las imágenes para el procesamiento regularizándolas a imágenes con valores entre 0 y 255

$$\hat{p}_{ij} = \frac{p_{ij} - \min(V(p_{ijk}))}{\max(V(p_{ijk})) - \min(V(p_{ijk}))}$$

Donde  $p_{ij}$  es el pixel en cuestión,  $V$  es el video completo (30 imagenes = k) ósea una función de pixeles en k imágenes y  $p_{ijk}$  es el pixel de la imagen  $k$  del video ( $V$ ), se hizo a nivel general del video para que no hubiese cambios de intensidades en las diferentes fases.

Se obtuvieron grandes resultados de los videos de SALT por encima de VIDOSAT y VBM3D como se muestra en la imagen 11.

El uso de pocos parámetros por parte de VBM3D no permitió generar buenos resultados, mientras que durante el ajuste de todas iteraciones se logró obtener mejoría con los parámetros tanto de VIDOSAT como SALT, los demás resultados los podrá encontrar en GITHUB. A nivel aislado VIDOSAT tuvo algunos filtrados buenos en comparación de VBM3D, pero en ningún caso supero al SALT.



**Fig9.** Mejores resultados a nivel visual, con mismos parámetros a) Imagen Referencia b) VBM3D c) VIDOSAT y d) SALT

## V. CONCLUSIONES

Se identificó la Superioridad del Método SALT para filtrar secuencias de imágenes, ratificando lo indicado en [9], donde el método representa una mejora respecto de los otros dos, basado en el concepto de eliminar y propender por el uso de vectores característicos más similares, por medio de la clusterización usando (kNN).

Los algoritmos VIDOSAT y VBM3D, difieren bastante en resultado aunque manejen preceptos parecidos, lo que apalanca a VIDOSAT es el cambio de parámetros de características, mientras el filtro Wiener usado en la deconvolución por parte de VBM3D, podría ser el causante de su pérdida.

Bajo las condiciones actuales los métodos implementados no cuentan con grandes avances respecto el modelo de filtrado Non-local means, posiblemente se deba a la naturaleza de las imágenes que favorece a este tipo de filtrado. Además de ser imágenes pequeñas y de poco contraste respecto a texturas y formas se refiere (precepto básico de búsqueda de los métodos estudiados).

## VI. REFERENCES

- [1] Guesalaga A., Neichel B., Correia C.M., Butterley T., Osborn J., Masciadri E., Fusco T. and Sauvage J.-F. (2017). Online estimation of the wavefront outer scale profile from adaptive optics telemetry. *Monthly Notices of the Royal Astronomical Society*, 465(2), 1984-1994.
- [2] Wang, Y. M., Li, Y., & Zheng, J. B. (2010, June). A camera calibration technique based on OpenCV. In *The 3rd International Conference on Information Sciences and Interaction Sciences* (pp. 403-406). IEEE.
- [3] Maire J., Ziad A., Borgnino J., Martin F., 2007, *MNRAS*, 377,1236
- [4] Huerfano Z., B. A., Numpaque, H., & Díaz, L. (2012). Algorithm Development for Controlling Movement of a Robotic Platform by Digital Image Processing. *Entre Ciencia e Ingeniería*, 6(12), 19–23. <http://biblioteca.ucp.edu.co/OJS/index.php/entrecei/article/view/598/563>
- [5] R. W. P. Gonzalez, «Tema 3,4,» de *Procesamiento digital de imágenes*, Addison-Wesley, 1996, pp. 89-269.
- [6] E. C. Nieto, «Manufacturing and automation,» *ingeniería e investigación*, vol. 26, n° 3, pp. 120-128, 2006.
- [7] Barbero, A., & Sra, S. (2018). Modular proximal optimization for multidimensional total-variation regularization. *The Journal of Machine Learning Research*, 19(1), 2232-2313.
- [8] Wen, B., Ravishankar, S., & Bresler, Y. (2018). VIDOSAT: High-dimensional sparsifying transform learning for online video denoising. *IEEE Transactions on Image processing*, 28(4), 1691-1704.
- [9] Wen, B., Li, Y., Pfister, L., & Bresler, Y. (2017). Joint adaptive sparsity and low-rankness on the fly: an online tensor reconstruction scheme for video denoising. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 241-250).
- [10] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3D transform-domain collaborative filtering. In *EUSIPCO*, 2007
- [11] Tejos|N, Cristian A. IEE3784 / IIC378 –Procesamiento Avanzado de Imágenes. Clases 1 a la 8 Pontificia Universidad Católica de Chile.II-2020
- [12] Chambolle et al. An introduction to Total Variation for Image Analysis, 2009. <hal-00437581> <https://hal.archives-ouvertes.fr/hal-00437581/document>
- [13] Thanh, D. N. H., Prasath, V. B. S., & Thanh, L. T. (2018). Total Variation L1 Fidelity Salt-and-Pepper Denoising with Adaptive Regularization Parameter. In *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*. IEEE. Retrieved from <https://doi.org/10.1109%2Fnics.2018.8606870>
- [14] Dabov, Kostadin; Foi, Alessandro; Katkovnik, Vladimir; Egiazarian, Karen (16 July 2007). "Image denoising by sparse 3D transform-domain collaborative filtering". *IEEE Transactions on Image Processing*. 16 (8): 2080–2095. Bibcode:2007ITIP...16.2080D. CiteSeerX 10.1.1.219.5398. doi:10.1109/TIP.2007.901238
- [15] Ehret, T., & Arias, P. (2020). Implementation of the VBM3D video denoising method and some variants. *arXiv preprint arXiv:2001.01802*.