



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

IMPLEMENTACIÓN Y EVALUACIÓN DE ALGORITMOS DE SUPER-RESOLUCIÓN  
PARA IMÁGENES TOMADAS CON NANO SATÉLITES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

RODOLFO ALEJANDRO JARA PINOCHET

PROFESOR GUÍA:  
MARCOS DIAZ QUEZADA

MIEMBROS DE LA COMISIÓN:  
CLAUDIO ESTÉVEZ MONTERO  
JUAN MANUEL BARRIOS

Este trabajo ha sido parcialmente financiado por proyecto ANILLO ACT 1405

SANTIAGO DE CHILE  
2018



RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO  
POR: RODOLFO ALEJANDRO JARA PINOCHET  
FECHA: 2018  
PROF. GUÍA: DR. MARCOS DIAZ QUEZADA

## IMPLEMENTACIÓN Y EVALUACIÓN DE ALGORITMOS DE SUPER-RESOLUCIÓN PARA IMÁGENES TOMADAS CON NANO SATÉLITES

Esta memoria se enmarca en el análisis de algoritmos de Súper Resolución (SR). Estos algoritmos recibirían imágenes tomadas por un futuro satélite *Cubesat* desde la órbita terrestre, obteniéndose una imagen con mayor resolución.

El principal objetivo de esta memoria consiste en estudiar el coste en recursos, rendimiento y otras consideraciones de estos algoritmos. Así se podrá determinar cuál algoritmo de SR de los considerados pueda ser utilizado con ventajas en una plataforma *Cubesat*.

Para lograr este objetivo se utiliza las experiencias de operación del primer *Cubesat* Chileno (SUCHAI 1). Usando esta información se ha podido determinarse cuáles serían los requerimientos que se deben cumplir en los futuros SUCHAI de manera de disponer de la capacidad de obtener imágenes en SR.

Se realizaron evaluaciones de los algoritmos de SR implementándose una interfaz que permitió comparar visualmente, correr reiteradamente (cambiando sus parámetros) los algoritmos en una plataforma Matlab, de manera de evaluar sus prestaciones y obtener una estimación de la mejora en resolución resultante de la imagen final, el tiempo empleado en su ejecución y la memoria utiliza. Como complemento se utilizó un método llamado *Mapas de Calor* que permite construir comparaciones de un dataset (se dispusieron de 100 imágenes y se usó una métrica llamada *Peak Signal to Noise Ratio* -PSNR-) construido *Ad hoc*.

Con los datos obtenidos se sugieren escenarios sobre los cuales utilizar los algoritmos a ser ejecutado a bordo del satélite (considerando sus características) y se postula que para obtener mejores prestaciones de SR lo práctico es bajar las imágenes y ejecutar en tierra (lo cual probablemente requerirá mucha más capacidad de cálculo). Se ven algunos pro y contra de la SR usando *Deep Learning* y se menciona como posible ampliación del campo de la SR a la combinación con *Mapas Digitales de Elevación* (DEM).



*“Inferencia.*

*Si un adulto quiere volver a ser niño,  
no debería comer verdura,  
ya que esta hace que los niños crezcan  
y se hagan adultos”*

---

*[Inteligencia Artificial,  
Alison Gopnik, IyC Sept2017, p71]*

*“Cuando todo parezca estar en tu  
contra,  
recuerda que los aviones despegan con  
el aire en contra,  
no a favor”*

---

*[Henry Ford]*

*“¿Y sientes que has hecho tu mejor  
esfuerzo en este caso, verdad?  
¿Que has puesto en práctica todo tú  
considerable ingenio?  
¿Que no has dejado ningún abismo de  
astucia inexplorada en la búsqueda de  
ese recuerdo?”*

---

*[Harry Potter y el Príncipe Mestizo,  
Rowling, Jk, pag 175]*

*“The best bad idea we have, sir. By far”*

---

*[ARGO (2012)]*

*¡Lha gyalo!*



# Tabla de Contenido

|   |           |
|---|-----------|
| <b>1. Introducción</b>  | <b>1</b>  |
| 1.1. Fundamentación . . . . .                                   | 1         |
| 1.2. Hipótesis . . . . .  | 2         |
| 1.3. Objetivo General . . . . .                                 | 2         |
| 1.4. Objetivos Específicos . . . . .                            | 2         |
| 1.5. Metodología . . . . .                                      | 3         |
| 1.6. Plan de trabajo . . . . .                                  | 3         |
| <b>2. Antecedentes</b>  | <b>5</b>  |
| 2.1. Algoritmos de Súper-Resolución . . . . .                   | 5         |
| 2.1.1. Algoritmos de SR usados actualmente . . . . .            | 7         |
| 2.1.2. Algoritmos más promisorios de SR . . . . .               | 8         |
| 2.2. <i>Cubesat</i> (nano Satélite estándar) . . . . .          | 9         |
| 2.3. Percepción Remota . . . . .                                | 11        |
| 2.3.1. Resolución de imágenes tomadas desde satélites . . . . . | 12        |
| 2.3.2. Distancia focal . . . . .                                | 14        |
| 2.3.3. Razón Focal . . . . .                                    | 14        |
| 2.4. Herramienta de cuantificación: PSNR . . . . .              | 15        |
| <b>3. Algoritmos de Súper Resolución Implementados</b>          | <b>16</b> |
| 3.1. Iterative Back-Projection -IBP- . . . . .                  | 18        |
| 3.2. Projection Onto Convex Sets -POCS- . . . . .               | 19        |
| 3.3. ROBUST . . . . .   | 21        |
| 3.3.1. Porque elegir la mediana de los píxeles . . . . .        | 21        |
| 3.3.2. Procedimiento de Detección de Sesgo . . . . .            | 22        |
| 3.4. SR y Deep Learning (DL) . . . . .                          | 23        |
| 3.5. SR, Modelos de Elevación Digital (DEM) y RGB-D . . . . .   | 24        |
| <b>4. Implementaciones y análisis de resultados obtenidos</b>   | <b>26</b> |
| 4.1. Ejecución de las implementaciones . . . . .                | 26        |
| 4.2. Interfaz . . . . .   | 27        |
| 4.3. Pruebas . . . . .  | 29        |
| 4.3.1. Grupo imágenes Eolian . . . . .                          | 29        |
| 4.3.2. Grupo imágenes Malvinas . . . . .                        | 32        |
| 4.3.3. Grupo imágenes Textos . . . . .                          | 33        |
| 4.3.4. Grupo imágenes Emily . . . . .                           | 34        |

|   |           |
|---|-----------|
| 4.3.5. <i>Mapas de Calor</i> de algoritmos considerados . . . . . | 35        |
| 4.4. Análisis de resultados obtenidos . . . . .                   | 39        |
| 4.5. Consideraciones en los <i>Cubesat</i> . . . . .              | 40        |
| 4.6. Situación de casos entre algoritmos y escenarios . . . . .   | 41        |
| <b>5. Conclusiones y trabajos futuros</b>                         | <b>43</b> |
| 5.1. Algoritmos de SR que utilizan DL . . . . .                   | 45        |
| 5.2. Algoritmos de SR, DEM y RGB-D . . . . .                      | 46        |
| <b>Bibliografía</b>   | <b>47</b> |
| <b>Anexo: Códigos</b>   | <b>54</b> |
| <b>Anexo: Acrónimos</b>   | <b>55</b> |

# Índice de Tablas

|   |    |
|---|----|
| 4.1. Resumen rendimientos algoritmos de SR para grupo imágenes Eolian (10 imágenes) . . . . .   | 31 |
| 4.2. Resumen rendimientos algoritmos de SR para grupo imágenes Malvinas (10 imágenes) . . . . . | 33 |
| 4.3. Resumen rendimientos algoritmos de SR para grupo imágenes Textos (30 imágenes) . . . . .   | 34 |
| 4.4. Resumen rendimientos algoritmos de SR para grupo imágenes Emily (53 imágenes) . . . . .    | 35 |
| 4.5. Sugerencias de aplicación de algoritmos de SR . . . . .                                    | 42 |

# Índice de Ilustraciones

|  |    |
|--|----|
| 2.1. Reconstrucción usando SR . . . . .  | 6  |
| 2.2. Ejemplo de SR en zona de estacionamiento . . . . .                        | 6  |
| 2.3. Sputnik: Primer satélite artificial . . . . .                             | 9  |
| 2.4. <i>Cubesat</i> : Características . . . . .                                | 10 |
| 2.5. Suchai 1: Maqueta . . . . .   | 11 |
| 2.6. Línea de tiempo del desarrollo de la observación terrestre . . . . .      | 12 |
| 2.7. Relación FOV con altura y ancho de zona observada . . . . .               | 13 |
| 2.8. Variación de la distancia focal . . . . .                                 | 14 |
| 3.1. Implementación IBP . . . . .  | 19 |
| 3.2. Implementación de POCS . . . . .  | 20 |
| 3.3. Implementación ROBUST . . . . .   | 21 |
| 3.4. Deep Learning mejorando resolución . . . . .                              | 23 |
| 3.5. RGB-D: Determinación de profundidad en imagen a corta distancia . . . . . | 24 |
| 4.1. SR e imágenes a color . . . . .   | 27 |
| 4.2. Interfaz despliegue de algoritmos de SR . . . . .                         | 27 |
| 4.3. Interfaz variando conjunto de entrada y parámetros . . . . .              | 28 |
| 4.4. Vista aérea del DIE y zoom de zona Eolian . . . . .                       | 29 |
| 4.5. Generación de las imágenes como fuente de datos . . . . .                 | 30 |
| 4.6. Algoritmo IBP contra interpolación bicubica e imagen ideal . . . . .      | 30 |
| 4.7. Comparación entre POCS y ROBUST lento . . . . .                           | 31 |
| 4.8. Las islas Malvinas captadas por el satélite Suchai 1 . . . . .            | 32 |
| 4.9. Comparación de imagen base Malvinas y sus SR generadas . . . . .          | 33 |
| 4.10. Comparación de imagen base texto y sus SR generadas . . . . .            | 34 |
| 4.11. Comparación de imagen base Emily y sus SR generadas . . . . .            | 34 |
| 4.12. 100 imágenes de la superficie terrestre . . . . .                        | 35 |
| 4.13. <i>Mapas de Calor</i> Algoritmos: IBP-POCS-ROBUST . . . . .              | 36 |
| 4.14. <i>Mapas de Calor</i> IBP ordenados y sus imágenes . . . . .             | 37 |
| 4.15. <i>Mapas de Calor</i> POCS ordenados y sus imágenes . . . . .            | 37 |
| 4.16. <i>Mapas de Calor</i> ROBUST ordenados y sus imágenes . . . . .          | 38 |
| 4.17. Rangos PSNR de los algoritmos . . . . .                                  | 38 |
| 4.18. Compresión Pixspan . . . . .   | 41 |

# Capítulo 1

## Introducción

### 1.1. Fundamentación

Desde el principio de la ciencia espacial el desarrollo y operación de un satélite ha sido una actividad de alta tecnología. Se han combinado en el esfuerzo astronáutico disciplinas como; electrónica, aerodinámica, astrodinámica, ciencia de materiales, computación y física entre otras.

Al usar la tecnología de motores cohete de combustible líquido se utiliza la llamada ecuación de Tsiolkovski, que relaciona la masa inicial (que es casi todo combustible) con la masa final (que es la carga útil) y con la velocidad de los gases expelidos, entregando la diferencia de velocidad que recibe el cohete, impulsando el ingenio en una trayectoria que permite alcanzar la altitud deseada (por ejemplo, una órbita terrestre baja -LEO-).

Así podemos determinar que al aumentar la masa de la carga útil significa incrementar significativamente el combustible (y por ende la masa del cohete) y por supuesto sus costos. Actualmente poner un kilogramo en órbita LEO tiene un precio bastante elevado (a junio del 2017 es del orden de US\$ 100.000, en el momento del lanzamiento del SUCHAI 1).

Con un enfoque de desarrollo basado en estándares se pueden obtener plataformas satelitales y realizar pruebas de concepto en el ambiente espacial por una fracción del coste, y en tiempos más reducidos validar tecnologías usando cargas útiles modulares. A pesar de los avances y de la estandarización todavía es costoso poner en órbita un satélite tradicional, razón por la cual se trata de optimizar su masa, volumen y prestaciones.

Motivado por los costos y complejidades de los satélites tradicionales en los últimos años se desarrolló un nuevo tipo, los *Cubesat* [1] (los cuales son una variante de los nano satélites, pero que se construyen siguiendo estándares y que ya tienen un mercado de suministros y servicios especializados que los soporta). En estos pequeños ingenios cuya unidad más básica es un cubo de 10 cm de lado, es posible implementar circuitos eléctricos y sensores que definen una carga útil (en inglés payload), que pueden funcionar en el ambiente espacial.

Una de las principales ventajas de los *Cubesat* es poder realizar pruebas de concepto que permitan validar nuevas tecnologías al menor costo (y cuyo tiempo de diseño-desarrollo y despliegue sea mínimo también). Si la carga útil es un telescopio, el tamaño del espejo primario determina la capacidad de captar luz y si se necesita mejorar la resolución esto implica aumentar el diámetro de la óptica y por ende el volumen y peso de toda la estructura del satélite que lo utilizará.

Si fuera posible después de recibir un grupo de imágenes, procesarlas y aumentar la resolución obtenida sin necesidad de incrementar el tamaño de la óptica se podría mejorar las capacidades de un satélite de observación manteniendo su masa y volumen y por ende el precio del ingenio espacial. Existe un conjunto de técnicas que mejoran la calidad de la imagen recibida por medios digitales, estas son llamadas Súper-Resolución (SR) [2, 3] y ha sido ya aplicada con resultados provisorios en varios satélites [4, 5, 6, 7, 8, 9].

El lanzamiento, puesta en órbita terrestre y los primeros meses de vida del primer *Cubesat* Chileno (SUCHAI 1) se produjo durante el segundo semestre del 2017. En este mismo periodo de tiempo se estaba realizando el diseño de los futuros SUCHAI. Al analizar las posibles cargas útiles a considerar se decantó por aquellos desarrollos que pudieran ser efectuados en el Laboratorio de Exploración Espacial y Planetaria (SPEL) y que tuvieran las características para ser instalados en una plataforma *Cubesat*.

## 1.2. Hipótesis

Con la experiencia que se ha adquirido con la operación del SUCHAI 1 se plantea la pregunta: sería posible utilizar algún modulo (de software) que mejorase la calidad de las imágenes captadas por la cámara en uso en el SUCHAI (o en futuros modelos), utilizando su electrónica, y sus características de vuelo.

## 1.3. Objetivo General

Evaluar el desempeño de algoritmos de Súper-Resolución para mejorar la resolución de imágenes obtenidas en dispositivos ópticos de nano satélites.

## 1.4. Objetivos Específicos

- Implementar tres algoritmos de Súper-Resolución
- Determinar de entre los algoritmos de Súper-Resolución implementados cuale(s) es(son) el(los) más adecuado(s) para su uso en nano satélites
- Analizar el impacto del uso de algoritmos de Súper-Resolución en el diseño de nuevos nano satélites

## 1.5. Metodología

- Implementación de algunos algoritmos de Súper-Resolución  
Prototipos en Matlab
- Evaluados sus características (rendimiento, calidad obtenida, complejidad de la electrónica involucrada)  
¿Cuánto se demora en procesar sus imágenes?  
¿Cuánto mejor es la resolución?  
¿Cuánta memoria utiliza el algoritmo?
- Determinar las particularidades que significaría implementarlos en un *Cubesat*  
¿Qué cambios hay que implementar en los satélites SUCHAI de manera de utilizar el poder de la SR?  
¿Se requiere modificar/agregar/eliminar procedimientos/algoritmos/electrónica para poder utilizar la SR?  
¿Se necesita adaptar alguna cosa en la estación terrena?

La metodología considera la posibilidad de iterar en algunas labores, ya que por ejemplo al evaluar los algoritmos, si sus consumos (tiempo, potencia, hardware específico) resultan ser excesivos (el valor exacto solo se podrá determinar en un último momento cuando se detallen las características de los futuros SUCHAI -que electrónica se utilizara-), se necesitaría volver a re-considerar algoritmos previamente descartados que no siendo óptimos tengan un rendimiento más acorde a lo necesitado para ser implementado en los próximos SUCHAI.

## 1.6. Plan de trabajo

1. Investigación y búsqueda bibliográfica (2 meses) La primera etapa consiste en la investigación para encontrar las mejores referencias tanto de los algoritmos de SR como de las especificaciones de los *Cubesat*
2. Análisis y evaluación de las alternativas más promisorias (2 meses) La segunda etapa es identificar los métodos de enfoque y sus implicancias en los algoritmos que posteriormente se elegirán y programarán, ya que no es lo mismo implementar un algoritmo por interpolación que usando redes neuronales.
3. Desarrollo de la codificación de los algoritmos elegidos (3 meses) La tercera etapa comprende la programación y estudio de tres algoritmos candidatos para determinar sus características de tiempo de ejecución, consumo de potencia, y otros por determinar
4. Determinación de performance de los algoritmos (3 semana) La cuarta etapa es la evaluación final de los algoritmos comparándolos entre si usando parámetros previamente establecidos (indicados en la metodología).
5. Análisis de requerimientos de los algoritmos para el caso de un *Cubesat* (2 meses) La quinta etapa es la determinación en detalle de los parámetros de rendimiento de los mejores algoritmos elegidos, para el caso de una plataforma *Cubesat*
6. Alternativas de implementación de Algoritmos de SR en *Cubesat* y sus resultados (1 mes). La última etapa es evaluar cuál es una versión adecuada (software o hardware)

que ejecute el algoritmo elegido y permita obtener un máximo de rendimiento en la plataforma *Cubesat*, bajo las restricciones imperantes.

Se considera como una actividad que cruza todos los meses del plan de trabajo a la documentación (o escritura de la memoria), ya que dejarla para último momento podría aumentar en exceso el tiempo dedicado al trabajo de título, mientras que si se va documentando a medida que se avanza se va registrando las labores realizadas apenas las ideas se van ordenando.

# Capítulo 2

## Antecedentes

### 2.1. Algoritmos de Súper-Resolución

Súper Resolución (SR) es un conjunto de técnicas y algoritmos que tomando como base imágenes de menor resolución pueden obtener una imagen de mayor resolución [10, 11, 12, 13].

Los algoritmos de SR se puede agrupar en primera aproximación como; Algoritmos basados en interpolación (resampling) de una sola imagen (monoframe); una segunda clase está compuesta de varias capturas de la misma escena (multiframe) procesadas adecuadamente; la tercera clase está basada en modelos de *machine learning* que utilizan redes neuronales convolucionales que tienen etapas de aprendizaje y que por medio de inferencias pueden aumentar la resolución de las imágenes de entrada [14, 15, 16].

Como antecedente podemos mencionar que la obtención de una imagen tiene las siguientes características [17, 18]:

1. Muestreo: transformación del sistema continuo de la escena real a un sistema discreto ideal.
2. Transformación geométrica: aplicación de una serie de transformaciones, como traslación o rotación, debido a la posición y al sistema de lentes de la cámara, de manera que los detalles de la escena llegan a cada zona del sensor.
3. Desenfoque: debido al sistema de lentes o al movimiento existente en la escena durante el tiempo de integración, que determina como los detalles se aglutinan alrededor del sensor.
4. Submuestreo: finalmente, el sensor solo integra el número de píxeles de los que dispone.

Durante el proceso de captura, los detalles de la escena son recibidos por los píxeles del sensor, de modo que al disponer de varias imágenes que están separadas sub-píxel cada una de ellas tiene elementos ligeramente distintos. Por ello, la SR (multiframe) se basa en tratar de encontrar la relación entre los píxeles de diferentes capturas que muestrean los diversos fragmentos del cuadro, y así poder crear un nuevo entramado de píxeles con más información [19]. Tal como se muestra en la Figura 2.1.

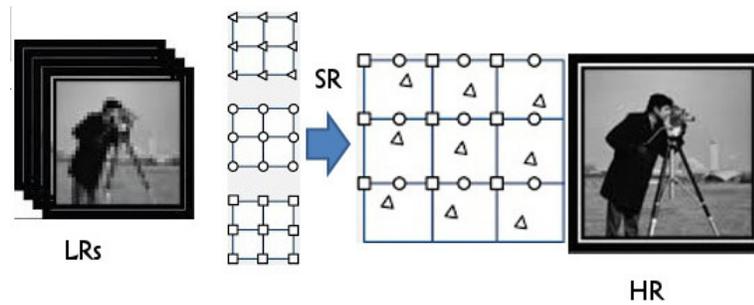


Figura 2.1: Reconstrucción usando SR

El objetivo último de la SR consiste en regenerar el escenario continuo del mundo real tomando como base una serie de imágenes que contienen fracciones de información de la captura original. En cambio, lo que se logra actualmente es recuperar una escena discretizada intermedia con mayor resolución [20, 21, 22].

En la Figura 2.2, se puede observar un ejemplo de SR, al tomar varias imágenes de una zona de estacionamiento y utilizando un algoritmo de SR, nos permitiría obtener una nueva imagen (de mayor resolución) en la cual se pueda distinguir el código escrito en dicha zona con claridad al aumentar la percepción de los detalles de dicho lugar.



Figura 2.2: Ejemplo de SR en zona de estacionamiento

La SR puede utilizar varios métodos, de entre los cuales podemos destacar:

**Interpolación:** usando las imágenes de LR se postula que es posible mejorar la resolución al interpolar la información entre píxeles de las imágenes iniciales y producir un conjunto de píxeles resultantes los que aportaran mayor resolución (al final se aplica un filtro para enfocar el producto final).

**Dominio Frecuencia:** este método está basado en el uso de información de las imágenes de LR en el dominio de la frecuencia, dado que se supone que las imágenes resultantes

estarán limitadas a una banda que pueden ser caracterizadas en el dominio de la frecuencia usando los desplazamientos y a que existe una relación entre la imagen de HR y la representación discreta de las imágenes de LR [23].

**Reconstrucción Regularizada:** en este caso se utiliza un modelo de la imagen a reconstruir y en el cual se define una función de coste a partir del error del modelo de generación y donde se realiza los cálculos de manera que se minimicen los parámetros de dicha función.

**Proyección sobre conjuntos convexos:** si suponemos que la solución al cálculo de la imagen de HR está acotado por un conjunto de datos convexo podemos tener un método que tomando las imágenes de LR se hace una proyección hacia un conjunto de soluciones (también convexas) y en forma iterativa se busca la imagen resultante.

**Proyección iterativa:** si utilizamos un modelo de generación de las imágenes de LR basado en una de HR y utilizamos un operador inverso se va calculando iterativamente una nueva estimación a partir del error existente entre la imagen de HR y la regenerada.

Estos métodos permiten agrupar los algoritmos según su kernel, los cuales utilizan sus propias técnicas que son aplicados dependiendo de sus características ya que cada uno de ellos tiene sus propias ventajas que son aprovechadas para resolver sus casos particulares.

Entre las técnicas actualmente disponibles tenemos [24, 25, 26]:

1. Sustitución de bandas de frecuencia-espacial.
2. Multiplexión de Bandas de frecuencia-espacial.
3. Parámetros múltiples usados con límites de difracción tradicional
4. Exploratorio de perturbación electromagnética de campo cercano.
5. Reducción de ruido de imagen de múltiple exposición
6. Limpieza (deblurring) con un marco simple
7. Localización de imagen con sub-pixel
8. Inducción bayesiana que va más allá del límite de difracción.

### 2.1.1. Algoritmos de SR usados actualmente

En astronomía se usan actualmente algunos métodos de SR, por ejemplo, reconstrucción lineal de pixel variable (o Drizzle). Este fue utilizado en el Hubble Deep Field de 1996, este hace uso del peso estadístico de los píxeles de diferentes figuras. Otro método empleado es denominado imágenes de suerte (Lucky imaging), la cual utiliza los cuadros tomados con un muy pequeño tiempo de exposición y de las figuras recogidas se escogen las que tengan una mejor calidad (que posean la mayor relación entre señal a ruido -SNR-, estas varían del 5%-20% del total) y luego se construye una grilla con todas ellas, de tal forma que se las promedia para obtener la imagen de alta resolución buscada [27].

En Machine Learning, se está usando entre otros un procedimiento llamado Aprendizaje Interno (Internal Learning) [18], el que hace uso del supuesto que muchas sub-regiones de la

figura de entrada son muy parecida a zonas de baja escala de la misma imagen. Cuando se tienen entradas de tipo disperso (Sparse Coding), la relación lineal entre señales puede ser reconstruida de las proyecciones de LR. Este método tiene buenas perspectivas en escenarios de múltiples vistas (como en espectáculos que tengan varias cámaras apuntando a un solo objetivo) [28, 29].

Un último ejemplo de los métodos actualmente utilizados, podemos mencionar el uso de la técnica de Fuera de Foco (Defocus Cue), la cual usa como modelo generador un Campo Aleatorio de Markov (MRF) para proporcionar un prior (o parámetro de regularización) y un método de máxima estimación a posteriori produciendo el campo de alta resolución [19].

### 2.1.2. Algoritmos más promisorios de SR

Los mejores resultados obtenidos (cuando no está incorporado algún mecanismo de aprendizaje) provienen de la reconstrucción regularizada mediante procesos estocásticos, a pesar que el principal problema existente es la determinación de los parámetros de los modelos de generación (del movimiento, relación entre imágenes, y otros) [30].

En general en SR se separa el registro de las imágenes de entrada de la reconstrucción en alta resolución, hay algunos nuevos métodos que tratan ambas etapas en forma iterativa que parecen buenos enfoques para su desarrollo a futuro. Actualmente la determinación de buenos parámetros de regularización o priores es un tema abierto que al parecer podrían ser ajustados y obtener novedosos estimadores. Por el momento que parámetros utilizar depende del tipo de imagen de LR utilizado y de los resultados que se quiere obtener [31, 32].

Al considerar la utilización de Redes Neuronales Convolucionales (CNN) se debe tener en cuenta que estas tienen una etapa de aprendizaje. En el transcurso del año 2017 se ha incrementado el uso de algoritmos de SR que usan *Deep Learning* (digamos CNN en general). En el entrenamiento de una red neuronal se ajustan los valores de sus pesos al exponer su entrada a una imagen que es parte del conjunto de aprendizaje, así mismo se tiene que tener suficientes imágenes para ajustar sus pesos adecuadamente, considerando el mayor número de casos posibles y también de los no-ejemplos, así se calculan los parámetros internos de la CNN (pesos) y le permiten que al ejecutarse la red pueda (en modo de operación) responder con una imagen de salida producto de las inferencias de su estructura (mejorando la resolución de las imágenes de entrada) [33, 34, 35].

## Generative Adversarial Network (GAN) y Súper Resolución

El aprendizaje en una CNN implica un mayor procesamiento que en los algoritmos sin aprendizaje. En el caso de *Deep Learning* que tenga como objetivo SR, se produce una generalización de los patrones observados en el conjunto de aprendizaje, lo que algunas veces conduce a que la salida de la red resulte inexacta [36, 37], con lo que se consiguen mejores resoluciones, pero no en todos los casos.

Este problema parcialmente se puede disminuir con las Generative Adversarial Networks (GAN) [38], las cuales generan imágenes/valores cercanos al límite entre las imágenes de aprendizaje que ayudan a separar el espacio de las posibilidades aceptables de las erróneas (al tener un conjunto de entrenamiento en el cual se tiene muy refinado el límite entre valores buscados y no deseados se logra reducir los falsos positivos) [39, 30]. Así como en el campo de visión de robot se utilizan muchas variantes de CNN, las cuales tienen cada una un uso específico [40], en la SR tenemos varios modelos CNN que mejoran situaciones particulares de imágenes, por ejemplo [41] que mejora imágenes de infrarrojo.

## 2.2. *Cubesat* (nano Satélite estándar)

Para que un satélite funcione adecuadamente se necesita la integración de diversas tecnologías (y un *Cubesat* no es una excepción). En el diseño es necesario establecer el (o los) circuito(s) que controlara(n) al dispositivo, la estructura que contendrá los componentes, la aislación que permitirá tener a una temperatura adecuada las distintas partes del satélite, la unidad de propulsión, la unidad de control de orientación y las varias cargas útiles que constituyen los mecanismos que realizarán las acciones para cumplir los objetivos de la misión.

Dada las peculiares condiciones donde trabajan los ingenios espaciales, la forma del satélite suele ser extraña al común de los mortales, por ejemplo, el primer satélite, el Sputnik, tenía un cuerpo esférico con 4 antenas que le permitían comunicarse, como se observa en la Figura 2.3.



Figura 2.3: Sputnik: Primer satélite artificial

Los primeros satélites fueron de diseño y desarrollo exclusivo, lo que implicaba que eran de un precio muy elevado. Con el paso de los años se desarrollaron plataformas previamente probadas y que permitían construir nuevos satélites a un menor costo, pero aun con estas mejoras el valor siempre se ha mantenido excesivo, es por ello que nació la iniciativa de los *Cubesat*.

Para producir un bien a bajo coste una opción es estandarizar los componentes y permitir que varios fabricantes puedan construir partes que al ser ensambladas con otras que cumplan un estándar. Así apareció el estándar de los *Cubesat* al normalizar estructura, partes y conexiones del nano satélite.

Como se resumen en la Figura 2.4, los *Cubesat* son pequeños satélites (peso de algunos kg, pocos componentes y de un tamaño de decenas de cm) que inicialmente trabajan en órbita baja (entre 200 y 1200 km de altitud) y tienen partes estándares lo cual ha permitido a instituciones con bajo presupuesto o que no tuviesen previamente experiencia con tecnología espacial acceder a desarrollar sus propios ensayos.

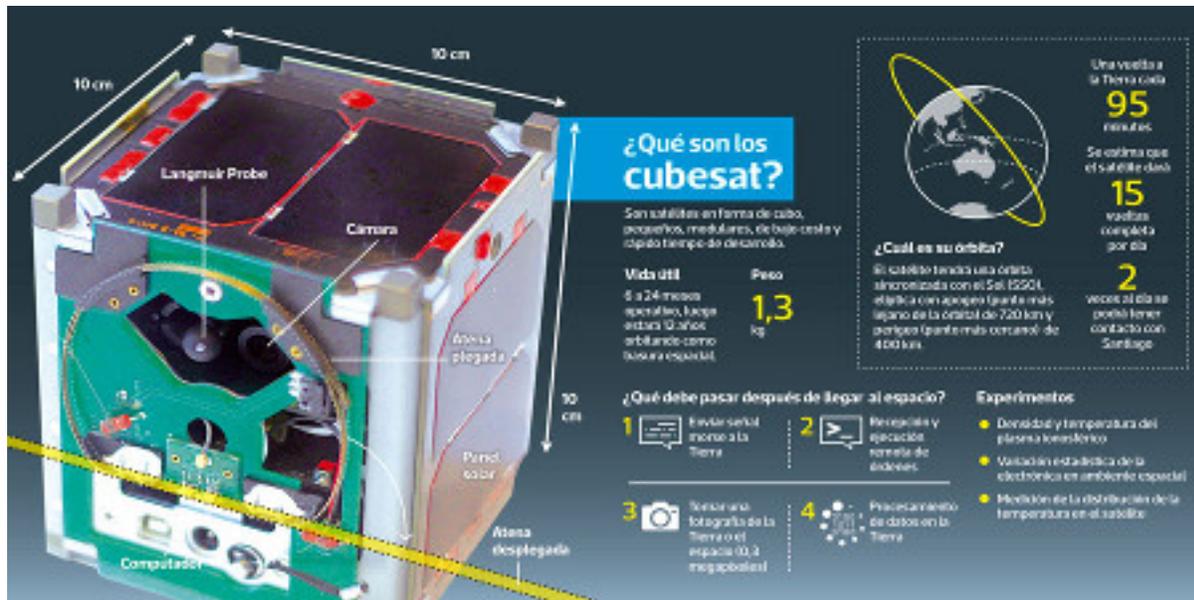


Figura 2.4: *Cubesat*: Características

Así, por ejemplo, el Cubesat SUCHAI 1 tiene una configuración llamada 1U, la cual consiste en un cubo de 10 cm de lado (SUCHAI 1 fue lanzado el 23 de junio de 2017, durante el desarrollo de esta memoria). Este tiene 4 subsistemas principales:

- Computador controlador
- Comunicaciones (y antena)
- Sistema de Energía
- Cargas útiles

Como ejemplo de cargas útiles de *Cubesat* podemos mencionar que los instrumentos de SUCHAI 1 son (algunas se pueden ver en Figura 2.5):

**Sonda Langmuir:** estudio del potencial eléctrico del plasma ionosférico

**Electrónica fuera de equilibrio:** estudia fluctuaciones en circuito RC

**Cámara:** pequeña cámara digital para estudiar la captura de imágenes desde el espacio

**GPS:** receptor GPS para determinar la posición del *Cubesat*

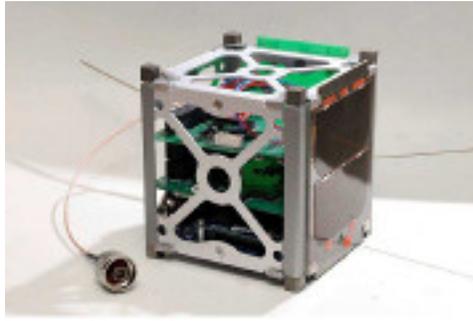


Figura 2.5: Suchai 1: Maqueta

Un mecanismo bastante utilizado para determinar la orientación de un satélite es un rastreador de estrellas (Star Tracker), el cual consiste en una pequeña cámara que toma imágenes del cielo y compara dicha imagen con un catálogo previamente definido y como resultado se logra determinar cuál es la orientación del satélite dada por la imagen captada por la pequeña cámara. Este mecanismo es de relativo bajo costo su implementación y entrega una buena estimación de la orientación del satélite.

Un uso muy utilizado de cámaras en ingenios espaciales es el estudio de los recursos naturales de la tierra (o percepción remota) y esto se consigue apuntando una cámara digital a la superficie terrestre y obteniendo imágenes con una resolución dada por la óptica y las características del dispositivo sensor (CCD o CMOS).

En los mencionados usos de las cámaras si se quisiera disponer de mayor captura de luz de las imágenes obtenidas se necesitaría aumentar el tamaño de la óptica utilizada (con un consiguiente aumento del volumen y peso del ingenio, lo que implica un mayor costo del satélite).

## 2.3. Percepción Remota

El mapeo topográfico ha evolucionado desde las tomas aéreas hasta las actuales imágenes tomadas por satélites [42]. Permitiendo en la actualidad registrar desde la órbita terrestre las características del terreno, captando en una sola imagen amplias zonas (aproximadamente varios kilómetros cuadrados) con una resolución del orden de medio metro.

La historia de la percepción remota desde satélites empieza por el año 1972 con el proyecto ERTS-1 (posteriormente llamado Landsat). En la Figura 2.6, se observan los principales satélites de observación terrestre y sus tiempos de vida.

En el satélite de observación terrestre Spot-5 (2002) se utilizaba el principio de la SR para mejorar su resolución al utilizar 2 cámaras de manera de captar simultáneamente imágenes de la superficie que están desplazados en medio píxel tanto en sentido horizontal como vertical (con respecto a la línea de su movimiento proyectada en la superficie) y a este proceso se le llama *Supermode* [5].

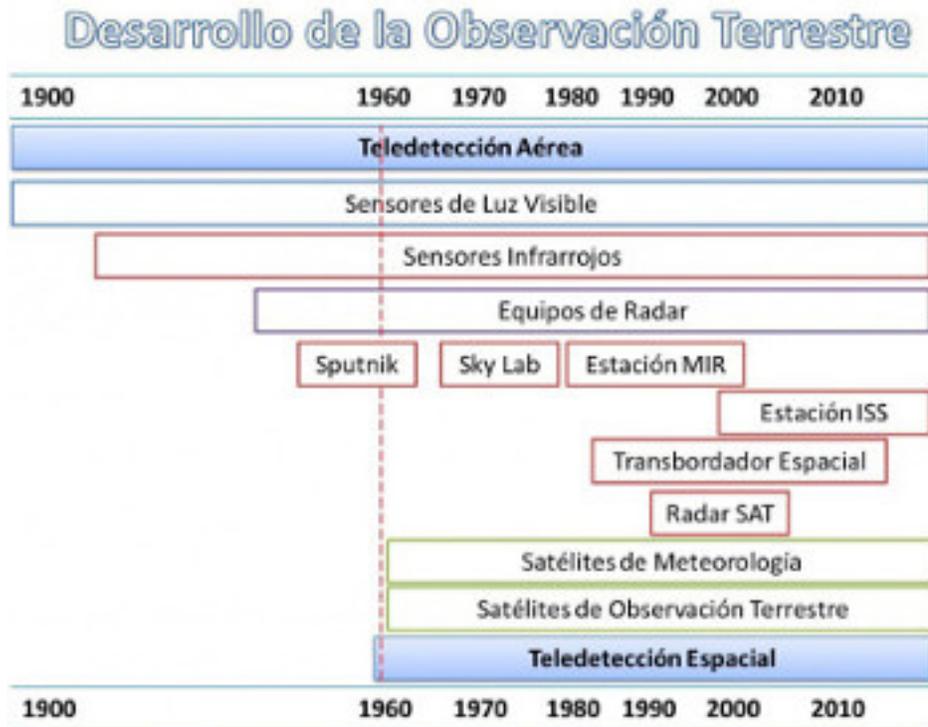


Figura 2.6: Línea de tiempo del desarrollo de la observación terrestre

Desde el año 2000 se ha venido desarrollando pequeños satélites para la captura de datos de la superficie terrestre [43]. Los sensores y mecanismos que se han ido utilizando han llegado a un nivel que hacen competitivo a estos satélites [44, 45, 46], produciendo similares resoluciones por una fracción del coste de misiones más onerosas (por supuesto con limitaciones, como menor ancho de banda de bajada, vida útil o capacidad de memoria, etc.).

### 2.3.1. Resolución de imágenes tomadas desde satélites

Existen 4 tipos de resoluciones que se utilizan en el trabajo de la percepción remota:

**Resolución Espacial:** tamaño mínimo detectable de la superficie en cada pixel.

**Resolución Espectral:** cantidad y características de las longitudes de onda observadas en los sensores.

**Resolución Radiométrica:** número de intervalos de intensidad observados en los sensores.

**Resolución Temporal:** el tiempo que transcurre entre la toma de 2 imágenes.

La resolución espacial depende de la distancia a la cual se encuentra la cámara del objeto a fotografiar (en nuestro caso la altura de paso del satélite) y del campo de visión (FOV) [47], descrito por la Ec. 2.1 y en la Figura 2.7 se muestra la relación espacial entre las variables.

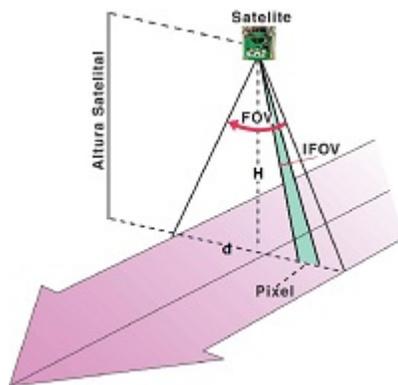


Figura 2.7: Relación FOV con altura y ancho de zona observada

$$d = 2H \tan\left(\frac{FOV}{2}\right) \quad (2.1)$$

donde FOV es el campo de visión, d es el ancho de faja espacial y H la distancia del sensor a la superficie terrestre.

Para darse una idea de los valores involucrados podemos mencionar que el satélite Meteosat y los satélites geoestacionarios ofrecen una resolución espacial de 5000 m en sus imágenes de la cobertura global de la superficie terrestre, los satélites de la serie NOAA-AVHRR tienen resoluciones que van entre 500 y 1100 m, Landsat-TM tiene 30 m y el sensor QuickBird 0.7 m. Actualmente (diciembre 2017) dentro de los satélites civiles de observación terrestre WorldView-4 tiene la mejor resolución espacial con 30 cm.

Generalmente, a mayor resolución espacial disminuye la temporal y también la espectral. El problema es que cada aumento de resolución multiplica el volumen de datos que el sensor entrega para preprocesar y enviar a las estaciones de Tierra. Así mismo la aparición de sensores de orientación modificable ha permitido compatibilizar alta resolución espacial y elevada resolución temporal a costa de una toma de imágenes no sistemática (o sea tenemos un aumento de procesamiento electrónico).

En el diseño de los sensores de un satélite se debe tener en cuenta todas las consideraciones mencionadas hasta ahora. Salvo excepciones, se va a trabajar con ventanas atmosféricas y se va a seleccionar aquella combinación de regiones del espectro que mayor información van a proporcionar acerca de los fenómenos que se quieren estudiar. Estas mismas consideraciones se deben tener en cuenta cuando se exploran otros planetas del sistema solar [48].

Cuando se disponía solo de fotografías aéreas la zona cubierta en el sobrevuelo eran de una fracción de la zona que actualmente cubren los satélites de observación. En contraposición la resolución de las fotos aéreas resulta ser mayor (en parte porque la altura en cada pasada es menor y es más calibrada).

Junto a las imágenes captadas también se utiliza el RADAR para obtener información de la superficie, el cual tiene la ventaja que puede obtener datos independientes de si está nublado o si es de día o noche [49]. Dependiendo de la frecuencia utilizada el RADAR puede

penetrar la superficie algunos metros.

Además de lo anterior se han combinado diversos métodos en el procesamiento de imágenes, así por ejemplo, en capturas aéreas se debe compensar el movimiento, brillo, y otros factores, a los cuales se incluye la SR junto a otras técnicas para mejorar la imagen final [50, 51].

De las plataformas que se han desarrollado para la ejecución de algoritmos que procesen en tiempo real imágenes, que provengan de la percepción remota se puede mencionar a [52], donde podemos observar que ya se ha diseñado el hardware para estos casos, y solo nos resta determinar una plataforma que tenga una relación coste/rendimiento de acuerdo a las restricciones que impongan los objetivos de diseño.

Con las nuevas tecnologías se ha abierto un gran ámbito de aplicación en disciplinas que anteriormente no contaban con estas herramientas, así pues ahora las modernas capacidades de la percepción remota pueden ser utilizadas en especialidades como la arqueología [53] y otras afines.

### 2.3.2. Distancia focal

La distancia focal de un dispositivo es la longitud entre el centro óptico de la lente (o espejo o diafragma, en el caso de una cámara) y el foco (o el lugar donde se concentran los rayos de luz).

Al aumentar la distancia focal se va disminuyendo el área de observación (o ángulo de visión, el zoom es justamente un ingenio que varía la distancia focal) de nuestro dispositivo óptico, tal como se puede apreciar en la Figura 2.8. Al utilizar el zoom se debe enfocar nuevamente la máquina para que la luz llegue al nuevo foco (no así en el zoom digital, el cual solo recorta la imagen y no cambia verdaderamente la distancia focal).



Figura 2.8: Variación de la distancia focal

### 2.3.3. Razón Focal

La razón focal (o número  $f$  es  $F/D$ , es la distancia focal dividido por el diámetro de la entrada de luz -comúnmente una lente-, que es redireccionada al objetivo-) es el índice de cuan luminoso es un telescopio.

Esta característica es aplicable en astrofotografía y no en la observación visual.

Muchas veces es llamada la "velocidad" del telescopio: se dice que un telescopio es rápido cuando su razón focal es baja (no tiene relación con las características mecánicas del mismo, sino con la rapidez de recolección de luz). Esto es especialmente conveniente en la astrofotografía, donde se pueden reducir sustancialmente los tiempos de exposición si se utilizan sistemas de F/D bajos.

## 2.4. Herramienta de cuantificación: PSNR

Para comparar la calidad de una señal después que se le ha aplicado compresión/pasado por un canal ruidoso/cambiado de formato/etc. se utiliza la herramienta llamada; Peak Signal to Noise Ratio (PSNR). La cual consiste en una relación entre el error de la imagen/señal original y la resultante de un proceso, y el valor máximo de los valores posibles de la entrada.

Si tenemos una imagen de tamaño  $m \times n$  podemos definir el error cuadrático medio (MSE), tal como se expresa en la Ec. 2.2, y la PSNR, utilizando escala log, será la expresada a continuación.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (2.2)$$

$$PSNR = 20 \log_{10}(MAX_I) - 10 \log_{10}(MSE) \quad [dB]$$

# Capítulo 3

## Algoritmos de Súper Resolución Implementados

Como los algoritmos de SR tienen una gran diversidad (que van desde interpolación, basado en el dominio de la frecuencia, reconstrucción regularizada, uso de machine learning y ahora ultimo *Deep Learning*) se aplicará a priori una selección que permita disponer de posibles combinaciones de algunas características de estos algoritmos de manera de poder elegir al final al que nos entregue una mejor combinación de sus prestaciones (dado nuestro objetivo de obtener un algoritmo a utilizar en una plataforma de recursos limitados como un *Cubesat*, hay tener en cuenta que no basta considerar elegido al que tenga mejor tiempo de respuesta, o asociado al menor uso de memoria, o el mínimo consumo de potencia, ya que probablemente deberemos analizar los valores obtenidos y evaluar las consecuencias en el proyecto).

Aunque el trabajo seminal de SR data de 1984 [54], los papers que condujeron a los algoritmos que trabajaremos son de los 80 del siglo XX y el último ya es del 2004 [55]. Empezaremos considerando un algoritmo iterativo que sea ampliamente descrito en la literatura y que tenga versiones para monoframe y para multiframe, así tendremos la opción de considerar otras variables si el análisis final nos aconseja estudiar implementaciones más simples. Este algoritmo dentro de un ciclo calcula una medida de error que en parte controla la evolución del ciclo y tiene un número de ciclos máximo que determina cuando detenerse al alcanzar una cota fijada de antemano. La medida de error es comparada en cada ciclo y si baja de un valor predeterminado se termina la secuencia de ciclos.

A continuación, vamos a considerar un algoritmo que sea más complejo y que haga uso de restricciones convexas que van acotando el espacio de evolución de la imagen a generar. También es un método en parte iterativo (que además fue propuesto años antes que el algoritmo IBP, pero que con nuevas ideas resulta obtener mayor resolución final).

Como tercer algoritmo, apuntaremos a uno que teóricamente sea el que entregue la mayor resolución -cada uno de estos es una implementación particular de los métodos indicados y no se excluye que estas puedan ser refinadas y los valores obtenidos de sus resoluciones podrían cambiar cuando se implementen modificaciones a estas ideas-, este esta explícitamente

desarrollado pensando en imágenes de satélites de percepción remota y dice entregar buenos resultados para una plataforma satelital de varias cámaras.

Existen implementaciones, por ejemplo, para cámaras de vídeo en el infrarrojo [56] que muestran que la SR también puede ser aplicada en multimedia. Se recomienda tener en cuenta que los algoritmos implementados solo son un caso particular de un amplio abanico de posibilidades en la lógica a utilizar y en la aplicación de esta tecnología.

La formulación más general de SR se puede describir de la siguiente manera:

$$\vec{Y}_k = D_k B_k M_k \vec{X} - \vec{E}_k ; \quad k = 1, \dots, N \quad (3.1)$$

Donde:

$\vec{X}$  es la imagen de alta resolución

$\vec{Y}_k$  es la imagen k-esima de entrada

$D_k$  es la matriz de downsampling

$B_k$  es la matriz de borrosidad

$M_k$  es la matriz de movimiento

$\vec{E}_k$  es el vector de error

Los algoritmos multiframe los podemos describir como aquellos que reciben una o varias imágenes LR, y entregan una imagen de HR  $\hat{x}$

$$\hat{x} = f(\vec{Y}_k) ; \quad k = 1, \dots, N \quad (3.2)$$

donde como caso particular los algoritmos POCS, se puede describir como:

$$x^{n+1} = P_m P_{m-1} \dots P_2 P_1 x^0 ; \quad k = 1, \dots, N \quad (3.3)$$

en el cual  $x^0$  es la imagen de inicio, y los  $P_i$  es el operador de proyección

Los algoritmos a implementar serán los siguientes:

### 3.1. Iterative Back-Projection -IBP-

Iterative Back-Projection (IBP) es un método de SR enfocado en remover factores de ruido y borrosidad (blur). *IBP* empieza estimando la imagen final y en forma iterativa refina el resultado (la suma del gradiente de error es calculada y utilizada en obtener la siguiente iteración de la imagen) [57]. El método fue inventado por Irani y Peleg basado en *Back Projection* para ser usado en tomografía [58].

*IBP* es un método ya muy conocido, es fácil de entender y utilizado en el arsenal de los algoritmos de SR. Su ventaja es su relativa sencillez lógica. A su vez su desventaja es que no tiene solución única debido a la forma como se resuelve la retro propagación (y a la imagen de inicio elegida), también es difícil encontrar limitadores a priori adecuados para esta técnica.

Este método se ha utilizado para procesar video, al aplicarle filtros de punto crítico [59] obteniendo un rendimiento superior al usar una interpolación bilineal y bicúbica.

IBP puede ser ocupado tanto en monoframe [60] como en multiframe, lo que lo convierte en un método bastante flexible.

Para IBP se trabajará en un ciclo de un tamaño máximo previamente determinado o haber llegado a un mínimo de error (también previamente determinado).

De esta manera, de los trabajos de [61] y [62], tenemos:

$$\begin{aligned} res &= lhs' * (rhs - lhs * x^n); \\ x^{n+1} &= x^n + \lambda res ; n = 1, \dots, M \end{aligned} \tag{3.4}$$

en que  $\lambda$  es un parámetro que sirve para ajustar el gradiente descendente para hacerlo funcionar para más de 8 imágenes y  $M$  es el Máximo de Iteraciones o llega a error mínimo, lhs y rhs son las matrices de la linealización del lado izquierdo (lhs) y derecho (rhs) de la Ec. 3.1. Se requiere en este caso una imagen semilla (o inicial  $x^0$ ) que se obtiene haciendo una interpolación y promediando los nuevos valores.

Como se muestra en la Figura 3.1, la implementación general de este algoritmo hace una interpolación de la imagen minimizando el error del proceso en los ciclos de su iteración.

Como base del código utilizado se utilizó [63], el cual se adaptó para poder recibir más de 8 imágenes de entrada y poder ser parte de la interfaz (esta se describe en el capítulo 4).

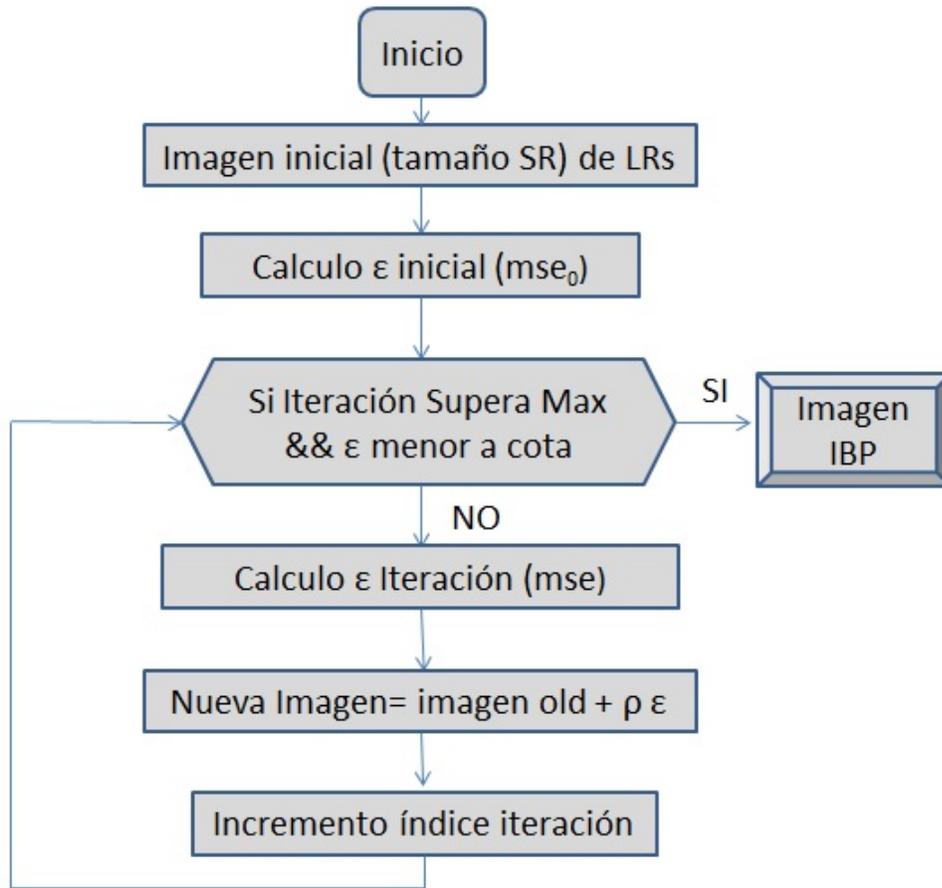


Figura 3.1: Implementación IBP

### 3.2. Projection Onto Convex Sets -POCS-

En matemáticas POCS algunas veces llamado método de proyección alterna es una forma de encontrar un punto dentro de la intersección de varios conjuntos convexos cerrados (una variante se le conoce como algoritmo de Dykstra).

En este método la imagen final se encontrará en la intersección de un conjunto de limitaciones típicamente convexas, que representan características de la figura de SR deseada tales como suavidad, fidelidad del resultado obtenido, energía del enlace, etc. (también se han usado métodos de wavelet, blur y movimiento relativo para caracterizar los parámetros a considerar) [64].

POCS al utilizar una imagen semilla, sus resultados dependen de dicho cuadro inicial (initial guess), tienen lenta convergencia y alto costo computacional.

Dado que este algoritmo tiene una aproximación normalizada es más fácil obtener los limitadores iniciales que en caso del método IBP y tiene mejor desempeño (bajo determinadas características su resolución lograda es mayor). Fue inicialmente presentado por Stark y Oskoui (1989) [65] y un diagrama general de cómo se estructura está indicado en la Figura 3.2.

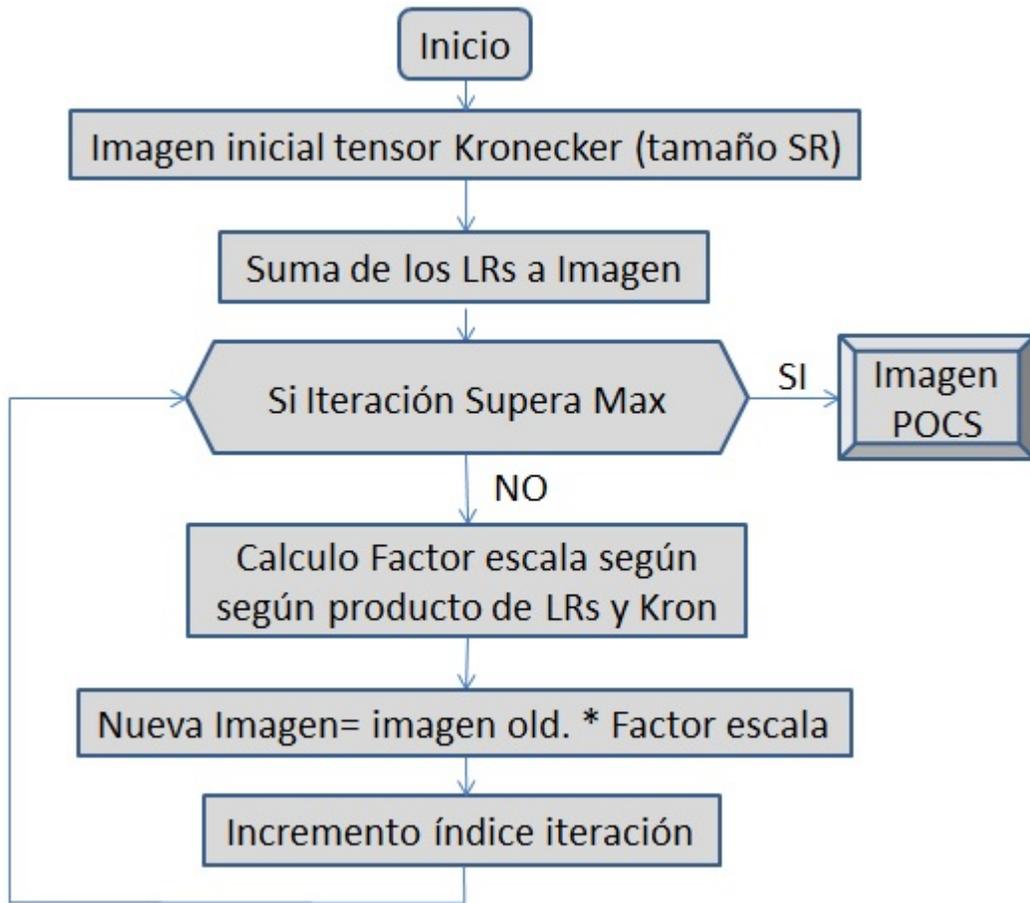


Figura 3.2: Implementación de POCS

Una variante postula utilizar restricciones en la consistencia de los datos, umbrales de corrección variables y agregar la habilidad humana de discriminar cambios en brillo [41], lo cual podría incrementar significativamente la capacidad de aumentar la resolución en dicho algoritmo.

Tomando la Ec. 3.3, postulamos que una aceptable aproximación es  $p(X) = \|X\|^2$  y es lo que permite hacer rápido el método ya que solo contiene multiplicar algunas matrices y no contiene iteración en él.

Para la implementación se tomó [66], transformando sus entradas de un solo conjunto de imágenes de manera que pudiera aceptar un objeto cell, que contuviera las imágenes de entrada y que pudiera interactuar con el resto de la interfaz creada para realizar las simulaciones.

### 3.3. ROBUST

Este algoritmo fue expresamente desarrollado para mejorar las prestaciones de observaciones desde satélites y puede resultar una buena opción una vez que ejecutemos los estudios comparativos correspondientes [67, 68].

En esencia este procedimiento aplica parámetros de borrosidad (*blurring*), desvanecimiento y de movimiento diferencial, además de considerar un error gaussiano dentro de una iteración en la cual utiliza el método del gradiente para ir minimizando el error entre iteraciones y aproxima los cuadros utilizados por el valor de la mediana de los píxeles, como se aprecia en la Figura 3.3.

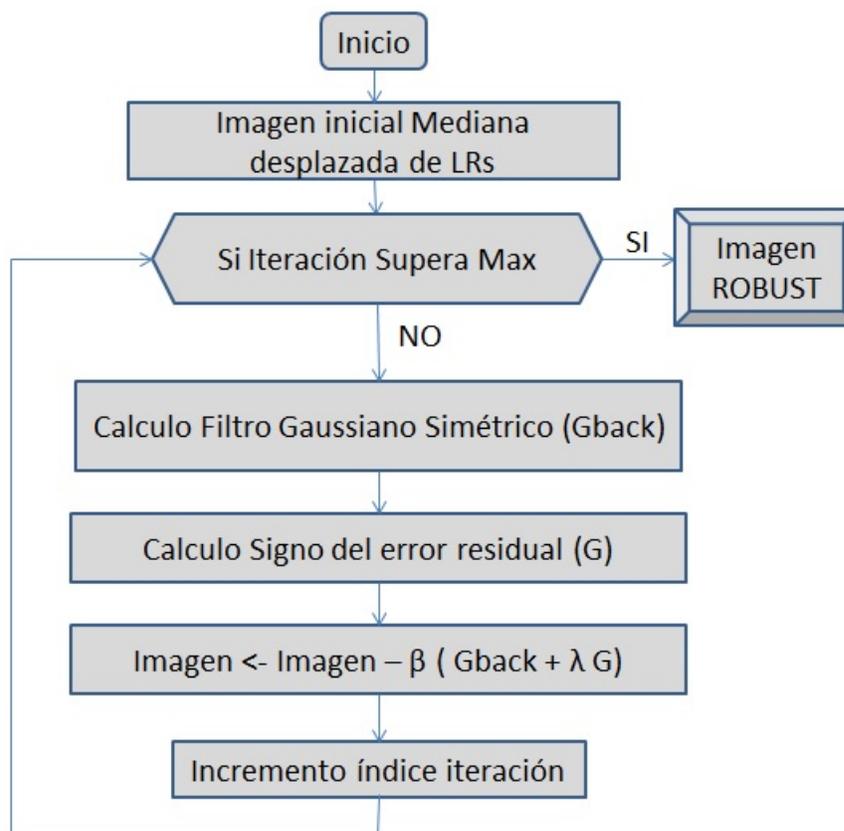


Figura 3.3: Implementación ROBUST

#### 3.3.1. Porque elegir la mediana de los píxeles

Una aproximación para producir un algoritmo de SR robusto es minimizar el error en cada norma diferente de modo que se puedan manejar los valores atípicos (outlier) mejor que en el caso más básico. Al elegir la mediana estamos limitando el espacio de las soluciones implementadas a los valores disponibles de nuestro conjunto de entrada.

### 3.3.2. Procedimiento de Detección de Sesgo

La meta del procedimiento de detección de sesgo es detectar muchas de las mediciones de valores atípicos (*outlier*) los cuales pueden implicar un error sistemático en el cálculo del estimador de la mediana [67].

Una mejora del método *Robust* [55] es que además de aumentar la resolución resultante también disminuye el tiempo de procesamiento al mejorar la convergencia dado su mejor estimación de candidatos.

si la ecuación que queremos resolver es:

$$\hat{X} = \underset{X}{\operatorname{argmín}} \left[ \sum_{k=1}^N \|D \cdot H \cdot F_k \cdot X - Y_k\|_1 + \lambda \sum_{l=-P}^P \sum_{-P}^P \alpha^{|m|+|l|} \|X - S_x^l S_y^m X\|_1 \right] \quad (3.5)$$

y esta ecuación se resuelve usando iteraciones en pasos descendientes:

$$\hat{X}_{n+1} = X_n - \beta \left[ \sum_{k=1}^N F_k^T \cdot H^T \cdot D^T \cdot \operatorname{sign}(D \cdot H \cdot F_k \cdot X_n - Y_k) + \lambda \sum_{l=-P}^P \sum_{-P}^P \alpha^{|m|+|l|} [I - S_x^{-m} S_y^{-l}] \operatorname{sign}(X_n - S_x^l S_y^m X_n) \right] \quad (3.6)$$

Para su implementación se tomó inicialmente el código de [69], del cual también se aprovechó inicialmente su interfaz, las cuales fueron modificadas para poder comparar simultáneamente varios algoritmos SR, y poder realizar varias pruebas cambiando sus parámetros (lo cual es una generalización del trabajo original del Sr. Hanson, el cual lo realizo como parte de un curso de sus estudios de magíster, en la Universidad de California, en Santa Cruz).

### 3.4. SR y Deep Learning (DL)

En los últimos años (especialmente en el 2017) se ha incrementado el uso de *DL* en el desarrollo de nuevos algoritmos de SR. Los cuales se han mostrado bastante buenos (obtienen mayor resolución final que los modelos anteriormente utilizados). El problema con este método es que exige un proceso de aprendizaje a fin de obtener los parámetros adecuados para su funcionamiento (en el caso de AlexNet requiere millones de ejemplos con la finalidad de funcionar adecuadamente), al realizar esto se va extrapolando a partir del conjunto de aprendizaje cuales podrían ser los patrones más probables de información a ser incorporada (algo que no se puede prometer que será real en todos los eventos). Esto acota el uso de esta forma de producir SR a las situaciones cuando se tienen muchos datos previamente y se dispone de suficiente poder de cálculo (y tiempo para procesar) [70, 38].

Como se observa en la Figura 3.4, *DL* implementa una red neuronal que procesa la imagen de entrada, toma cada uno de los píxeles y aplicando mecanismos de convolución (un filtro) entrega un producto que es procesado por las restantes etapas de la red convolucional, en las cuales se realiza una inferencia controlada por los pesos de las redes neuronales que han sido previamente definidos por la etapa de aprendizaje [71, 72].

Las mejoras logradas en los últimos años han ido reduciendo el tiempo de entrenamiento, junto con aumentar la exactitud de los resultados al encontrar métodos que utilizan información contextual que exploran amplias regiones de las imágenes en forma eficiente [37, 73].

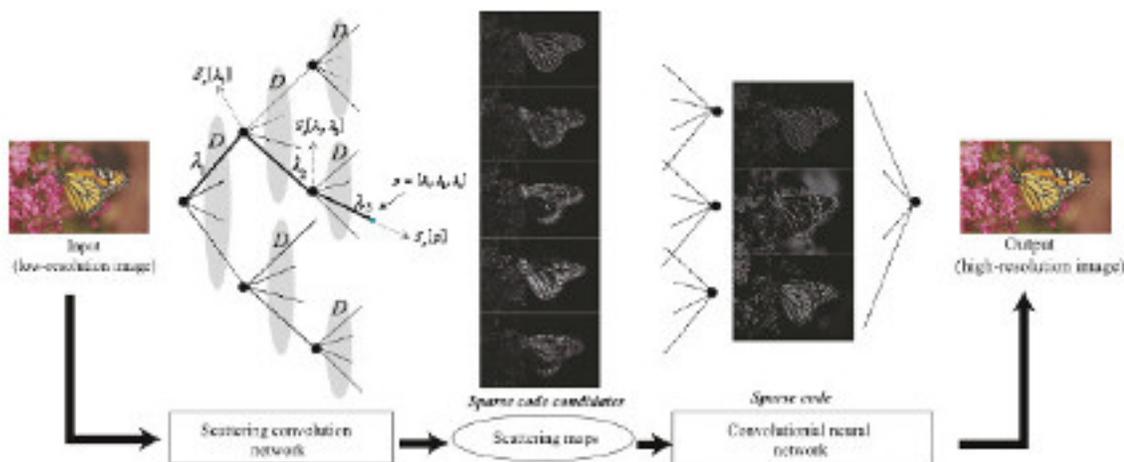


Figura 3.4: Deep Learning mejorando resolución

Los métodos *DL* son parte de las técnicas de redes neuronales, que a su vez es un subconjunto del *Machine Learning*<sup>1</sup> [74, 75], que a su vez es una parte de la Inteligencia Artificial (IA)<sup>2</sup>.

<sup>1</sup>Machine Learning es el uso de algoritmos para analizar datos, aprender de ellos para realizar una determinación o predicción sobre algo. La machine es entrenada con una gran cantidad de datos que le dan la habilidad para aprender cómo manejar una tarea

<sup>2</sup>Inteligencia Artificial es el conjunto de algoritmos que pretenden imitar la forma como funciona el cerebro humano -en términos muy simplista-

Aunque como se ha dicho en el 2017 se ha incrementado significativamente el uso del *DL* en las técnicas de SR, usando *Machine Learning* ya se han intentado varias aproximaciones [76] que han sido la base para posteriores mejoras y muestran que se ha estado trabajando en el tema dentro de la *IA* por unos cuantos años.

### 3.5. SR, Modelos de Elevación Digital (DEM) y RGB-D

Una técnica ampliamente utilizada en los tiempos de la toma de imágenes por aviones era la medición de la altura sobre la cual se efectuaban los vuelos, de manera de estimar el tamaño de los objetos (al tener la distancia al objeto y las características de la óptica se pueden aproximar las dimensiones de las figuras observadas).

En el último tiempo se están haciendo mediciones de profundidad de las imágenes a través de la utilización de sensores adicionales como los *RGB-D* [77, 78, 79], estas técnicas permiten discriminar por áreas la altitud dentro de la imagen aportando más información que puede ser aprovechada por la SR [80, 81] (digamos como un complemento a la mejora de SR, al disponer de datos de elevación es posible determinar más detalles).

Por ejemplo *Kinect* utiliza sensores adicionales a los CMOS (antiguamente los CCD), como un emisor y receptor de infrarrojo para emitir una grilla e ir registrando para cada uno de sus elementos cual es la profundidad registrada y así tener un punto de comparación para determinar cuál es la altitud de sectores de la imagen [82, 83, 84], como se trata de mostrar en la Figura 3.5.

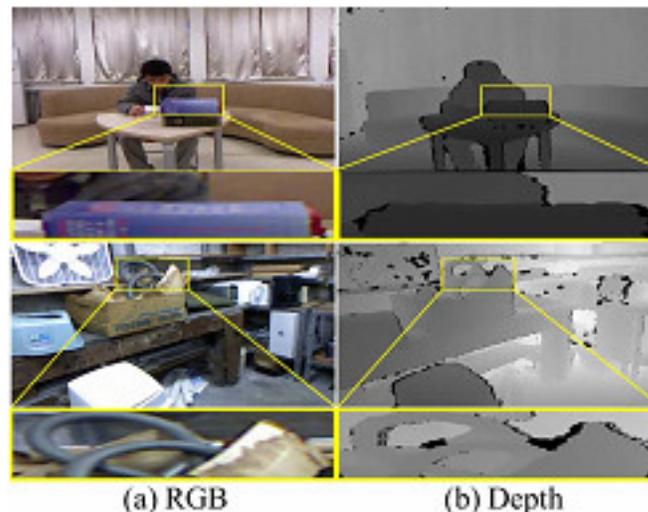


Figura 3.5: RGB-D: Determinación de profundidad en imagen a corta distancia

Para escalas mayores se utiliza el concepto de DEM, el cual utilizando Radars o imágenes estereoscópicas, LIDAR (como en Marte donde la misión *Mars Global Surveyor* usando Laser genero un mapa altimétrico de Marte con una cobertura mejor de la que se dispone para La Tierra) obtiene un mapa en que cada píxel de información tiene asociado su elevación -o altitud- [85].

De la misma manera que la SR multiframe utiliza varias imágenes, los modelos de elevación digital procesan dos (o más) imágenes de una misma zona y entregan mapas 3D, que aunque no tienen mayor resolución si dan otra perspectiva y ayudan entre otros al control de desastres naturales, estudios de perfiles de zonas montañosas, modelos de perfiles territoriales y muchas otras posibilidades.

# Capítulo 4

## Implementaciones y análisis de resultados obtenidos

### 4.1. Ejecución de las implementaciones

Los algoritmos de SR se implementaron en Matlab, ya que esta plataforma es muy versátil y permite tener rápidamente implementados modelos muy complejos, es de amplio uso y puede ejecutar los mismos algoritmos en varias plataformas. Por supuesto, siendo una ambiente en el cual se interpretan las instrucciones del lenguaje, lo que lo hace lentos en su ejecución. Esto no es un problema ya que al mostrar una comparación no estamos buscando medir los rendimientos con precisión, sino que estimamos las diferencias para escoger la opción que presenta las mejores características en el caso de utilizarse en otra instalación (por ejemplo, la que será utilizada en los *Cubesat*).

Dada las variaciones de tiempo obtenidos a los procesos en nuestras simulaciones se corrieron a lo menos 10 veces los procesamientos para calcular los valores promedio y sus desviaciones y así registrar el comportamiento de los algoritmos de manera de proceder a posteriores comparaciones.

La aplicación de los algoritmos de SR se ejecuta sobre una sola longitud de onda de la luz observada (o en su defecto se aplica a una transformación de una imagen color a una de escalas de grises). De esta manera en el caso de querer aplicar SR a imágenes color, se debe trabajar cada color (RGB) por separado y cuando se tengas los 3 resultados integrarlos en una nueva imagen SR color.

Como se observa en la Figura 4.1, al tener una imagen origen en color, SR se aplica por separado en los canales RGB (Rojo, Verde, Azul), obteniéndose 3 imágenes SR las cuales deben ser combinadas para producir la imagen color final.

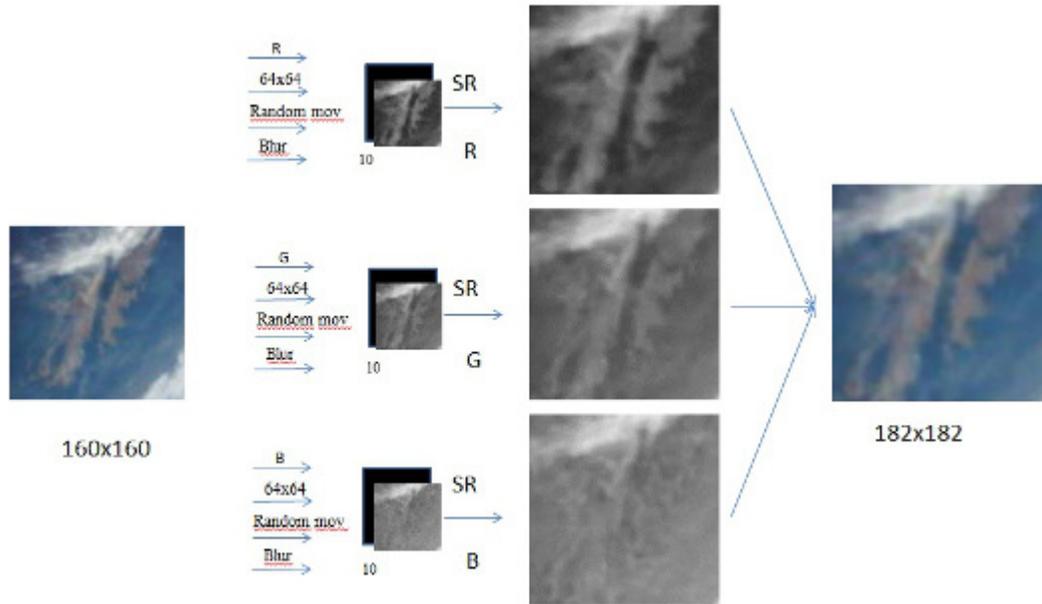


Figura 4.1: SR e imágenes a color

## 4.2. Interfaz

Para realizar comparaciones que puedan permitir un análisis visual (entre otros) se tomó de inicialmente el código de [69] y sobre él se implementó una interfaz generalizada de comparación utilizando la herramienta *Guide* de Matlab, en la cual se puede cargar un conjunto de imágenes (archivo \*.mat), calcular una matriz de los desplazamientos que tiene cada imagen con respecto a una base (por lo general la primera imagen). Como se observa en Figura 4.2.

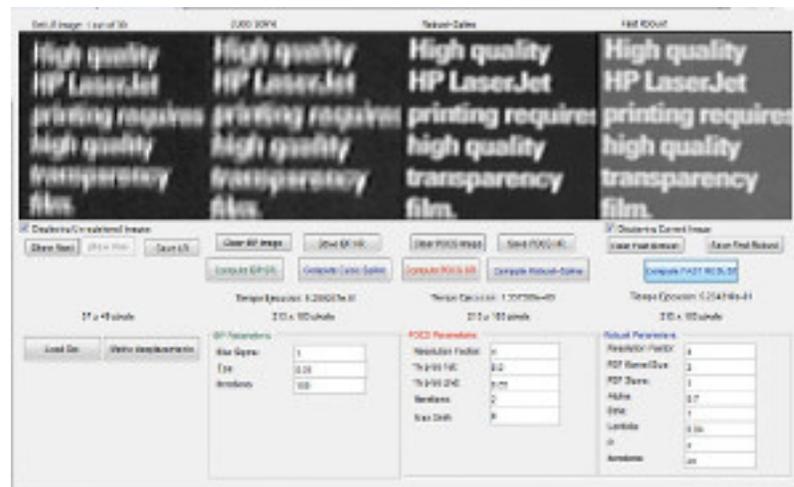


Figura 4.2: Interfaz despliegue de algoritmos de SR

En l interfaz se puede ejecutar los algoritmos IBP, POCS, y ROBUST, también se pueden mostrar los casos de interpolación bicubica y como resultaría ROBUST con una imagen semilla distinta. En cada caso se muestran los tiempos de ejecución y el tamaño en píxeles de cada imagen resultante.

Esta interfaz es suficientemente versátil ya que puede cargar conjuntos de imágenes sin límite a priori (los casos utilizados tenían entre 10 y 55 imágenes). Es posible cambiar los parámetros de cada algoritmo y luego ejecutarlos para estudiar las nuevas imágenes (variando como por ejemplo el factor de resolución, número de iteraciones, etc.).

En la Figura 4.3, se aprecia otro conjunto de imágenes de baja resolución y como se muestran cuando se van ejecutando al ir cambiando sus parámetros y corriéndolos nuevamente.

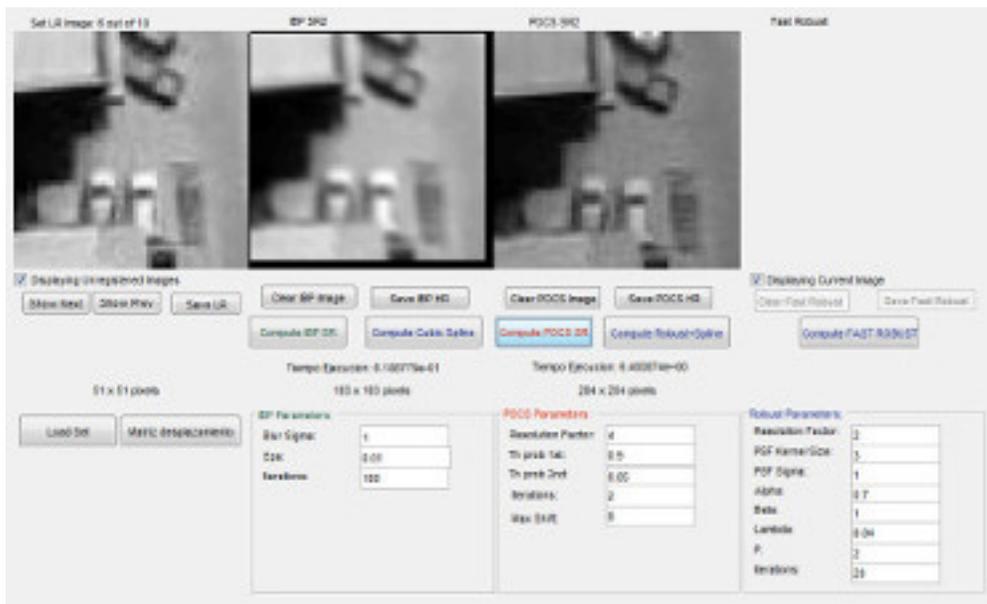


Figura 4.3: Interfaz variando conjunto de entrada y parámetros

Los conjuntos de imágenes tienen que estar en un formato `.mat` los cuales deben ser leídos como un arreglo de celdas (cell) en Matlab. Esta capacidad se aprovecha de la existencia de base de datos de imágenes estándares muy utilizados en pruebas<sup>1</sup>, las cuales permiten realizar evaluaciones de manera de poder entregar resultados comparables. Estas imágenes están constituidas por pequeñas variaciones de una imagen fuente, constituyendo una agrupación de similares características que permiten disponer de una secuencia de imágenes tomadas por una cámara (simulando un movimiento aleatorio con respecto al objeto en encuadre, en perpendicular, alejándose o acercándose levemente). Se asume que los desplazamientos son tales que las imágenes reflejan una variación subpixel en forma aleatoria de una imagen origen.

De los grupos posibles, después de realizar diversas ejecuciones utilizando la interfaz creada y variando algunos parámetros, como razón de resolución, grupo de imágenes, se eligió cuantificar los resultados para los casos de *grupo Eolian* (donde se muestra una zona accesible del Departamento de Ingeniería Eléctrica, Universidad de Chile (DIE), para comparación), *grupo Malvinas*, *grupo texto* y *grupo Emily*, ya que se podía evaluar mejor el comportamiento de los algoritmos al considerar cambios de contraste y tamaño de la imagen.

<sup>1</sup>Algunos de los grupos de imágenes fueron tomados del sitio MDSP Super-Resolution And Demosaicing Datasets, <https://users.soe.ucsc.edu/~milanfar/software/sr-datasets.html>

## 4.3. Pruebas

### 4.3.1. Grupo imágenes Eolian

Utilizando como base una imagen tomada de Google Map de la Facultad de Ciencias Físicas y Matemáticas (FCFM), específicamente el área sobre el DIE, Figura 4.4, podemos observar el contexto y elegir una parte de ella de fuente de las varias imágenes a utilizar y posteriormente ejecutar los algoritmos de SR.



Figura 4.4: Vista aérea del DIE y zoom de zona Eolian

Para realizar una comparación con una zona conocida se eligió una sub-área de  $128 \times 128$  píxeles, de la foto de la imagen aérea de la FCFM, la cual se puede apreciar en el recuadro ampliado de la Figura 4.4, esta fue elegida porque está en una zona iluminada y accesible del Departamento de Ingeniería Eléctrica de manera que podemos hacer un contraste entre las imágenes procesadas con los algoritmos de SR y mediciones en el terreno, para estimar las resoluciones obtenidas (por ejemplo, si en la imagen vemos una parte de un vehículo que esta muestreada a un píxel, luego vamos y medimos el objeto en cuestión y obtenemos por este medio la resolución en metros de cada píxel de nuestras imágenes procesadas).

Sobre esta imagen se procedió a generar 10 imágenes distintas de LR de  $51 \times 51$  píxeles<sup>2</sup> (con desplazamiento y borrosidad aleatorios) y luego usando estas como base se generó un objeto Eolian<sup>3</sup> (container que almacena 10 imágenes que contienen pequeñas variaciones de la original, siguiendo un patrón ya utilizado en dataset estándares ampliamente utilizados, por ejemplo, el MDSP ya mencionado), como se observa en la Figura 4.5. Es igual a disponer de un conjunto de 10 imágenes tomadas por una cámara en rápida sucesión, la cual ha efectuado pequeños movimientos en este tiempo.

Con los datos ya generados podemos ejecutar el algoritmo IBP iterativamente, y así ob-

<sup>2</sup>Se escogió trabajar con imágenes de pocos píxeles porque se descubrió que a medida que el tamaño de las imágenes aumentaba, el tiempo de ejecución se disparaba y los resultados de la comparación final no depende del tamaño de los ejemplos utilizados

<sup>3</sup>El nombre Eolian viene de que el área de estacionamiento observada es donde se guardan los vehículos eolian-vehículos solares diseñados y fabricados por estudiantes, funcionarios y académicos de la Universidad de Chile-

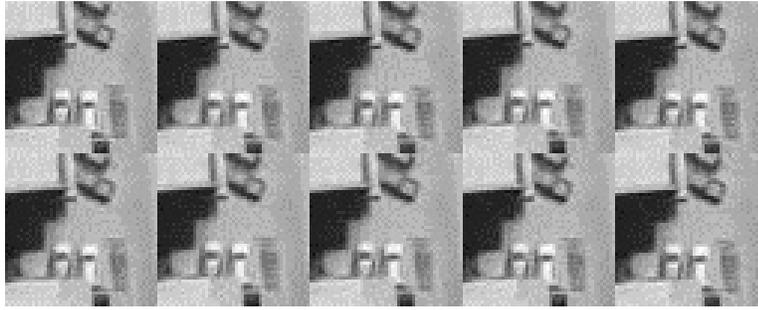


Figura 4.5: Generación de las imágenes como fuente de datos

tener valores representativos de la aplicación de este método (cada ejecución depende de la prioridad que le otorgue la CPU en ese momento, por lo que en cada corrida se obtendrán valores distintos). Se obtuvo el promedio de sus tiempos a partir de comandos de Matlab (*tic*, *toc*) y para calcular el uso de memoria se ocupó el comando que entrega el espacio de trabajo y sus tamaños (*whos*).

Esta imagen procesada se puede comparar con un algoritmo de interpolación bicubica. Se observó que el algoritmo IBP entrega mejores resultados, y estos además se pueden contrastar contra la imagen de HR que será nuestra imagen ideal, tal como se aprecia en la Figura 4.6.



Figura 4.6: Algoritmo IBP contra interpolación bicubica e imagen ideal

Para el algoritmo POCS se utilizó la misma imagen que para IBP, generándose un conjunto de varias decenas de imágenes desplazadas subpixel y así simular como si se hubiera estado en un vehículo en movimiento y con este como base ejecutar las iteraciones de POCS. Luego se ejecutó varias veces y se obtuvo un promedio de sus tiempos de ejecución y evaluando el uso de memoria usando el comando *whos* (igual que en IBP). Para poder apreciar la efectividad de este algoritmo ahora se comparó contra una interpolación bilineal y estos resultados se pueden observar en la Figura 4.7.

POCS es más complejo que IBP y presenta varias fases dentro de sus iteraciones. Por una parte, tiene un cálculo de valores promedio, que genera la imagen inicial sobre la cual se iterará en las otras etapas y posteriormente va comparando las imágenes generadas con las restricciones impuestas en el método. Ya que el cálculo de los datos iniciales toma un tiempo



(a) SR POCS



(b) SR ROBUST lento

Figura 4.7: Comparación entre POCS y ROBUST lento

constante no se consideró en el tiempo estimado en su ejecución y solo se utilizó las etapas de iteración en las cuales procede a calcular la imagen final.

El tiempo de ejecución y la complejidad de POCS son muy dependientes de la estructura y forma de cálculo del conjunto de las restricciones convexas que configuran el método.

En la implementación en curso solo se consideraron restricciones usuales. En la literatura se ha comentado que cuando se consideran las adecuadas (y numerosas) límites convexas la capacidad (digamos el aumento de la resolución) de este método aumenta significativamente. Por razones de tiempo no se alcanzó a implementar POCS más complejos y se dejara esta opción como trabajo futuro.

Un resumen de las pruebas efectuadas se muestra en la Tabla 4.1.

Para determinar cuál es la resolución obtenida de cada algoritmo se procedió a evaluar los datos variando la resolución y estimando en forma visual cual es la que entrega la mejor imagen. Así, en la tercera columna de la Tabla 4.1, se muestra el resultado de la ejecución del algoritmo ROBUST usando resoluciones 2,3,4,5,6 y se determinó visualmente que la mejor era cuando se tenía el valor 4 (asimismo se realizó con los otros algoritmos).

| Tipo       | Algoritmo | Resolución  | Tiempo Ejecución [seg] | Uso de Memoria [byte] |
|------------|-----------|-------------|------------------------|-----------------------|
| MultiFrame | IBP       | $\approx 2$ | $0.1873 \pm 0.0013$    | 7502056               |
|            | POCS      | $\approx 3$ | $4.1282 \pm 0.0333$    | 566752                |
|            | ROBUST    | $\approx 4$ | $0.2838 \pm 0.0021$    | 995856                |

Tabla 4.1: Resumen rendimientos algoritmos de SR para grupo imágenes Eolian (10 imágenes)

### 4.3.2. Grupo imágenes Malvinas

Aprovechando las imágenes capturadas por el satélite SUCHAI 1, se escogió una que fuera identificable con algún rasgo geográfico y así poder evaluar su resolución espacial con medios tradicionales (esto es calcular distancia en la superficie terrestre con Google Map). Esta imagen corresponde a las islas Malvinas las cuales tienen un ancho aproximado de 200 km. Dada la escala de la imagen se puede estimar su resolución entre 1 y 5 km, tal como se puede apreciar en la Figura 4.8.



Figura 4.8: Las islas Malvinas captadas por el satélite Suchai 1

Las pruebas en esta imagen nos acercan a un caso real de utilizar SR en imágenes captadas en un *Cubesat* a una altitud de 500 km. La mayor parte de las imágenes reales tomadas desde la órbita captan nubes, las cuales no permiten a priori determinar su resolución por comparación con estructuras terrestres, ocultando accidentes geográficos que están bajo ellas.

Para tratar las imágenes se escoge una sola banda (Rojo) lo que dificulta la interpretación (no se trataron las 3 bandas RGB para enfocarnos en demostrar el proceso, aunque es claro que se pueden procesar los colores RGB por separado y posteriormente integrarlos en una imagen color). En principio es difícil discriminar las mejoras con esta imagen tal como se observa en la Figura 4.9, pero es relevante observar que al menos se puede mejorar la resolución entre un factor 2 y 3.

El grupo de imágenes de LR tendrá unas dimensiones de 64x64 píxeles. Se escogió este tamaño acercándonos a los otros grupos de imágenes que se están comparando y nos permite un tiempo de procesamiento manejable de manera de ejecutar múltiples veces nuestros códigos y obtener estimadores de sus rendimientos con el error correspondiente.

Luego los resultados de las pruebas realizadas se muestran en la Tabla 4.2.

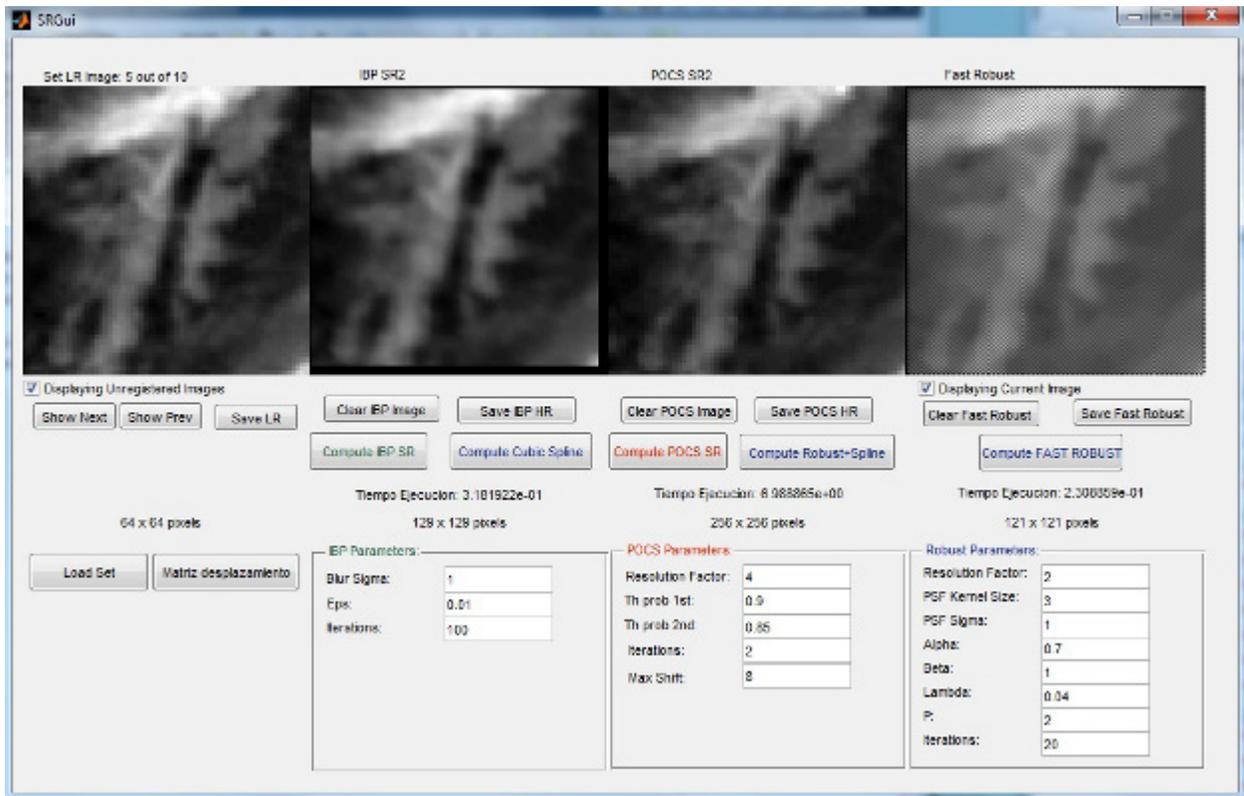


Figura 4.9: Comparación de imagen base Malvinas y sus SR generadas

| Tipo       | Algoritmo | Resolución  | Tiempo Ejecución [seg] | Uso de Memoria [byte] |
|------------|-----------|-------------|------------------------|-----------------------|
| MultiFrame | IBP       | $\approx 2$ | $0.3351 \pm 0.0094$    | 11810152              |
|            | POCS      | $\approx 3$ | $6.9535 \pm 0.0351$    | 877712                |
|            | ROBUST    | $\approx 4$ | $0.2396 \pm 0.0121$    | 874816                |

Tabla 4.2: Resumen rendimientos algoritmos de SR para grupo imágenes Malvinas (10 imágenes)

### 4.3.3. Grupo imágenes Textos

Aprovechando que la interfaz lo permitía se realizaron pruebas con otras imágenes de entrada, así por ejemplo, un conjunto de imágenes estándares que contienen texto en el. Como se muestra en la Figura 4.10, y que presenta un gran contraste entre el texto y el fondo.

Haciendo el mismo tipo de pruebas que con el grupo de imágenes Eolian podemos presentar los resultados para el grupo de imágenes Textos, los cuales se presentan en la Tabla 4.3.

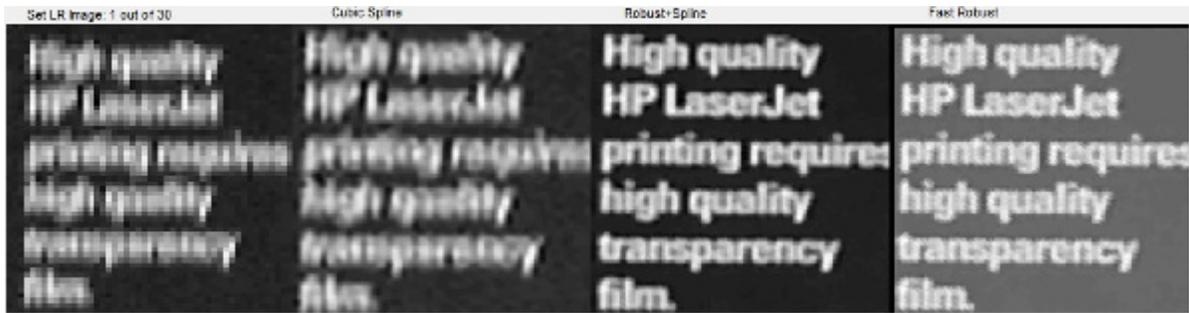


Figura 4.10: Comparación de imagen base texto y sus SR generadas

| Tipo       | Algoritmo | Resolución  | Tiempo Ejecución [seg] | Uso de Memoria [byte] |
|------------|-----------|-------------|------------------------|-----------------------|
| MultiFrame | IBP       | $\approx 2$ | $0.1562 \pm 0.0057$    | 28540940              |
|            | POCS      | $\approx 3$ | $12.9404 \pm 0.0931$   | 1104768               |
|            | ROBUST    | $\approx 4$ | $0.4026 \pm 0.0066$    | 1456176               |

Tabla 4.3: Resumen rendimientos algoritmos de SR para grupo imágenes Textos (30 imágenes)

#### 4.3.4. Grupo imágenes Emily

Ahora utilizando otro grupo de imágenes estándares, llamado *Emily*, haremos las mismas pruebas. Las cuales se pueden apreciar en la Figura 4.11. En este caso la figura es de una persona de la cual se puede identificar desde una parte del brazo izquierdo, el torso, la cabeza y el pelo, sin mucho contraste con el fondo del cuadro.



Figura 4.11: Comparación de imagen base Emily y sus SR generadas

En este caso se distingue perfectamente la mejora en la resolución espacial al aplicar SR (especialmente el método *ROBUST*). Ya que ahora se puede observar con suficiente detalle los ojos, rasgos de la cara y algunas letras de la camiseta observada.

Luego los resultados de las pruebas realizadas se muestran en la Tabla 4.4.

| Tipo       | Algoritmo | Resolución  | Tiempo Ejecución [seg] | Uso de Memoria [byte] |
|------------|-----------|-------------|------------------------|-----------------------|
| MultiFrame | IBP       | $\approx 2$ | $0.2208 \pm 0.0061$    | 29017376              |
|            | POCS      | $\approx 3$ | $0.8023 \pm 0.0341$    | 1179104               |
|            | ROBUST    | $\approx 4$ | $0.3069 \pm 0.0054$    | 1690304               |

Tabla 4.4: Resumen rendimientos algoritmos de SR para grupo imágenes Emily (53 imágenes)

#### 4.3.5. *Mapas de Calor* de algoritmos considerados

Como complemento a las evaluaciones de algoritmos de SR sobre imágenes individuales midiendo sus tiempos de ejecución y uso de memoria para cada algoritmo (utilizando ejemplos de dataset estándar, una fotografía obtenida por el Cubesat SUCHAI 1, y una imagen de una vista aérea del departamento de ingeniería eléctrica), podemos utilizar un despliegue llamado *Mapas de Calor*.

El *Mapa de Calor* consiste en un conjunto de imágenes dispuestas en una matriz, donde en cada celda se obtiene un valor numérico correspondiente a la evaluación de su Peak Signal to Noise (PSNR [86], cuya fórmula puede ser encontrada en la sección 2.4 de esta memoria)<sup>4</sup>, entre la imagen obtenida por el algoritmo de SR y la imagen original, cada valor resultante es asignado a un color (usando un mapa de color específico) el cual es desplegado a un lado para mostrar una gráfica más clara[87].

En este caso se eligieron 100 imágenes (una matriz de 10x10, Figura 4.12) de la superficie terrestre, tomadas de Google Map, de manera de representar las posibles imágenes que un satélite podría obtener. Cada algoritmo se aplica a toda la grilla de imágenes y se obtiene una matriz de valores PSNR, los cuáles los podemos relacionamos con un color determinado, con lo cual se observa una imagen de colores (o patrón) que permite realizar una comparación con otras evaluaciones (por ejemplo, el resultado de otro algoritmo).



Figura 4.12: 100 imágenes de la superficie terrestre

<sup>4</sup>Es necesario mencionar que, aunque el PSNR nos provee de una herramienta cuantitativa para comparar imágenes, esta comparación debe ser hecha entre clases de algoritmos similares. PSNR se desarrolló como una estimación cuantitativa de la calidad de la percepción humana a una imagen

Ahora podemos comparar la ejecución de los algoritmos utilizando los *Mapas de Calor*. El valor máximo corresponde a una estimación de que se dispone de una mejor representación de la imagen original (y el valor más pequeño a las imágenes que tienen el peor desempeño para estos algoritmos). Se ha dispuesto la barra de color en escala log para mejorar su entrega de información visual.

Para mejorar la apreciación visual de los *Mapas de Calor*, se procedió a ordenar (de menor a mayor y de arriba a abajo) los PSNR obtenidos de la grilla de imágenes obtenidas por el algoritmo IBP, con lo que obtenemos el cuadro simétrico en el lado izquierdo de la Figura 4.13. Asimismo ocupando el mismo orden en los resultados de los algoritmos POCS y ROBUST podemos buscar patrones que nos permitan comparar sus rendimientos.

En la Figura 4.13, podemos observar que los algoritmos IBP y ROBUST al parecer se comportan mejor que el POCS para las imágenes incluidas en esta comparación, ya que el color de índice mayor están más presentes en estos casos y nos inducen a pensar que tienen mejor desempeño). De los tres, ROBUST presente un mayor número de PSNR altos, lo cual nos confirma que sería razonable utilizar este algoritmo para procesar imágenes de satélite.

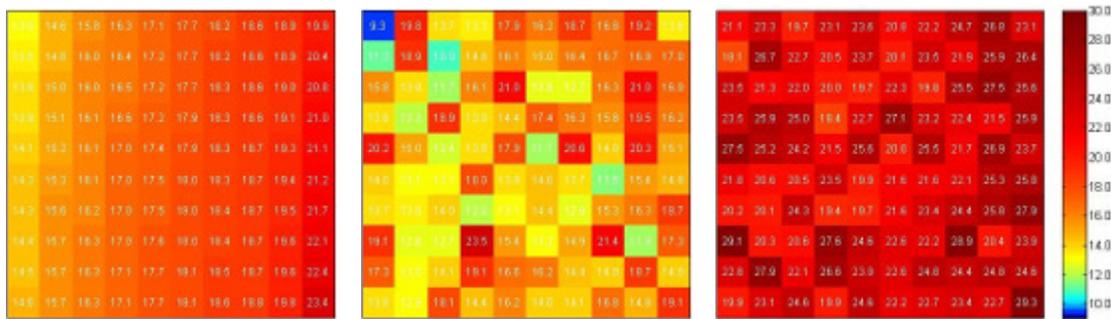


Figura 4.13: *Mapas de Calor* Algoritmos: IBP-POCS-ROBUST

Para buscar patrones y poder determinar cuáles son las imágenes que sean más adecuadas y en cuales son los de peor rendimiento en el algoritmo IBP, podemos poner junto la matriz ordenada de los *Mapas de Calor* del resultado de los PSNR junto a su grilla de imágenes. Lo que veríamos sería lo indicado en la Figura 4.14.

En el mapa IBP no se observa que tengan una característica de agrupar estructuras entre las de menor a mayor PSNR. Si no más bien que las imágenes que tienen más zonas con vegetación son las que presenten un mejor desempeño para este algoritmo (aunque los mejores de IBP solo alcanzan un PSNR que está en la mitad del rango de los resultados de ROBUST).



Figura 4.14: *Mapas de Calor* IBP ordenados y sus imágenes

Si ahora hacemos la comparación con el resultado de la evaluación del algoritmo POCS, obtendremos la Figura 4.15. Aquí se ven más distribuidas las estructuras y zonas de vegetación que en el caso de IBP. Ello nos lleva a considerar que el algoritmo POCS tiene un rendimiento más parejo para las imágenes del dataset utilizado.

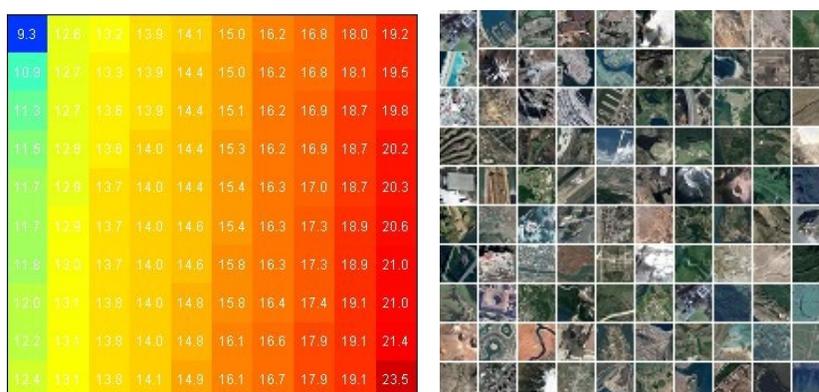


Figura 4.15: *Mapas de Calor* POCS ordenados y sus imágenes

Y por último, haciendo la búsqueda de patrones para el algoritmo ROBUST, obtendríamos la Figura 4.15. Claramente, este tiene un mejor desempeño que los otros algoritmos (tiene un rango de PSNR mayor que los otros y siempre cercanos a los mejores valores de los 3 algoritmos, por eso el color rojo de la matriz ROBUST). Aunque se nos hace difícil determinar cuál es la estructura que mejor desempeño tiene en ROBUST, igual podemos enfocarnos en elegir a ROBUST dado su alto rango de valores).



Figura 4.16: Mapas de Calor ROBUST ordenados y sus imágenes

Por último, los rangos de los valores PSNR de cada algoritmo se pueden apreciar en la Figura 4.17. Podemos apreciar que ROBUST es quien tiene el mejor desempeño de los 3, con un rango y un promedio mejor que los otros.

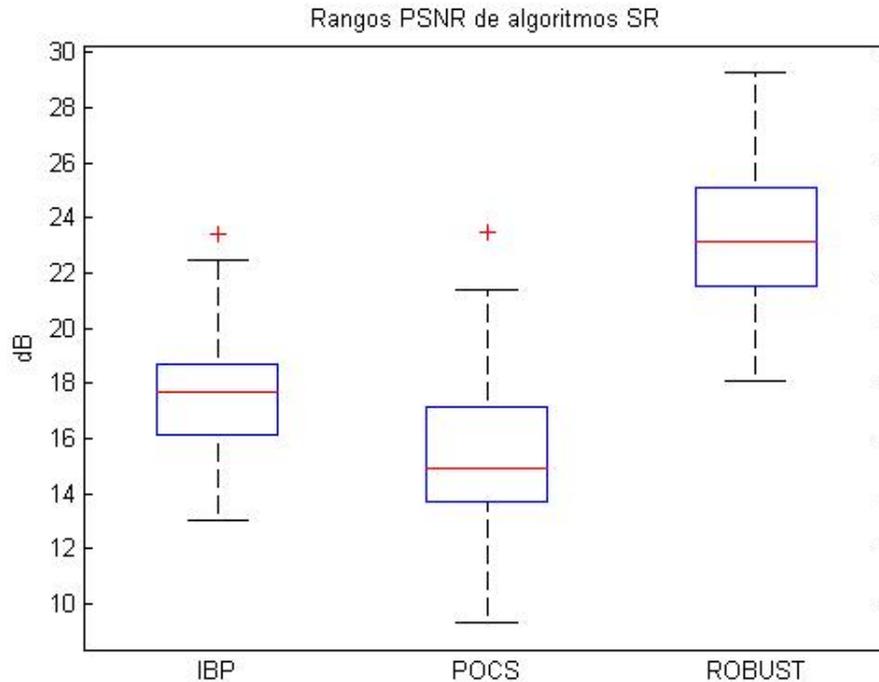


Figura 4.17: Rangos PSNR de los algoritmos

## 4.4. Análisis de resultados obtenidos

Se presenta a continuación una discusión de los resultados obtenidos en los experimentos y simulaciones efectuados con los modelos elegidos.

De los datos obtenidos del *uso de memoria entre IBP, POCS y ROBUST* vemos que de la imagen básica de 128x128 píxeles (o sea de 49.152 byte), para IBP aprox. 7 MB y para POCS se necesita casi 570 kB, por una parte. Ahora si consideramos los tiempos de ejecución, IBP puede procesar varias imágenes en rápida secuencia (algo bastante deseable si se quiere obtener alta información desde una plataforma en órbita), mientras que el algoritmo POCS solo podría tener respuestas en modo batch para imágenes tomadas espaciadas varios minutos.

A su vez ROBUST tiene un consumo de memoria similar al de POCS pero un tiempo de ejecución significativamente menores. Estos parámetros son los más significativos a considerar, ya que harán factible su uso en la plataforma *Cubesat* (en un futuro SUCHAI).

Vemos que utilizando los *Mapas de Calor* se observa una cierta aleatoriedad en el desempeño de los algoritmos en las imágenes (o si se quiere, que tienen desempeños similares) con lo que podemos priorizar con plena confianza los parámetros de memoria y tiempo de ejecución al escoger el algoritmo a sugerir para proceder a su implementación en el futuro SUCHAI.

Considerando la necesidad de entrenamiento para algunos algoritmos, se puede separar a priori los algoritmos de manera que las imágenes que serán procesadas en tierra utilicen algoritmos *Deep Learning*. Esto significa que una vez definido la estructura de la red convolucional esta debe ser entrenada con los millones de ejemplos que ajustaran los pesos de la red neuronal, esto probablemente necesitara un tiempo significativo antes de poder utilizar la red *Deep Learning*. Por otra parte, como los aprendizajes dependen de la imagen específica a procesar (mejorar la resolución en nuestro caso), los conjuntos de aprendizajes deberán ser personalizados dependiendo del tipo de estructura (o superficie) que se quiera reconocer en la imagen, lo que implicara fijar primero el tipo de imagen a procesar, luego obtener el conjunto de aprendizaje y por último aplicar la red ya entrenada a la imagen a la cual se desea mejorar la resolución.

De lo anterior podemos establecer que de los algoritmos que no requieran entrenamiento, es posible obtener a los candidatos para su ejecución en la electrónica a bordo de los *Cubesat*. Así nos quedan los SR tradicionales, como IBP, POCS y ROBUST.

De entre ellos vemos que IBP es de peor resultado en cuanto a su resolución, pero utiliza menos recursos (medido en su tiempo de ejecución) que cualquier otro.

POCS, por su parte mejora la resolución obtenida, pero a un precio de un tiempo de ejecución varias veces mayor y con un requerimiento de memoria también significativo. Lo más probable es que más que los tiempos de ejecución o el uso específico de memoria lo que debemos considerar son las razones entre los tiempos y los usos de memoria, ya que al cambiar a otra plataforma estas razones muy probablemente se mantendrán.

ROBUST, usa un espacio de memoria similar al de POCS, pero en un tiempo de ejecución

más bajo y con un factor de resolución mejor (en la mayoría de los casos), posicionándose como el algoritmo, al parecer más adecuado, a ser implementado.

## 4.5. Consideraciones en los *Cubesat*

En el SUCHAI 1, se puede utilizar una velocidad de bajada de datos de 1.2, 2.4 o 4.8 kbps, tiene una pequeña cámara y una tarjeta de compresión VGA C329-SPI, el cual permite obtener una imagen de 640x480 píxeles -VGA- (aunque también puede enviar de 320x240) y entrega la imagen en formato jpeg (compresión con pérdida, o sea la imagen se ha reducido en tamaño y reduciéndose la información disponible en el proceso), de los cuales se han obtenido imágenes con tamaños entre 30 y 20 kbyte (aproximadamente se puede bajar hasta 90 kbyte de datos en cada pasada, pero de ellos por problemas de ruido en la línea solo es efectivo y en el mejor de los casos 45 kbyte, según la experiencia acumulada en la estación terrena entre el lanzamiento del SUCHAI 1 y finales de Diciembre de 2017).

Entre las pruebas realizadas se probó 4.8 kbps de velocidad de transmisión y se obtuvieron muchas pérdidas en la recepción y al probar con 1.2 kbps los datos obtenidos eran escasos, así que al final se utiliza 2.4 kbps (como solución de compromiso). La transmisión es en la Banda UHF (de 300 MHz a 1000 MHz) específicamente a los 430 MHz.

En los SUCHAI 2 y 3 se estima que se utilizará del orden de 9.6 kbps de velocidad de transmisión de datos entre satélite y estación terrena, y el tiempo de transmisión -dado por el tiempo que es visible el satélite en cada pasada- es menor a 10 minutos. Considerando que junto a la información de las cargas útiles (payload) se necesita información de estado del satélite, se tendrá por cota superior 5760000 bits (o 720 kbyte) los datos que podrán ser enviados en cada pasada de los SUCHAI.

Una buena opción para transmitir datos es la banda S (entre 2 y 4 GHz), así en los futuros SUCHAI se podría usar 115 kbps, lo que permitiría bajar y subir una gran cantidad de información, pero al coste de desarrollar nueva electrónica en el satélite y en la estación terrena con el objeto de tener la capacidad de emitir y recibir en este rango de frecuencias.

Para usar adecuadamente la capacidad de SR es deseable que las imágenes obtenidas se envíen a la estación terrena en forma comprimida (con el objeto de optimizar el ancho de banda utilizado) pero sin pérdidas, de manera de no perder información obtenida en los procesamientos.

En el SUCHAI 1 se utilizó el formato jpeg, ya que estaba disponible un circuito de costo accesible que realizaba la transformación (raw to jpg) y así poder disminuir el ancho de banda asociado a la imagen obtenida por la cámara utilizada. En la literatura se indica que una buena opción de compresión sin pérdida está dada por el algoritmo deflación [88] (en inglés Deflate), el cual es, un algoritmo de compresión de datos sin pérdidas que usa una combinación del algoritmo LZ77 y la codificación *Huffman*.

Se podría considerar una alternativa de compresión sin pérdida el usar software comerciales actualmente disponibles [89], quien dispone de chips y software a ser integrado en los circuitos

del satélite y en la plataforma en tierra. Como se muestra en la Figura 4.18. Por supuesto, esto significa intervenir las actuales plataformas en servicio (la del SUCHAI y la estación terrena para adaptarse a este dispositivo).

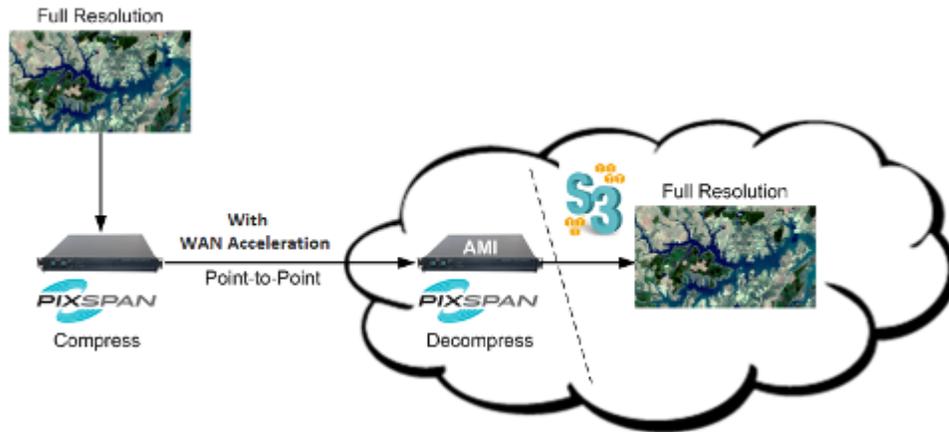


Figura 4.18: Compresión Pixspan

Otra opción de compresión, que representa el estado del arte se puede observar en [90] (PhD Tesis 2017), pero tiene la desventaja de requerir una FPGA Xilinx de a lo menos 100 MHz, el cual podría consumir del orden de 5 W en el procesamiento. Aunque trabajos en compresión sin pérdidas para satélites (incluso usando elementos comerciales -COTS-) [91, 92] hay hace años, los actuales agregan facilidades como por ejemplo, comprimir parte de la imagen, utilizar distintos algoritmos de compresión según las características en frecuencia de las regiones de la imagen, indexar las partes de la imagen, enviar sub-cuadros de la imagen total, entre otros. (funciones que habrían servido mucho en la operación del SUCHAI 1, el cual ha sufrido por la limitación de su velocidad de bajada).

## 4.6. Situación de casos entre algoritmos y escenarios

Con lo mencionado en las secciones precedentes podemos establecer que los algoritmos de SR basados en multiframe y que no tienen etapa de aprendizaje, pueden ser implementados en los *Cubesat*, siempre y cuando sus rendimientos (en tiempo) y uso de memoria sean soportados en la electrónica del *Cubesat*. Tenemos el ejemplo de la Constelación de satélites Flock (de la empresa Planet Lab), la cual consiste en *Cubesat* de 3U, que sirven como cámaras para monitoreo remoto de la superficie, las cuales tienen un telescopio con una abertura de entrada de luz de cercano a los 10 cm, y con una resolución del orden de los 3.5 m por píxel y una capacidad de imágenes del orden de los 29 MPíxeles. Como este satélite hace uso de banda S, disponen de 110 kbps para bajar sus imágenes y aquí podríamos colocar sin ninguna restricción el algoritmo más lento o que use mucha memoria para obtener las imágenes procesadas y obtener evidentes mejoras en sus prestaciones.

Al parecer, los futuros SUCHAI solo dispondrán de un enlace de 9.6 kbps y una electrónica ligeramente de mayores prestaciones que la actual (es esperable que la tarjeta que se llegue

a usar tenga la memoria suficiente para ejecutar algoritmos como el POCS o el ROBUST, pero con una velocidad de bajada bastante limitadas, lo cual hará complicado poder utilizar imágenes de más de un MPíxeles). No resulta arriesgado suponer que para el caso de los SUCHAI 2,3 la opción practica es implementar el algoritmo ROBUST (con más memoria que el SUCHAI 1), junto a un algoritmo de compresión de imágenes sin perdida y no pensar en imágenes de más de 1 MPíxeles.

| Algoritmo | Escenarios                                  | Resolución | Lugar procesamiento |
|-----------|---|------------|---------------------|
| ROBUST    | Bajo procesamiento,<br>Respuesta $\sim seg$ | $\sim 4$   | Órbita              |
| POCS      | Bajo procesamiento,<br>Respuesta $\sim seg$ | $\sim 3$   | Órbita              |
| IBP       | Bajo procesamiento,<br>Respuesta $\sim seg$ | $\sim 2$   | Órbita              |

Tabla 4.5: Sugerencias de aplicación de algoritmos de SR

# Capítulo 5

## Conclusiones y trabajos futuros

Con la acumulación de los resultados de las ejecuciones de las implementaciones utilizadas y los datos obtenidos de las comparaciones, se logró determinar un algoritmo de SR que en forma práctica y con algunas ventajas puede ser utilizado en la electrónica de las futuras versiones de los Cubesat SUCHAI. Como las circunstancias pueden variar se han establecido escenarios donde usar los algoritmos y así priorizar las opciones a utilizar según el diseño de misión que se llegue a establecer a futuro.

Tomar los tres algoritmos y llevarlos a una plataforma común de manera que sobre ellos se tuviera como fuente las mismas entradas motivo a utilizar una interfaz que mostrase los resultados de cada ejecución de las implementaciones. Con esta interfaz se pudo realizar comparaciones visuales (además de registrar los valores numéricos de cada ejecución).

Se encontró que los tres algoritmos pueden ser ejecutados en una plataforma *Cubesat*, pero cada uno tenía un factor de resolución distinto. IBP y ROBUST tienen un tiempo de ejecución menor. IBP tiene un uso de memoria mayor que los otros. Aun con estos resultados se necesitaba agregar cual mostrase mejores valores sobre una muestra de imágenes objetivo de percepción remota.

Se implementó un dataset de 100 imágenes tomadas de Google Map, las cuales se utilizaron midiéndolas con la métrica PSNR. Como resultado se obtuvieron *Mapas da Calor*, donde se podía establecer que el algoritmo ROBUST presentaba un rendimiento superior a los otros algoritmos y apoyaba la sugerencia que el algoritmo a utilizar en el futuro SUCHAI sea este.

Como un paso adicional a la evaluación de los algoritmos y para aportar el proyecto SUCHAI se tradujo el código Matlab de la implementación ROBUST al lenguaje C, de manera que pudiera ser utilizada por el equipo que este a cargo de los SUCHAI 2 y 3. Se creó un proyecto en el repositorio de proyectos GitHub, para que sea fácilmente accesado la última versión de estos códigos (la dirección puede ser consultada en el Anexo: Códigos).

Junto con la implementación de los algoritmos de SR multiframe, sus ejecuciones y al estudio bibliográfico efectuado de la literatura se determinó que la codificación efectuada de estos no es la única posible ni la de mayor rendimiento: En realidad los algoritmos de SR

de tipo IBP, por ejemplo, no son solo un determinado y único código sino una familia de algoritmos que pueden ser intervenidos y mejorados a medida que nuevos aportes se agregan a la disciplina.

Habiendo acotado las posibles capacidades de los futuros Cubesat SUCHAI, es claro que cuando se dispone de baja tasa de transmisión de datos, no es lógico disponer de cámaras de alta resolución y junto a ellos de procesamiento para aumentar dicha resolución dado el cuello de botella que implicaría el tiempo que se necesitaría para poder bajar una foto después de muchas pasadas sobre la estación terrena. No es práctico.

De la experiencia adquirida en la operación de SUCHAI se puede sacar algunas conclusiones, por ejemplo, SUCHAI utiliza una velocidad de bajada de datos de 2.4 kpbs, puede obtener imágenes de 640x480 (en la práctica baja de 320x240) disminuyendo el tamaño de la imagen y utiliza compresión JPG (con pérdida) para ello. Al final, la imagen que obtiene el operador del SUCHAI tiene un tamaño del orden de 30 kbyte, y en ese contexto hay un equilibrio entre los recursos (tiempo, velocidad de transmisión y las capacidades de la electrónica del SUCHAI, etc.).

Si consideramos ahora, otro *Cubesat*, pero que quiere mejorar la resolución de sus imágenes procesando a bordo las tomas antes de bajarlas, necesita realizar una compresión, pero sin pérdida, de manera de no perder el trabajo que significa el aumento de resolución a bordo. Lo que significa que mínimo tiene que disponer de la capacidad de procesamiento del aumento de resolución y de la compresión sin pérdida. Como se supone que las capacidades del sensor (CCD o CMOS) serán las que se disponen actualmente, estamos hablando de MegaPíxeles a ser procesados, comprimidos y enviados a la estación terrena, esto solo es viable con velocidades de transmisión mayores a 100 kbps (en realidad deberíamos hablar desde Megabits por segundo, ya que la órbita de estos satélites esta entre 500 y 700 km sobre la superficie, lo que da un tiempo de sobre vuelo a la vista de las estaciones terrena de a lo más 10 minutos, tiempo límite para la bajada de datos, a menos que el dueño del *Cubesat* disponga de varias ubicaciones para las estaciones de manera que se puedan coordinar, como tiene Planet Labs).

Habiendo determinado que la implementación del algoritmo ROBUST para el caso de un SUCHAI (con sus restricciones), no se hizo ninguna comparación medible de las alternativas de compresión de imágenes sin pérdida, sin embargo, en el capítulo 4, se indican varias opciones que se sugiere pueden ser estimadas de manera que complementen al uso de ROBUST, en la plataforma SUCHAI futura. No es menor, aunque sea reiterativo la necesidad de disponer de una mayor velocidad de transmisión de datos entre los SUCHAI y las estaciones terrenas, ya que constituye un cuello de botella a cualquier sistema de SR+*compresión de imágenes* (la velocidad de transmisión o si se prefiere el ancho de banda utilizado por los satélites, especialmente el tramo de bajada es un bien escaso).

Después de establecer el cómo se aplican los algoritmos de SR, ahora podemos establecer dónde y cuándo usar el procesamiento en tierra o en la plataforma satelital. Si tenemos SR en órbita dispondremos de las imágenes apenas el dispositivo las capte. Esto tiene sentido dependiendo de los tiempos en los cuales se necesitan las imágenes. Si fuera nuestro SUCHAI una versión de un telescopio espacial (como el Hubble) no sería necesaria tener la SR en la plataforma en el espacio, pero si sería necesario el disponer de compresión sin pérdida para la bajada de las imágenes (y no perder información) antes de que sean procesadas en tierra.

Ahora, poniéndonos en otro caso antagónico, si la cámara sirve como mecanismo de control de actitud de un satélite (digamos como un star tracker) es deseable que la imagen obtenida sea procesada a la brevedad y con las mejores prestaciones). Es verdad que la resolución de un star tracker no son necesariamente las más altas ya que se limita a buscar entre las estrellas más brillantes, identificarlas y determinar la orientación del satélite con base en sus posiciones conocidas en el cielo, pero, por ejemplo, mejorando su resolución se podría aumentar las capacidades de los satélites (el star tracker de la sonda Juno, que al momento de escribir esta memoria se encuentra orbitando el planeta Júpiter pudo obtener nueva evidencia de la distribución de nubes en su superficie y mostrar detalles nunca antes vistos de la formaciones y estructuras sobre los polos de Júpiter).

Resumiendo, si se quiere usar SR se necesita que también estén disponibles algoritmos de compresión sin pérdidas para el envío de estas a la tierra de manera de aprovechar las ventajas de la SR. Si se necesita que la imagen esté disponible en corto tiempo (y se tiene el adecuado hardware y velocidad de transmisión de datos) una buena opción es tener SR a bordo. Si se dispone de suficiente tiempo para mejorar la imagen (o se necesita tener la mayor resolución) la opción sería transmitir la imagen sin pérdida a la tierra, y procesar donde se disponga de mejores equipos.

Un tema no considerado en el presente trabajo, pero que sería interesante estudiar en el futuro es el límite de SR. En los escasos artículos sobre las fronteras de la SR destaca el de [93], el cuál estima un cierto límite en el caso de aplicar filtros físicos de SR. Posteriormente en [94, 95] se refinan estas estimaciones, pero sobre límites de SR geométrico (como los vistos aquí) no se encontró material. De trabajarse en el tema se podría responder la pregunta de: dado un grupo de imágenes a procesar cual es la mejor resolución a obtener (en este momento solo podemos indicar cuál es la resolución obtenida para un conjunto de imágenes que sean procesadas usando un algoritmo específico, y después de un esfuerzo de prueba y error buscando la mejor resolución de esta combinación).

## 5.1. Algoritmos de SR que utilizan DL

Tal como se menciona en [96], *DL* todavía no logra alcanzar las capacidades del cerebro humano, ya que *DL* debe aprender de un conjunto significativo de enseñanza, en cambio la mente de un niño necesita pocos ejemplos de los cuales generalizar para producir sus modelos del mundo. Así, aunque *DL* logre significativos avances en el procesamiento de imágenes (y en SR en particular) hay espacio para nuevos perfeccionamientos que nos permiten esperar mejores resultados de esta técnica en SR en el futuro.

Como todo en la vida, *DL* tiene de luz y de sombra, la ventaja es que produce un aumento de resolución significativo, a un costo computacional elevado (especialmente considerando su etapa de aprendizaje). Una vez entrenada la red neuronal, esta puede funcionar en forma independiente y sin necesidad de muchos recursos computacionales (digamos una CPU con elevados requerimientos de velocidad y mucha memoria donde procesar).

En [37], observamos una elevada capacidad de mejorar la resolución de las imágenes de

entrada, pero a costa de un elevado tiempo de aprendizaje, se necesita un enorme conjunto para enseñarle a la red y de disponer de las habilidades de manejar estas redes. DL está en pleno auge y no sería extraño que logre mejores resultados en poco tiempo más. Existe el pequeño detalle de los costos de aprendizaje para quienes recién nos incorporamos al tema, pero una vez superados DL se vuelve una herramienta realmente significativa.

## 5.2. Algoritmos de SR, DEM y RGB-D

No está de más mencionar, la sinergia existente entre la SR y el modelamiento 3D de planos digitales. Aunque hasta ahora han llevado vertientes independientes, existen evidentes ventajas de un uso combinado de ambas técnicas, en varios tipos de aplicaciones. Se menciona en la literatura los DEM junto a los RGB-D (unos como herramientas de cartografía y los otros como ensayos en labs), siendo que han tenido usos y usuarios muy distintos. La visión de futuro del uso conjunto de estas herramientas podría permitir que a una imagen mejorada por SR se tenga en forma complementaria su información de altura sobre el terreno de partes de sus estructuras y con la percepción 3D se percibiría detalles no considerados con el uso de estas técnicas por separado.

# Bibliografía

- [1] Cubesat.org. (2017) Cubesat Design Specification (CDS) REV 13 (accessed December 7, 2017). [Online]. Available: <http://www.cubesat.org>
- [2] S. Borman and R. Stevenson, “Super-resolution from image sequences: A review,” *Published in: Circuits and Systems, 1998. Proceedings 1998 Midwest Symposium*, pp. 1–5, 1998.
- [3] J. Tian and K.-K. Ma, “A survey on super-resolution imaging,” *Signal, Image and Video Processing*, pp. 329–342, 2011.
- [4] L. Li, W. Wang, H. Luo, and S. Ying, “Super-resolution reconstruction of high resolution satellite zy-3 tlc images,” *Sensors*, vol. 5, p. 1062, 2017.
- [5] K. H. Lim and L. K. Kwok, “Super-resolution for spot5 - beyond supermode,” *30th Asian Conference on Remote Sensing 2009, ACRS 2009*, 2009.
- [6] P. Goudy, P. Kubic, B. Rouge, and C. Latry, “From image processing concepts to instrument design in remote sensing satellites,” *XI European Signal Processing Conference Eusipco/2002*, 2002.
- [7] F. Caro, “Analysis and development of multi frame super resolution algorithms for astronomical images,” Engineer thesis, Universidad de Chile, Depto. Ciencias de la Computación, 2016.
- [8] J. Kim and H. Shin, *Algorithm & Soc Design for Automotive Vision Systems*. London: Springer, 2014.
- [9] M. Barker, E. Mazarico, G. Neumann, M. Zuber, J. Haruyama, and D. Smith, “New lunar digital elevation model lunar: Selene terrain camera,” *ICARUS*, pp. 346–355, 2016.
- [10] M.-C. Chiang, “Imaging-consistent warping and super-resolution,” Ph.D. dissertation, Columbia University, 1998.
- [11] L. Yue, “Image Super-Resolution: The techniques, applications, and future,” *Signal Processing*, vol. 128, pp. 398–408, 2016.
- [12] J. Salvador, *Example-Based Super Resolution*. London: Elsevier, 2017.

- [13] F. A. El-Samie, M. Hadhoud, and S. El-Khamy, *Image Super-Resolution and Applications*. New York: CRC Press, 2012.
- [14] H. Ur and D. Gross, “Improved resolution from subpixel shifted pictures,” *CVGIP: Graphical Models and Image Processing*, vol. 54, pp. 181–186, 1992.
- [15] W. Shen, lincong Fang, X. Chen, and H. Xu, “Projection onto convex sets method in space-frequency domain for super resolution,” *JOURNAL OF COMPUTERS*, vol. 9, pp. 1959–1966, 2014.
- [16] J. Sun, Z. Xu, and H.-Y. Shum, “Image super-resolution using gradient profile prior,” *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008.
- [17] P. Milanfar, *Super-Resolution Imaging*. Boca Raton, Florida: CRC Press, 2011.
- [18] D. Capel, *Image Mosaicing and Super-resolution*. London: Springer-Verlag, 2004.
- [19] X. X. Nguyen, “Numerical algorithms for image superresolution,” Ph.D. dissertation, Stanford University, 2000.
- [20] S. Zhang, “Application of super-resolution image reconstruction to digital holography,” *EURASIP Journal and Applied Signal Processing*, pp. 1–7, 2006.
- [21] K. Nasrollahi and T. Moeslund, “Super-resolution: a comprehensive survey,” *Signal, Image and Video Processing*, pp. 1423–1468, 2014.
- [22] S. C. Park, M. K. Park, and M. G. Kang, “Super-resolution image reconstruction: A technical overview,” *IEEE Signal Processing Magazine*, pp. 21–36, 2003.
- [23] P. Vandewalle, S. Süsstrunk, and M. Vetterli, “A frequency domain approach to registration of aliased images with application to super-resolution,” *In Proceedings of EURASIP Journal on applied signal processing, 2006*, 2006.
- [24] V. Lakshminarayanan, M. Calvo, and T. Alieva, *Mathematical Optics: Classical, Quantum, and Computational Methods*. Boca Raton, Florida: CRC Press, 2013.
- [25] S. Chaudhuri and J. Manjunath, *Motion-Free Super-Resolution*. New York: Springer, 2005.
- [26] H. Greenspan, “Multi-resolution image processing and learning for texture recognition and image enhancement,” Ph.D. dissertation, California Institute of Technology, 1994.
- [27] F. Ghesu, T. Kohler, S. Haase, and J. Hornegger, “Guided image super-resolution: A new technique for photogeometric super-resolution in hybrid 3-d range imaging,” *36th German Conference on Pattern Recognition, GCPR 2014*, pp. 227–238, 2014.
- [28] J. Thorley, “Super-resolution microscopy a comparison of commercially available option,” *Fluorescence Microscopy*, p. Chapter 14, 2014.

- [29] G. I. M. Callicó, “Mejora de la calidad en imagen estática y vídeo basada en súper-resolución con prestaciones de tiempo real y bajo coste mediante codificación híbrida,” Ph.D. dissertation, Universidad de las Palmas de Gran Canaria, 2003.
- [30] C. Ledig. (2016) Photo-realistic single image super-resolution using a generative adversarial network (accessed december 1, 2017). [Online]. Available: <https://arxiv.org/abs/1609.04802v3>
- [31] V. Bannore, *Iterative-interpolation Super-resolution Image Reconstruction*. Adelaida: Springer, 2009.
- [32] X. Li, “A multi-frame image super-resolution method,” *Signal Processing*, vol. 90, pp. 405–414, 2010.
- [33] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, “Deep networks for image super-resolution with sparse prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 370–378, 2015.
- [34] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 295–307, 2016.
- [35] W. Wu, Z. Liu, and X. He, “Learning-based super resolution using kernel partial least squares,” *Image and Vision Computing*, vol. 29, pp. 394–406, 2011.
- [36] K. Hayat. (2017) Super-resolution via deep-learning (accessed december 1, 2017). [Online]. Available: <https://arxiv.org/abs/1706.09077>
- [37] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1646–1654, 2016.
- [38] L. Genemeier, “When will computers have common sense? ask facebook,” *Scientific American*, vol. June, 2016.
- [39] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. Bharath. (2017) Generative adversarial networks: An overview (accessed december 1, 2017). [Online]. Available: <https://arxiv.org/abs/1710.07035>
- [40] J. R. del Solar, P. Loncomilla, and N. Soto, “A survey on deep learning methods for robot vision,” *Personal Communication*, 2017.
- [41] J. Liu, S. Dai, Z. Guo, and D. Zhang, “An improved pocs super-resolution infrared image reconstruction algorithm based on visual mechanism,” *Infrared Physics & Technology*, 2016.
- [42] I. Dowman, K. Jacobsen, G. Konecny, and R. Sandau, *High Resolution Optical Satellite Imagery*. Dunbeath: Whittles Publishing, 2012.

- [43] R. Sandau, H.-P. Roser, and A. Valenzuela(eds), *Small Satellites for Earth Observation*. Springer, 2008.
- [44] H. Kramer(ed), *Observation of the Earth and Its Environment: Survey of Missions and Sensors*. Heidelberg: Springer Berlin Heidelberg, 1996.
- [45] L. Di and H. Ramapriyan, *Standard-Based Data and Information Systems for Earth Observation*. Heidelberg: Springer-Verlag Berlin Heidelberg, 2010.
- [46] D. McNerney and P. Kempeneers, *Open Source Geospatial Tools: Applications in Earth Observation*. London: Springer International Publishing, 2015.
- [47] F. A. Sarría, *Técnicas de Detección Aplicadas a la Gestión de Recursos Hídricos*. Murcia: Universidad de Murcia, 2008.
- [48] R. Hanel, B. Conrath, D. Jennings, and R. Samuelson, *Exploration of the Solar System by Infrared Remote Sensing*. London: Cambridge University Press, 2002.
- [49] N. L. Trillat, *Manual de interpretación de fotos e imágenes*. Santiago: Servicio Aero-Fotogrametrico, FACH, 1976.
- [50] J. Dijk, A. van Eekeren, O. R. Rojas, G. Burghouts, and K. Schutte, “Image processing in aerial surveillance and reconnaissance:from pixels to understanding,” *Electro-Optical Remote Sensing, Photonic Technologies, and Applications VII*, pp. 1–17, 2013.
- [51] K. Jacobsen, “Hic mapping with high resolution optical space images,” *ISPRS WG IV/2*, pp. 42–57, 2013.
- [52] L. Araneda, “Modelo y plataforma de cómputo heterogéneo para video infrarrojo,” Engineer thesis, Universidad de Concepción, Depto. Ingeniería Eléctrica, 2016.
- [53] R. Lasaponara and N. Masini, *Satellite Remote Sensing: A New Tool for Archaeology*. New York: Springer Netherlands, 2012.
- [54] T. Huang and R. Tsai, *Advances in Computer Vision and Image Processing: Image Reconstruction from Incomplete Observations*. Greenwich, CT: JAI Press Inc., 1984.
- [55] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multiframe super resolution,” *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13, pp. 1327–1344, 2004.
- [56] R. Redlich, L. Araneda, A. Saavedra, and M. Figueroa, “An embedded hardware architecture for real-time super-resolution in infrared cameras,” *Euromicro Conference on Digital System Design*, pp. 184–191, 2016.
- [57] M. Bareja and C. Modi, “An effective iterative back projection based single image super resolution approach,” *In 2002 International Conference on Communication Systems and Network Technologies*, pp. 95–99, 2012.

- [58] A. Nazren, S. Yaakob, R. Ngadiran, M. Hisham, and N. Wafi, "Improving iterative back projection super resolution model via anisotropic diffusion edge enhancement," *Robotics, Automation and Sciences (ICORAS), International Conference on*, vol. Nov, 2016.
- [59] Y. Zhang, "Video superresolution reconstruction using iterative back projection with critical-point filters based image matching," *Advances in Multimedia*, vol. 2015, pp. 1–10, 2015.
- [60] B. Ngocho and E. Mwangi, "An iterative back-projection technique for single image super resolution with natural texture preservation," *In Proceedings of the First International Conference on Electrical, Electronic and Communications Engineering (ELECOM 2016), Bagatelle, Mauritius, November 25-27, 2016*, pp. 210–219, 2016.
- [61] M. Irani and S. Peleg, "Improving resolution by image registration," *GRAPHICAL MODELS AND IMAGE PROCESSING*, vol. 53, pp. 231–239, 1991.
- [62] M. Elad and A. Feuer, "Super-resolution reconstruction of an image," *Electrical and Electronics Engineers in Israel, 1996., Nineteenth Convention of*, 1996.
- [63] V. May. (2017) Image super-resolution - iterative back projection algorithm (accessed december, 2017). [Online]. Available: <https://la.mathworks.com/matlabcentral/fileexchange/33839-image-super-resolution-iterative-back-projection-algorithm>
- [64] C. Fan, C. Wu, G. Li, and J. Ma, "Function estimation of low-resolution remote sensing images projections onto convex sets super-resolution reconstruction based on point spread," *Sensors*, vol. 362, pp. 1–19, 2017.
- [65] R. Lukac, *Computational Photography: Methods and Applications*. Bonn: CRC Press, 2010.
- [66] S. Cheng. (2017) Superresolution demo (accessed december, 2017). [Online]. Available: <https://la.mathworks.com/matlabcentral/fileexchange/30488-superresolution-demo>
- [67] A. Zomet, A. Rav-Acha, and S. Peleg, "Robust super resolution," *In Proceedings of the International Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8-14 December*, pp. 645–650, 2001.
- [68] X. Zhang, M. Tang, and R. Tong, "Robust super resolution of compressed video," *The Visual Computer*, vol. 28, pp. 1167–1180, 2012.
- [69] O. Hanson. (2017) Robust and fast super resolution (accessed december, 2017). [Online]. Available: <http://www1.idc.ac.il/toky/videoProc-07/projects/SuperRes/srproject.html>
- [70] Y. Zhang and M. An, "Deep learning and transfer learning-based super resolution reconstruction from single medical image," *Journal of Healthcare Engineering*, vol. 217, pp. 1–20, 2017.
- [71] J. Wan, D. Wang, S. Chu, P. Wu, J. Zhu, Y. Zhang, and J. L. and, "Deep learning for content-based image retrieval: A comprehensive study," *In 14 Proceedings of the 22nd*

*ACM international conference on Multimedia*, pp. 157–166, 2014.

- [72] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” *In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014*, pp. 1717–1724, 2014.
- [73] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ICLR 2015 Conference on*, 2015.
- [74] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
- [75] N. Nilsson, *Introduction to machine learning*. Rootic Laboratory: Department of Computer Science, Stanford University, 1997.
- [76] L. Pickup, “Machine learning in multi-frame image super-resolution,” Ph.D. dissertation, Oxford University, 2007.
- [77] R. Takaoka and N. Hashimoto, “Depth map super-resolution for cost-effective rgb-d camera,” *2015 International Conference on Cyberworlds*, pp. 133–136, 2015.
- [78] D. Mesquita, M. Campos, and E. Nascimento, “A methodology for obtaining super-resolution images and depth maps fro rgb-d data,” *Conference on Graphics, Patterns and Images, 28 (SIBGRAPI)*, 2015.
- [79] R. Maier, J. Stuckler, and D. Cremers, “Super-resolution keyframe fusion for 3-d modeling with high-quality textures,” *3D Vision (3DV), 2015 International Conference on*, 2015.
- [80] Q. Wang, S. Li, and H. Qin, “Super-resolution of multi-observerd rgb-d images based on nonlocal regression and total variation,” *IEEE Transactions on Image Processing*, vol. 25, pp. 1425–1440, 2016.
- [81] T. Kohler, S. Haase, S. Bauer, J. Wasza, T. Kilgus, L. Maier-Hein, H. Feubner, and J. Hornegger, “Tof meets rgb: novel multi-sensor super-resolution for hybrid 3-d endoscopy,” *Med Image Comput Comput Assist Interv*, vol. 16, pp. 139–146, 2013.
- [82] G. Borenstein, *Making Things See: 3D vision with Kinect, Processing, Arduino, and MakerBot*. Beijing: Maker Media, Inc, 2012.
- [83] R. Wang, *Augmented Reality with Kinect*. Birmingham: Packt Publishing, 2013.
- [84] J. Kramer, N. Burrus, D. Herrera, F. Ehtler, and M. Parker, *Hacking the Kinect*. New York: Apress, 2012.
- [85] Z. Li, Q. Zhu, and C. Gold, *Digital Terrain Modeling: Principles and Methodology*. Boca Raton: CRC Press, 2005.
- [86] wikipedia. (2017) Peak signal-to-noise ratio (accessed december, 2017). [Online]. Available: [https://en.wikipedia.org/wiki/Peak\\_signal-to-noise\\_ratio](https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio)

- [87] ——. (2017) Heat map (accessed december, 2017). [Online]. Available: [https://en.wikipedia.org/wiki/Heat\\_map](https://en.wikipedia.org/wiki/Heat_map)
- [88] D. Salomon, G. Motta, and D. Bryant, *Data compression: The Complete Reference*. London: Springer, 2006.
- [89] Pixspan. (2017) pixspan compress technology (accessed december 7, 2017). [Online]. Available: [http://www.pixspan.com/satellite\\_and\\_surveillance.php](http://www.pixspan.com/satellite_and_surveillance.php)
- [90] K. Manthey, “Reconfigurable architecture for real-time image compression on-board satellites,” Ph.D. dissertation, Technischen Universität Berlin, 2017.
- [91] N. Ismailoglu, O. Benderli, I. Korkmaz, H. Sunay, T. Kolcak, and Y. Tekmen, “Gezgin:a case study of a real-time image processing subsystem for micro-satellites,” *Recent Advances in Space Technologies, 2003. RAST '03. International Conference on. Proceedings of*, pp. 302–307, 2003.
- [92] G. Yu, T. Vladimirova, X. Wu, and M. Sweeting, “A new high-level reconfigurable lossless image compression system for space applicatios,” *Adaptive Hardware and Systems, 2008. AHS '08. NASA/ESA Conference on*, pp. 183–190, 2008.
- [93] T. Sales and M. Morris, “Fundamental limits of optical superresolution,” *OPTICS LETTERS*, vol. 22, pp. 582–584, 1997.
- [94] H. Liu, Y. Yan, Q. Tan, and G. Jin, “Theories for the design of diffractive superresolution elements and limits of optical superresolution,” *J. Opt. Soc. Am. A*, vol. 19, pp. 2185–2193, 2002.
- [95] H. Liu, Y. Yan, and G. Jin, “Design theories and performance limits of diffractive super-resolution elements with the highest sidelobe suppressed,” *J. Opt. Soc. Am. A*, vol. 22, pp. 828–838, 2005.
- [96] M. Yao. (2017) Understanding the limits of deep learning (accessed december 7, 2017). [Online]. Available: <https://venturebeat.com/2017/04/02/understanding-the-limits-of-deep-learning/>

# Anexo: Códigos

Los programas de la versión implementada de ROBUST traducidos a C, se pueden encontrar en el siguiente GitHub: <https://github.com/rodolfojara/SR4SUCHAI>

Probablemente el código podría ser actualizado a medida que se decanten los requerimientos y restricciones de las siguientes versiones de los SUCHAI.

# Anexo: Acrónimos

|        |  |
|--------|--|
| RS     | Super Resolution   |
| GSD    | Ground Sample Distance                                       |
| CCD    | Charged-Coupled Device                                       |
| CMOS   | Complementary Metal Oxide Semiconductor                      |
| FFT    | Fast Fourier Transform                                       |
| PSF    | Point Spread Function  |
| RGB    | Red Green Blue   |
| RGB-D  | Red Green Blue and Deep                                      |
| PSNR   | Peak Signal to Noise Ratio                                   |
| IBP    | Iterative Back Projection                                    |
| POCS   | Projection Onto Convex Sets                                  |
| IFOV   | Instantaneous Field Of View                                  |
| LEO    | Low Earth Orbit  |
| GEO    | Geostationary Earth Orbit                                    |
| SUCHAI | Satellite of university of CHile for Aerospace Investigation |
| NOAA   | National Oceanic and Atmospheric Administration              |
| AVHRR  | Advanced Very High Resolution Radiometer                     |
| IA     | Intelligence Artificial                                      |
| ERTS-1 | Earth Resources Technology Satellite 1                       |
| RADAR  | RADio Detection And Ranging                                  |
| SSO    | Sun Synchronous Orbit  |
| ISS    | International Space Station                                  |
| NIR    | Near InfraRed  |
| HSD    | High-Speed Data  |
| COTS   | Commercial Off The Shelf                                     |
| SIG    | Sistemas de Información Geográfica                           |
| JPEG   | Joint Photographic Experts Group                             |
| UHF    | Ultra High Frecuencia  |
| ADU    | Analogue to Digital conversion Unit                          |
| CID    | Charge injection device                                      |
| DAC    | Digital-Analog Converter                                     |
| FWHM   | Full Width Half Max  |
| HST    | Hubble Space Telescope                                       |
| IRAF   | Image Reduction and Analysis Facility                        |
| RON    | Read-out Noise   |
| TES    | Transition edge sensor                                       |