

**PONTIFICIA UNIVERSIDAD CATÓLICA
DEL PERÚ**

FACULTAD DE CIENCIAS E INGENIERÍA



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ**

**PROGRAMACIÓN DEL ALGORITMO POCS
PARA MEJORAR LA RESOLUCIÓN DE
IMÁGENES DE CAMPOS DE CULTIVO**

Tesis para optar el Título de Ingeniero Electrónico, que
presenta el bachiller:

Daniel Ernesto Pérez Reátegui

ASESOR: Willy Eduardo Carrera Soria

Lima, Julio de 2015

RESUMEN

El presente trabajo de tesis surgió debido a la necesidad de obtener imágenes de campos de cultivo de alta resolución a partir de un conjunto de imágenes de baja resolución capturadas por un dispositivo óptico.

La propiedad de desplazamiento sub-píxel existente entre las imágenes de baja resolución capturadas por el dispositivo óptico hace posible la obtención de imágenes digitales de alta resolución mediante la aplicación de un algoritmo de reconstrucción de imágenes. Con el fin de poder incrementar la resolución de las imágenes de campos de cultivo, de tal modo que estas puedan ser procesadas y analizadas con mayor precisión, se seleccionó y programó el algoritmo POCS (Projections Onto Convex Sets) debido a su robustez, simplicidad y flexibilidad para incorporar información conocida a priori de las imágenes deseadas al proceso de reconstrucción.

El presente trabajo se desarrolla en cuatro capítulos. En el primer capítulo se presenta el concepto de Agricultura de Precisión y la necesidad de aplicar técnicas que mejoren la resolución de imágenes de campos de cultivo. En el segundo capítulo se describen los principales problemas que presentan los dispositivos ópticos para aumentar la resolución de imágenes, se presenta el modelo que describe el proceso de degradación de imágenes y se realiza un recuento de los principales algoritmos que aumentan la resolución de las mismas. En el tercer capítulo se presentan los objetivos del presente trabajo de tesis, se detalla el marco teórico del algoritmo POCS y se muestra la programación del mismo. En el cuarto capítulo se definen las métricas usadas para evaluar el algoritmo y se presentan los resultados de diversas pruebas aplicadas a las imágenes de campos de cultivo.

Finalmente, se concluyó que el algoritmo POCS incrementa la resolución de imágenes de campos de cultivo de manera satisfactoria, además, se concluyó que POCS presentó mejor desempeño con respecto a los algoritmos de interpolación y de deconvolución que formaron parte de las pruebas.

ÍNDICE DE FIGURAS

Figura 1: Aplicación de tecnologías de información para un uso eficiente de los recursos destinados a la agricultura.	3
Figura 2: Esquema de degradación de una imagen digital de una escena.	6
Figura 3: Diagrama de bloques del proceso de degradación de una imagen digital de alta resolución.....	7
Figura 4: Etapas del proceso de reconstrucción de Súper-resolución.	9
Figura 5: Etapas de registro e interpolación.....	10
Figura 6: Comparación entre imagen original (alta resolución) de prueba y la obtenida con POCS.....	15
Figura 7: Noción de convexidad de conjuntos.....	16
Figura 8: Ilustración del teorema de la proyección en dos dimensiones.	19
Figura 9: Aplicación de las proyecciones alternantes a dos conjuntos convexos.	20
Figura 10: Aplicación de proyecciones alternantes a dos conjuntos cuya intersección es nula.	21
Figura 11: Diagrama de flujo de la función que ejecuta el algoritmo POCS.....	29
Figura 12: Aplicación de POCS a las imágenes de baja resolución.....	30
Figura 13: Valores de los elementos de la matriz que caracteriza a la PSF.	30
Figura 14: Generación de imágenes degradadas de Lena y aplicación de POCS.....	37
Figura 15: Curva de porcentaje de error por iteración con imagen inicial cero.	38
Figura 16: Comparación entre la curva de porcentaje de error por iteración con imagen inicial diferente de cero y diferente de cero.....	39
Figura 17: Histograma de Lena (imagen de referencia).....	39
Figura 18: Histograma de la imagen restaurada de Lena.	40
Figura 19: Imágenes de Lena desplazadas a nivel sub-píxel, difuminadas y sub-muestreadas.	40
Figura 20: Lena magnificada y restaurada con POCS.	41
Figura 21: Lena original (alta resolución).	41
Figura 22: Disposición de escenas de campos de cultivo.	43
Figura 23: Generación de imágenes degradadas de campo de cultivo y aplicación de POCS, interpolación bilineal e interpolación bicúbica.	43
Figura 24: Imágenes de campo de cultivo 1 degradadas.....	44
Figura 25: Imagen de campo de cultivo 1 magnificada y restaurada con POCS.	44
Figura 26: Imagen de campo de cultivo 1 magnificada con interpolación bilineal.	44
Figura 27: Imagen de campo de cultivo 1 magnificada interpolación bicúbica.....	45
Figura 28: Imagen de campo de cultivo 1 de referencia (alta resolución).	45
Figura 29: Imágenes de campo de cultivo 2 degradadas.....	46
Figura 30: Imagen de campo de cultivo 2 magnificada y restaurada con POCS.	47

Figura 31: Imagen de campo de cultivo 2 magnificada con interpolación bilineal.	47
Figura 32: Imagen de campo de cultivo 2 magnificada con interpolación bicúbica.	47
Figura 33: Imagen de campo de cultivo 2 de referencia (alta resolución).	48
Figura 34: Aplicación de los algoritmos de restauración de MATLAB a las imágenes resultantes de los algoritmos de interpolación.	49
Figura 35: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución ciega para restauración a campo de cultivo 1.	50
Figura 36: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución Lucy-Richardson para restauración a campo de cultivo 1.	50
Figura 37: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución filtro regularizado para restauración a campo de cultivo 1.	51
Figura 38: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución ciega para restauración a campo de cultivo 1.	51
Figura 39: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución Lucy-Richardson para restauración a campo de cultivo 1.	52
Figura 40: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución filtro regularizado para restauración a campo de cultivo 1.	52
Figura 41: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución ciega para restauración a campo de cultivo 2.	54
Figura 42: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución Lucy-Richardson para restauración a campo de cultivo 2.	55
Figura 43: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución filtro regularizado para restauración a campo de cultivo 2.	55
Figura 44: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución ciega para restauración a campo de cultivo 2.	56
Figura 45: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución Lucy-Richardson para restauración a campo de cultivo 2.	56
Figura 46: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución filtro regularizado para restauración a campo de cultivo 2.	57
Figura 47: Aplicación de algoritmo POCS a las imágenes de campos de cultivo.	59
Figura 48: Aplicación de algoritmo POCS a las imágenes de campos de cultivo de la escena 3.	60
Figura 49: Aplicación de algoritmo POCS a las imágenes de campos de cultivo de la escena 4.	60

ÍNDICE DE TABLAS

Tabla 1: Resultados de la evaluación del desempeño de algoritmos de Súper-resolución.	16
Tabla 2: Descripción de los parámetros de entrada de la función POCS.	28
Tabla 3: Descripción de los parámetros de salida de la función POCS.	28
Tabla 4: Indicadores cuantitativos del proceso de magnificación y restauración de Lena.	42
Tabla 5: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 1.	46
Tabla 6: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 2.	48
Tabla 7: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 1 con los algoritmos de interpolación bilineal y de deconvolución.	53
Tabla 8: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 1 con los algoritmos de interpolación bicúbica y de deconvolución.	53
Tabla 9: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 2 con los algoritmos de interpolación bilineal y de deconvolución.	57
Tabla 10: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 2 con los algoritmos de interpolación bicúbica y de deconvolución.	58
Tabla 11: Indicadores cuantitativos obtenidos al comparar las imágenes de baja resolución originales con las generadas artificialmente con las escenas 3 y 4.	61

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: AGRICULTURA DE PRECISIÓN.....	2
1.1. Definición de Agricultura de Precisión.....	2
1.2. Tecnologías.....	2
1.3. Beneficios de la Agricultura de Precisión.	3
1.4. Agricultura de Precisión e imágenes de campos de cultivo.	4
CAPÍTULO 2: ALGORITMOS DE SÚPER-RESOLUCIÓN	5
2.1. Necesidad de alta resolución de imágenes.....	5
2.2. Definición de Súper-resolución.....	6
2.3. Modelo de degradación.....	6
2.4. Etapas de Súper-resolución.	9
2.5. Características de los algoritmos de Súper-resolución.	10
2.5.1. Reconstrucción en el dominio de la frecuencia.	10
2.5.2. Algoritmos de interpolación.....	11
2.5.3. Algoritmos de regularización.....	11
2.5.4. Proyecciones a conjuntos convexos (POCS).....	12
2.5.5. Retro-proyección Iterativa (IBP).....	12
2.5.6. Populis-Gerchberg (PG).....	13
2.5.7. Reconstrucción Híbrida (ML-POCS).....	13
CAPÍTULO 3: IMPLEMENTACIÓN DEL ALGORITMO POCS	14
3.1. Alcance.....	14
3.2. Objetivos.....	14
3.2.1. Objetivo General.....	14
3.2.2. Objetivos Específicos.....	14
3.3. Justificación de POCS.....	14
3.4. Conjuntos y funciones Convexas.....	16
3.4.1. Conjuntos convexos.....	16
3.4.2. Funciones convexas.....	17

3.4.3. Problema de optimización convexa.....	17
3.5. Proyecciones hacia conjuntos convexos (POCS).	18
3.5.1. Norma vectorial euclidiana.	18
3.5.2. Teorema de la proyección.....	18
3.5.3. Algoritmo de las proyecciones alternantes (POCS).	19
3.5.3.1. Definición del algoritmo de las proyecciones alternantes.....	20
3.5.3.2. Proyecciones alternantes para dos conjuntos convexos.	20
3.5.3.3. Propiedades del algoritmo de las proyecciones alternantes.....	21
3.6. Programación del algoritmo POCS aplicado a Súper-resolución.	22
3.6.1. Súper-resolución con optimización convexa.	22
3.6.1.1. Restricciones de la imagen deseada.....	22
3.6.1.2. Función Objetivo.....	25
3.6.1.3. Determinación de los proyectores.	25
3.7. Programación de POCS.....	26
CAPÍTULO 4: RESULTADOS	34
4.1. Condiciones de las pruebas.....	34
4.2. Indicadores.....	35
4.2.1. Porcentaje de error (ek).....	35
4.2.2. Error cuadrático medio (ECM).....	35
4.2.3. Peak signal-to-noise ratio (PSNR).....	36
4.2.4. Histograma.....	36
4.3. Prueba del funcionamiento de POCS.....	37
4.4. Comparación de POCS con los algoritmos de MATLAB.....	42
4.5. Aplicación de POCS a imágenes de campos de cultivo.....	59
CONCLUSIONES	62
RECOMENDACIONES.....	63
FUENTES	64

INTRODUCCIÓN

La Súper-resolución es el conjunto de técnicas de procesamiento de señales que permiten obtener imágenes de alta resolución a partir de varias imágenes de baja resolución observadas. Dichas técnicas son utilizadas en diversas aplicaciones, como, por ejemplo, imágenes médicas, satelitales, video-vigilancia, campos de cultivo, etc.

A la fecha, existen algoritmos de Súper-resolución en el dominio de la frecuencia y en el dominio espacial, sin embargo, estos últimos son los más utilizados debido a que ofrecen mejores resultados de resolución y más flexibilidad en cuanto a la incorporación de información que se conoce previamente de la solución a la cual se desea llegar.

El principal objetivo del presente trabajo de investigación es programar un algoritmo para mejorar la resolución de imágenes de campos de cultivo. El algoritmo a implementar, perteneciente al dominio espacial, es parte de una etapa de pre-procesamiento, de tal modo que permitirá que las imágenes tengan mayor información pictórica acerca de los campos de cultivo cuyo estado se desea analizar.

Previamente, se realizó un estudio acerca de los algoritmos existentes de súper-resolución de imágenes. A partir de esta investigación, tomando en cuenta las ventajas y desventajas de cada algoritmo, se determinó que el POCS (Projections onto Convex Sets) es adecuado para responder a la necesidad planteada. Esto se debe a su simplicidad de implementación, robustez y óptimos resultados para altos valores de resolución deseada.

Finalmente, se compararán los resultados obtenidos de POCS con los algoritmos de Súper Resolución existentes en MATLAB.

CAPÍTULO 1

AGRICULTURA DE PRECISIÓN

En el presente capítulo se brinda una definición de Agricultura de Precisión, se exponen las tecnologías que forman parte de ella, así como también sus principales beneficios y, finalmente, se presenta la necesidad de incrementar la resolución de imágenes digitales de campos de cultivo.

1.1. Definición de Agricultura de Precisión.

En [1] se define Agricultura de Precisión como la aplicación de tecnologías de información para adecuar el manejo de suelos y cultivos, de tal modo que se pueda obtener datos precisos del estado de los campos y de las herramientas empleadas para trabajar dichos campos; los datos obtenidos con las tecnologías de información permiten la selección y utilización de recursos de forma eficiente, de tal modo que se genere una disminución en costos de producción y reducción de impacto ambiental.

1.2. Tecnologías

Las tecnologías de información que se emplean en la Agricultura de Precisión son las siguientes [2]:

- Sistemas de Posicionamiento Global (GPS)
- Sistema de información geográfica (SIG)
- Banderillero satelital
- Imágenes satelitales
- Imágenes aéreas
- Sensores de rendimiento en tiempo real
- Redes de comunicaciones
- Ordenadores

La figura 1 muestra un ejemplo de aplicación de las tecnologías GPS y SIG (indicadas en la lista anterior) para una gestión eficiente de los recursos destinados a campos de cultivo y también para la solución de problemas comunes que afectan a los mismos, como, por ejemplo, la aparición de plagas, enfermedades y malezas.

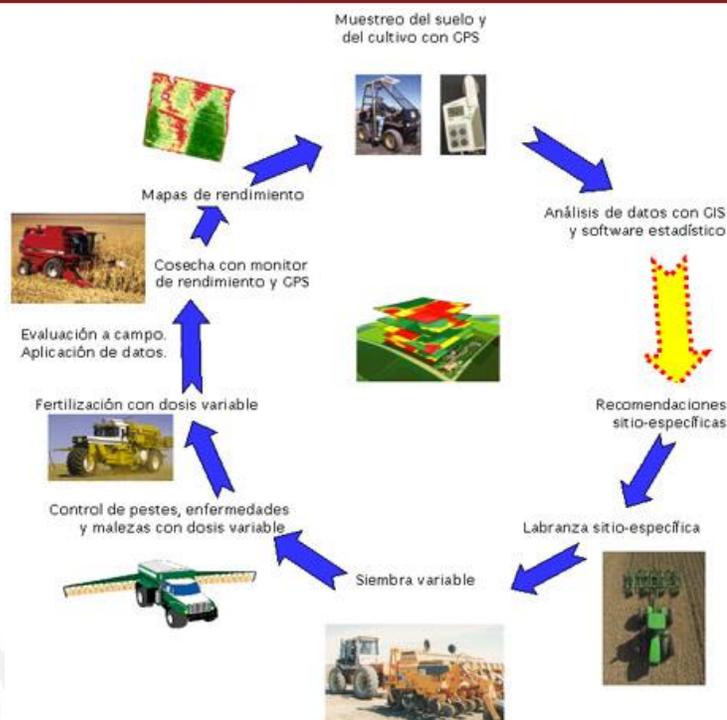


Figura 1: Aplicación de tecnologías de información para un uso eficiente de los recursos destinados a la agricultura.

[3]

1.3. Beneficios de la Agricultura de Precisión.

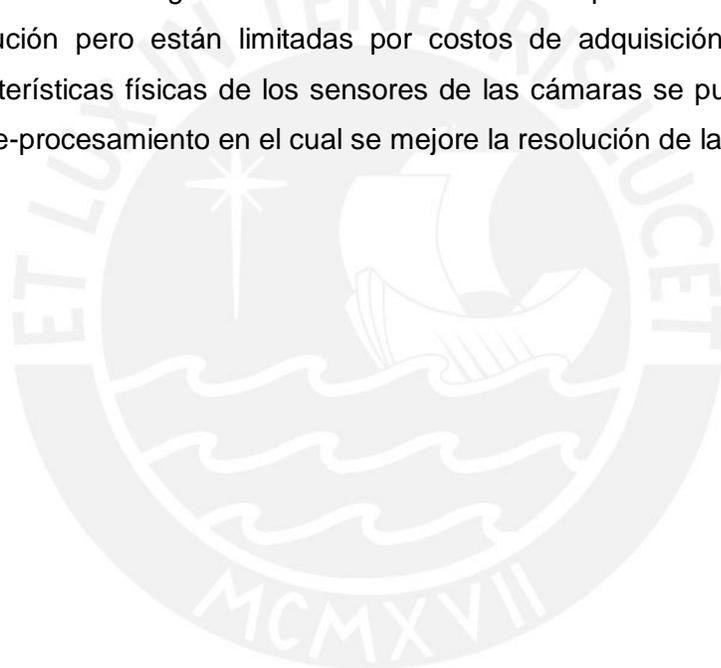
A continuación se presentan algunos de los beneficios que se obtienen al aplicar las tecnologías características de la agricultura de precisión [4]:

- Reducción del uso de pesticidas y fertilizantes.
- Reducción del impacto ambiental.
- Productos alimenticios de mayor valor nutritivo.
- Reducción del uso de combustible en tractores.
- Gestión óptima de explotaciones.

1.4. Agricultura de Precisión e imágenes de campos de cultivo.

En la agricultura de Precisión se utilizan tecnologías y procedimientos que involucran imágenes (satelitales o aéreas) para obtener datos precisos del estado de los campos de cultivo, sin embargo, la adquisición imágenes de alta calidad, en términos de la resolución, está sujeta a costos, al tipo de sensores con los que se capturan las imágenes y a las condiciones con las que se toman las fotografías (caso de imágenes aéreas) [5].

Por otro lado, a mayor resolución de imágenes de campos de cultivo, se puede realizar un análisis de mayor detalle y fidelidad del estado de los mismos, por lo tanto, para aplicaciones de Agricultura de Precisión en las que se requieren imágenes de alta resolución pero están limitadas por costos de adquisición de imágenes o por las características físicas de los sensores de las cámaras se puede incorporar un bloque de pre-procesamiento en el cual se mejore la resolución de las imágenes adquiridas.



CAPÍTULO 2

ALGORITMOS DE SÚPER-RESOLUCIÓN

En el presente capítulo se exponen los principales problemas que existen al intentar incrementar la resolución de imágenes digitales, luego se da una definición del término “Súper-resolución”, posteriormente se presenta el modelo de degradación de las imágenes digitales, el cual conforma el marco teórico básico o de referencia para el diseño de los algoritmos que incrementan la resolución de imágenes, finalmente, se realiza una descripción de las principales características de los algoritmos de Súper-resolución.

2.1. Necesidad de alta resolución de imágenes.

En muchas disciplinas en las que se emplean imágenes tales como astronomía, medicina, video-vigilancia, agricultura de precisión, entre otras, es necesario obtener imágenes digitales de mayor resolución con respecto a la cámara con la que se obtuvieron dichas imágenes, sin embargo, existen limitaciones a nivel de hardware inherentes en los sistemas de adquisición de imágenes que impiden que se obtengan imágenes de mayor resolución [6].

El tamaño de pixel es un factor determinante en la resolución, ya que, mientras menor sea el tamaño de pixel, habrá mayor área disponible para formar una imagen con ellos pero una excesiva reducción limitaría la cantidad de luz disponible, por lo que, en consecuencia, se generaría ruido de disparo, el cuál degrada la calidad de una imagen [6]. El tamaño óptimo de pixel es alrededor de $40\mu m^2$ para un sensor CMOS de $0.35\mu m$ y las tecnologías de fabricación de sensores están llegando a este límite [6].

Por otro lado, el incremento de tamaño del chip de sensores, con un tamaño de pixel fijo, podría aumentar la resolución espacial de las imágenes pero esto provocaría un aumento en la capacitancia y, debido a que la capacitancia es un factor que determina la rapidez con la que los detectores de luz colectan los fotones incidentes y generan carga eléctrica para determinar los valores de pixeles, al aumentar la capacitancia la velocidad de transferencia de carga disminuye de pixel a pixel, lo cual, a su vez, provoca degradaciones en proceso de formación de imagen [7].

Por los motivos expuestos en los párrafos anteriores, surge la necesidad de un conjunto de técnicas y algoritmos de procesamiento de imágenes a nivel de software que permitan incrementar la resolución de las imágenes a pesar de las restricciones impuestas por la física de los dispositivos de adquisición de imágenes.

2.2. Definición de Súper-resolución.

Reconstrucción o restauración de imágenes por Súper-resolución es el proceso por el cual se obtiene una imagen digital de alta resolución espacial a partir de una o varias imágenes digitales de baja resolución de una determinada escena. Las imágenes de baja resolución deben presentar desplazamiento a nivel sub-píxel entre sí para que se pueda obtener una imagen de alta resolución [6].

2.3. Modelo de degradación.

Para el diseño e implementación de algoritmos de Súper-resolución de imágenes es importante tener un modelo matemático (también denominado modelo de degradación o modelo de observación) que describa el proceso de adquisición de imágenes de una cámara y relacione de forma cuantitativa la imagen de alta resolución deseada con las imágenes de baja resolución obtenidas.

La figura 2 ilustra el proceso por el que una escena (representada por una imagen de alta resolución) captada por una cámara sufre de distorsión de movimiento, solapamiento (aliasing), difuminación y ruido hasta llegar a transformarse en la imagen de baja resolución observada.

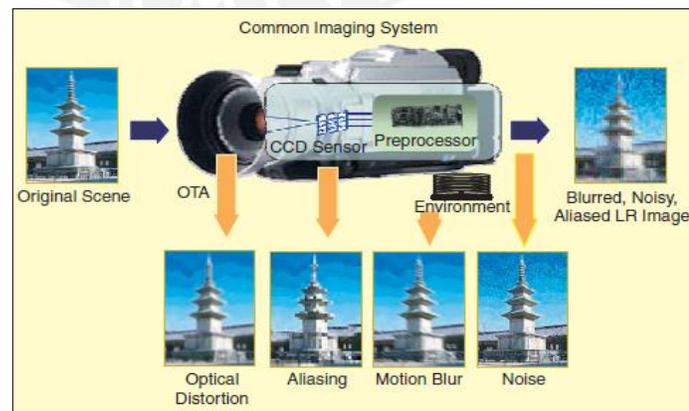


Figura 2: Esquema de degradación de una imagen digital de una escena.

[6]

En la figura 3 se muestra, mediante un diagrama de bloques, la relación entre la imagen de alta resolución deseada y las de baja resolución observadas; como se aprecia en la misma, las imágenes de baja resolución se obtienen como resultado, según el modelo de degradación, de operadores matemáticos que representan sub-muestreo, difuminación, distorsión y ruido que actúan sobre la imagen de alta resolución de una escena [6].

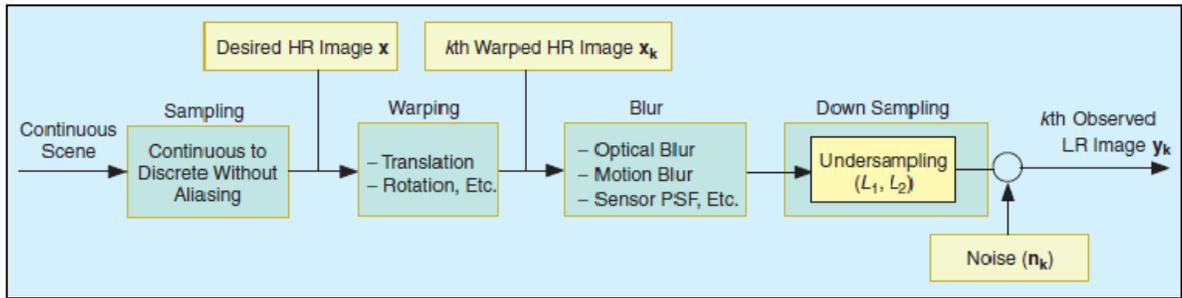


Figura 3: Diagrama de bloques del proceso de degradación de una imagen digital de alta resolución.

[6]

Teniendo en cuenta el diagrama de bloques de la figura 3, el modelo de degradación se puede expresar matemáticamente mediante la siguiente ecuación matricial [6]:

$$y_k = D B_k M_k x + n_k, \quad 1 \leq k \leq p \quad \dots(2.1)$$

En (2.1), x es la imagen de alta resolución deseada de tamaño $L_1 N_1 \times L_2 N_2$ y se expresa de forma vectorial como $x = [x_1, x_2, \dots, x_N]^T$, $N = L_1 N_1 \times L_2 N_2$; L_1 y L_2 son parámetros de sub-muestreo de la imagen de alta resolución, mientras que N_1 y N_2 son las dimensiones de la imagen deseada [6].

Además, y_k es la k -ésima imagen de baja resolución observada que se represente mediante el vector $y_k = [y_{k1}, y_{k2}, \dots, y_{kM}]^T$ donde $M = N_1 \times N_2$, mientras que p es el número total de imágenes que se disponen de una misma escena [6].

Por otro lado, D es la matriz de sub-muestreo de tamaño $(N_1 N_2)^2 \times L_1 N_1 L_2 N_2$, la cual reduce el tamaño de las imágenes que se capturan; B_k es una matriz de tamaño $L_1 N_1 \times L_2 N_2 \times L_1 N_1 L_2 N_2$ que representa la difuminación de la imagen, la cual es generada por la función de transferencia del sistema óptico, el cual es usualmente denominado PSF

(Point Spread Function) [8]. M_k es la matriz de distorsión de imagen, cuyas dimensiones son $L_2 N_2 \times L_1 N_1 L_2 N_2$, la cual representa el desplazamiento sub-píxel producido durante la adquisición de una imagen; finalmente, n_k es vector de ruido de tamaño $N_1 \times N_2$ [6].

En [9] se indica que no es una tarea práctica calcular las matrices de sub-muestreo, difuminación y de distorsión, que caracterizan al modelo de degradación, por separado, motivo por el que se suele unificar los efectos de dichas matrices en una sola matriz, de tal manera que el modelo de degradación expresado en (2.1) se puede re-escribir de mediante la siguiente ecuación matricial [6]:

$$y_k = W_k x + n_k, \quad k = 1, 2 \dots p \dots (2.2)$$

En (2.2) W_k es una matriz de tamaño $(N_1 N_2)^2 \times L_1 N_1 L_2 N_2$ que representa la difuminación, distorsión y sub-muestreo que sufre la imagen de alta resolución; a esta matriz se le suele llamar PSF combinada [8].

La PSF combinada (W_k) de una cámara es una función de muestreo bidimensional que describe la forma en la que los píxeles de alta resolución de la imagen deseada afectan a un píxel de baja resolución en un área determinada [10]. Dicha función debe depender únicamente de la distancia entre los píxeles involucrados en el muestreo, además, debe ser una función decreciente con respecto a la distancia y ser separable, es decir, $W_k(x, y) = W_k(x) W_k(y)$ [11].

La matriz W_k puede ser calculada, si es que se tiene acceso al hardware del sistema óptico de adquisición de imágenes, mediante la determinación del tamaño de punto o analizando una línea en el espacio de Fourier [8] [12]. Sin embargo, si se desconocen las características del mencionado sistema óptico, se puede emplear el método CEPSTRUM para caracterizar la PSF combinada o, también, se puede seleccionar un W_k arbitrario para el proceso de restauración pero ello implicaría la utilización de un algoritmo de Súper-resolución que sea robusto con respecto a errores de la función de muestreo con el fin de evitar la obtención de una imagen final degradada [13].

2.4. Etapas de Súper-resolución.

La mayoría de los algoritmos de Súper-Resolución emplean, por lo general, tres etapas para reconstruir la imagen, las cuales son: registro, interpolación y restauración.

En la etapa de registro se obtiene información del movimiento relativo entre las imágenes capturadas de una misma escena desde posiciones diversas o utilizando diferentes sensores. A partir de una imagen de referencia seleccionada, se estima el desplazamiento sub-píxel existente con respecto a las demás mediante la aplicación de una transformación espacial a cada imagen que no es la de referencia [6].

En la etapa de interpolación, los píxeles que han sido estimados en la etapa de registro son re-ubicados de tal manera que ocupen un lugar en la cuadrícula correspondiente a la imagen de alta resolución que se va a reconstruir [6].

Finalmente, en la etapa de restauración, se aplica un filtro para eliminar el ruido y la difuminación (o borrosidad) de la imagen; si la información de la borrosidad (datos de la PSF) no es conocida se debe incorporar en alguna etapa del proceso de reconstrucción la estimación de la misma [6].

En la figura 4 se muestran las etapas de registro, interpolación y restauración que conforman el proceso Súper-resolución.

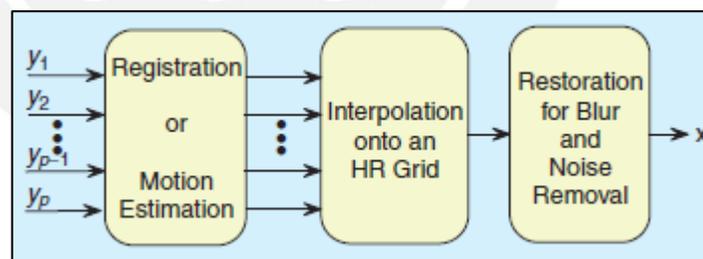


Figura 4: Etapas del proceso de reconstrucción de Súper-resolución.

[6]

Por otro lado, en la figura 5 se ilustran los procesos de registro e interpolación; en la etapa de registro, los píxeles en forma de triángulo y círculo son los que se han desplazado en relación a los píxeles de la imagen de referencia (píxeles en forma de círculo uniformemente espaciados), mientras que, en lo referente a la etapa de

interpolación, los píxeles que se han estimado en la etapa de registro tienen que ser reubicados uniformemente en la cuadrícula de la imagen de alta resolución.

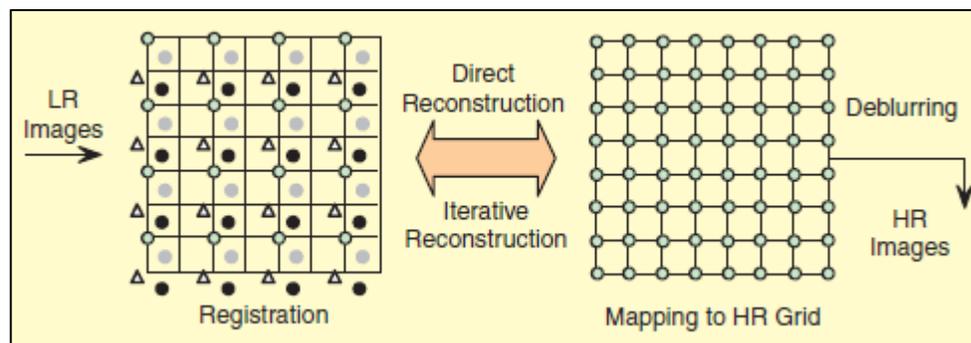


Figura 5: Etapas de registro e interpolación con los datos de desplazamiento-sub-píxel. Luego de ejecutarse ambas etapas, la imagen magnificada entra a la etapa de restauración.

[6].

2.5. Características de los algoritmos de Súper-resolución.

Ya que se ha definido el concepto de Súper-resolución y se ha expuesto el modelo de degradación de imágenes sobre el cual está basado, a continuación se realizará una descripción de los principales algoritmos de Súper-resolución con sus principales características.

2.5.1. Reconstrucción en el dominio de la frecuencia.

Súper-resolución en el dominio de la frecuencia se basa en tres premisas para lograr la reconstrucción de la imagen deseada: la propiedad de desplazamiento de la Transformada de Fourier, la relación de “aliasing” existente entre la Transformada Continua de Fourier (CFT) de la imagen de alta resolución y la Transformada Discreta de Fourier (DFT) de las imágenes de baja resolución y la asunción de que la imagen deseada se encuentra limitada en banda [14].

La ventaja de la reconstrucción en frecuencia es que no presenta mucha complejidad teórica en comparación con otros métodos de Súper-resolución pero tiene limitaciones como, por ejemplo, el hecho de que solo se admita movimiento de traslación global entre las imágenes y que es complicado aplicar métodos de regularización que utilicen información de a priori de la imagen deseada [6].

2.5.2. Algoritmos de interpolación.

Estos algoritmos, en base a la información de desplazamiento sub-píxel de la etapa de registro, llenan los píxeles en la cuadrícula de la imagen de alta resolución mediante el uso de un polinomio interpolador; dicho polinomio asigna pesos a la vecindad de cada píxel cuyo valor se desconoce; los pesos asignados se determinan en función de la distancia que separa al píxel cuyo valor se va a calcular con respecto de sus vecinos. Los algoritmos más utilizados son los siguientes: vecinos más cercanos, interpolación bilineal e interpolación bicúbica. El algoritmo de vecinos más cercanos simplemente toma en cuenta el píxel más cercano para la interpolación, mientras que el bilineal toma en cuenta los cuatro píxeles más cercanos y la interpolación bicúbica toma en cuenta los dieciséis píxeles más cercanos [15].

Estos algoritmos producen bajo costo computacional pero no incorporan información a priori de la imagen de alta resolución que se desea obtener, además, solo son útiles si las propiedades de la difuminación y del ruido son las mismas para todas las imágenes de baja resolución adquiridas por la cámara [6].

2.5.3. Algoritmos de regularización.

El modelo de degradación expuesto en el apartado 2.3 puede ser visto como el planteamiento de un problema inverso, es decir, dado los datos finales (imágenes de baja resolución) de un modelo (en este caso el de degradación de una imagen de alta resolución), se desea estimar algún parámetro (imagen de alta resolución) de dicho modelo. En algunos problemas inversos, puede ocurrir que cambios menores en los datos generan cambios significativos en la solución; a este tipo de problemas se les denomina “problemas mal planteados” y Súper-resolución, debido a la insuficiente cantidad de imágenes de baja resolución disponible, pertenece a esta categoría [6].

Por tal motivo, por la condición de problema mal planteado de Súper-resolución, existen técnicas de regularización para solucionar el inconveniente de falta de información; algunas de las mencionadas técnicas (o algoritmos) son Constrained Least Squares (CLS) y Maximum a Posteriori (MAP) [6].

CLS es una técnica determinista la cual asume información a priori de la solución mediante el uso de una ecuación cuadrática que contiene un parámetro de regularización (usualmente multiplicador de Lagrange) que controla el equilibrio entre

suavidad de la solución y fidelidad de los datos. El objetivo de CLS es calcular el valor x que minimice la ecuación mencionada. La principal ventaja de este algoritmo es la unicidad de la solución [6].

MAP es una técnica estocástica basada en el teorema de Bayes para maximizar la función densidad de probabilidad a posteriori (PDF), siempre y cuando sea posible determinarla, del vector x dado y_k . La principal ventaja de este algoritmo es la flexibilidad y robustez en el modelamiento del ruido que afecta a las imágenes.

2.5.4. Proyecciones a conjuntos convexos (POCS).

Proyecciones a conjuntos convexos o POCS (en inglés) es un algoritmo iterativo que emplea información previa de la solución al proceso de reconstrucción. Esto significa que la solución (el vector x) se restringe a un número finito de conjuntos convexos que representan la información a priori de la imagen de alta resolución que se desea obtener [6].

POCS produce buenos resultados para valores bajos y altos de magnificación (siempre y cuando las imágenes de baja resolución hayan sido adecuadamente registradas) y, además, es un método simple de programar pero produce alto costo computacional [6].

2.5.5. Retro-proyección Iterativa (IBP).

IBP consiste en reconstruir la imagen de alta resolución a partir de minimizar el error entre las estimaciones sucesivas de la imagen deseada y las imágenes de baja resolución, este método genera imágenes de baja resolución artificiales a partir de una imagen de alta resolución seleccionada como punto inicial; las imágenes artificiales varían en cada iteración según el error entre ellas y las imágenes de baja resolución adquiridas por la cámara [6].

IBP produce buenos resultados para altos valores de magnificación de imagen, por otro lado, es un método de fácil comprensión; sin embargo, para valores bajos de magnificación presenta resultados de baja calidad visual, además, es complicado hacer uso de restricciones a priori de la solución [6].

2.5.6. Populis-Gerchberg (PG).

Populis-Gerchberg es un algoritmo iterativo que realiza el proceso de reconstrucción tanto en el dominio espacial (basado en el modelo de degradación) y en el de la frecuencia mediante filtrado y truncamiento de altos valores de frecuencia hasta que se cumpla un determinado criterio de convergencia [16].

Este algoritmo no toma en cuenta los parámetros de movimiento, por lo que puede generar degradación en la imagen reconstruida [16].

2.5.7. Reconstrucción Híbrida (ML-POCS).

Este algoritmo combina las características de los algoritmos de regularización estocásticos con las de proyecciones a conjuntos convexos; consiste en minimizar la función costo Maximum a Posteriori (MAP) o Maximum Likelihood (ML) mientras se restringe la solución a ciertos conjuntos convexos y cerrados. La ventaja de este algoritmo es que todo el conocimiento previo, definido en términos de conjuntos convexos y cerrados, se combina efectivamente en una sola solución [6].

CAPÍTULO 3

IMPLEMENTACIÓN DEL ALGORITMO POCS

En el presente capítulo se exponen los alcances y objetivos del trabajo de tesis, se justifica la selección del algoritmo POCS, se dan a conocer sus características, así como también el marco teórico sobre el cual está basado y, finalmente, se presenta la implementación del mismo.

3.1. Alcance.

El presente trabajo de tesis busca mejorar la resolución de imágenes digitales de campos de cultivo mediante la aplicación de un algoritmo de Súper-resolución en el dominio espacial; el algoritmo debe tener en consideración el desplazamiento sub-píxel existente entre las imágenes de baja resolución que representan una escena de campo de cultivo y, en base a ello, obtener una imagen digital magnificada por un factor de 2.

3.2. Objetivos.

A continuación se detallan los objetivos del presente trabajo de tesis.

3.2.1. Objetivo General.

Programar un algoritmo de Súper-resolución para mejorar la resolución de imágenes de campos de cultivo.

3.2.2. Objetivos Específicos.

- Programar el algoritmo de Súper-resolución POCS.
- Comparar los resultados de POCS con los algoritmos de Súper-resolución del software MATLAB.
- Aplicar el algoritmo POCS a imágenes de campos de cultivo.

3.3. Justificación de POCS.

En [6] se determinó que el algoritmo POCS es simple de programar, incorpora todos los elementos del modelo observacional correspondiente al dominio espacial y, a su vez, tiene la capacidad de incluir información a priori de la escena (imagen de alta resolución) que se desea obtener.

Por otro lado, en [16] se realizó una evaluación del desempeño de los algoritmos de Súper-resolución POCS, interpolación bicúbica, Populis-Gerchberg, Robust Super-Resolution (RSR), IBP y Structured-Adaptive Normalize Convolution (SANC), de tal manera que se determinaron aquellos algoritmos que presentaron mejores resultados en términos de calidad visual y del indicador HCI (índice de comparación de histograma) [17]. La evaluación consistió en magnificar imágenes de baja resolución de una misma escena por diversos factores mediante la aplicación de los algoritmos que formaron parte de la prueba; luego de calcular los valores de HCI de las imágenes restauradas, se concluyó que los algoritmos SANC, POCS e interpolación bicúbica produjeron buenos resultados para distintos factores de magnificación.

La figura 6 muestra la comparación visual entre la imagen que fue objeto de pruebas y la imagen restaurada con POCS en [16], mientras que la tabla 1 muestra los resultados cuantitativos (HCI) de la aplicación de los algoritmos de Súper-resolución en [16].

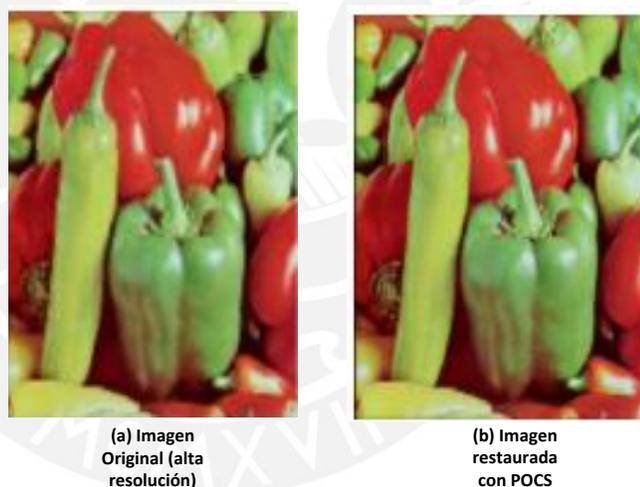


Figura 6: Comparación entre imagen original (alta resolución) de prueba y la obtenida con POCS.

[16].

Tabla 1: Resultados de la evaluación del desempeño de algoritmos de Súper-resolución en [16].

	POCS	Interpolación Bicúbica	Populis-Gerchberg	RSR	IBP	SANC
HCI	0.0688	0.0698	0.4275	0.1184	0.1343	0.0653

Obtenido de [16].

Por los motivos expuestos en los párrafos anteriores, se seleccionó el algoritmo POCS para aplicarlo a imágenes de campos de cultivo y poder determinar la factibilidad de su uso según los indicadores usados en el presente trabajo de tesis (ver apartado 4.1).

3.4. Conjuntos y funciones Convexas.

Desde 3.4 hasta 3.6 se van a exponer los fundamentos teóricos del algoritmo POCS necesarios para la programación del mismo.

3.4.1. Conjuntos convexos.

Intuitivamente, un conjunto es convexo si, al unir dos puntos cualesquiera (que pertenecen al conjunto) mediante una recta, la recta trazada está incluida dentro del conjunto [18]. La figura 7 ilustra la noción de convexidad de un conjunto.

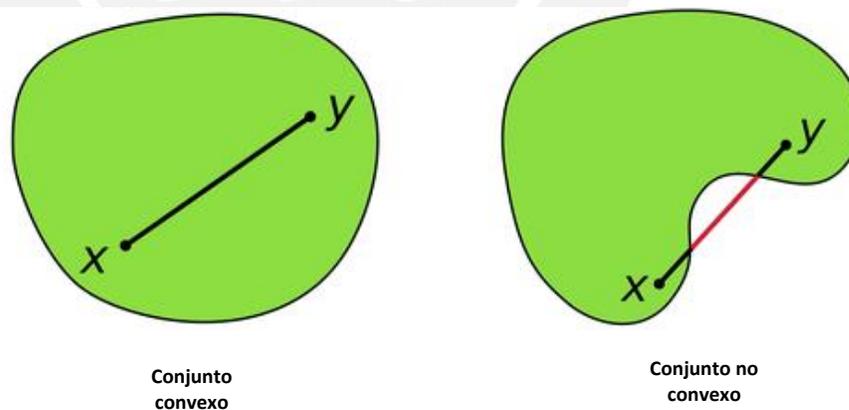


Figura 7: Noción de convexidad de conjuntos.

[19].

La definición matemática de la convexidad de un conjunto es la siguiente [18]:

Un conjunto C es convexo si $\forall x, y \in C \wedge \forall \lambda \in [0,1]$ se cumple:

$$\lambda y + (1 - \lambda)x \in C \quad \dots(3.1)$$

Adicionalmente, cabe resaltar que un conjunto convexo tiene la propiedad de ser cerrado si es que dicho conjunto incluye los límites que lo definen, caso contrario, el conjunto convexo es abierto [20].

3.4.2. Funciones convexas.

Una función es convexa si es que está definida sobre un dominio que es un conjunto convexo [18]. Matemáticamente, una función es convexa si cumple con la siguiente condición (definición de Jensen) [18]:

$$f(\lambda y + (1 - \lambda)x) \leq \lambda f(y) + (1 - \lambda)f(x), \forall x, y \in C \wedge \lambda \in [0,1] \quad \dots(3.2)$$

3.4.3. Problema de optimización convexa.

Se define un problema de optimización convexa de forma siguiente [21]:

Minimizar $f(x)$ sujeto a:

$$g_i(x) \leq 0$$

$$h_i(x) = 0$$

La variable x es la variable de optimización, $x \in R^n$.

$f(x): R^n \rightarrow R$ es la función objetivo o función costo.

$g_i(x): R^n \rightarrow R$, $i = 1, 2, \dots, m$ son las restricciones del problema en forma de desigualdades.

$h_i(x): R^n \rightarrow R$, $i = 1, 2, \dots, m$ son las restricciones del problema en forma de igualdades.

Las funciones h_i y g_i deben ser convexas, además, al menor valor de x que satisface todas las restricciones del problema se le denomina valor óptimo [21].

3.5. Proyecciones hacia conjuntos convexos (POCS).

A continuación, se expondrán definiciones preliminares y las bases conceptuales del algoritmo POCS, las cuales sirven para la programación y aplicación al contexto Súper-resolución.

3.5.1. Norma vectorial euclidiana.

La norma vectorial $\|x\|$ es una abstracción o generalización de distancia en un espacio n -dimensional [22], además, existen diversos tipos de normas según el espacio vectorial con el que se trabaje, sin embargo, todas las normas vectoriales deben presentar las siguientes propiedades [22]:

$$x \neq 0 \rightarrow \|x\| > 0 \quad \dots(3.3)$$

$$\|\alpha x\| = |\alpha| \|x\|, \alpha \in R \quad \dots(3.4)$$

$$\|x + y\| \leq \|x\| + \|y\|, \forall x \in R^n \quad \dots(3.5)$$

La norma vectorial euclidiana cumple con (3.3), (3.4), (3.5) y se define según la siguiente fórmula [19]:

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2} \quad \dots(3.6)$$

La especificación de una norma o “distancia” es necesaria para los conceptos que se expondrán en los apartados siguientes, los cuales conforman la base del algoritmo POCS, además, cabe indicar que en el presente trabajo de tesis se seleccionó la norma vectorial euclidiana, ya que es la más utilizada en los trabajos de investigación concernientes al algoritmo en cuestión [6] [13] [18].

3.5.2. Teorema de la proyección.

El teorema de la proyección se define de la siguiente manera:

Dado un sub-espacio C incluido en el espacio de Hilbert (es un espacio vectorial con producto interno completo respecto a la norma vectorial definida por el producto interno; el espacio de Hilbert puede ser concebido como la generalización de operaciones y propiedades que se emplean en espacios bidimensionales y tridimensionales –espacios Euclidianos- a un espacio de dimensión mayor o infinita

inclusive, dichas operaciones y propiedades incluyen proyecciones, distancias, convergencia de sucesiones, cálculo de ángulos, ortogonalidad, etc. [23]) y un vector x perteneciente al espacio de Hilbert, existe un único vector \hat{x} perteneciente a C (denominado proyección de x a C) tal que [24]:

$$\|x - \hat{x}\| = \min_{y \in C} \|x - y\| \quad \dots(3.7)$$

La ecuación (3.7) representa una minimización de distancia (del vector x al sub-espacio C), donde y es un vector que pertenece al conjunto C y $\|\cdot\|$ es la norma euclidiana, por otro lado, si el sub-espacio C es un conjunto convexo cerrado (una restricción) y el vector x es un punto exterior a C , entonces el vector \hat{x} es el punto más cercano a x que cumple con la restricción impuesta, es decir, \hat{x} es la solución al problema de minimización planteado en (3.7) [25]. La figura 8 ilustra el teorema de la proyección en dos dimensiones.

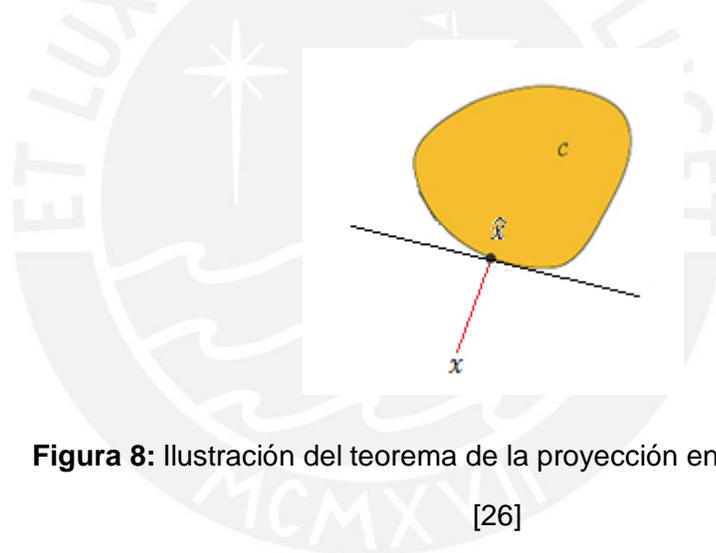


Figura 8: Ilustración del teorema de la proyección en dos dimensiones.

[26]

Para un problema de optimización convexa con más de una restricción, es necesario recurrir a una herramienta matemática que, basada en el teorema de la proyección, permita realizar proyecciones sucesivas del vector x para determinar el valor de \hat{x} que cumpla con todas las restricciones impuestas.

3.5.3. Algoritmo de las proyecciones alternantes.

Con el algoritmo de las proyecciones alternantes se demostrará que es matemáticamente posible aplicar iterativamente proyecciones a cada conjunto convexo (restricción) para hallar una solución que cumpla con todas las condiciones del problema de optimización.

3.5.3.1. Definición del algoritmo de las proyecciones alternantes.

El algoritmo de las proyecciones alternantes permite encontrar un punto en la intersección de una serie de conjuntos convexos mediante la aplicación sucesiva de proyecciones a cada conjunto [25]. Dichas proyecciones se llevan a cabo mediante proyectores, los cuales son fórmulas analíticas que resultan de la minimización de (3.7) [25].

3.5.3.2. Proyecciones alternantes para dos conjuntos convexos.

Dados dos conjuntos convexos y cerrados C y D , se denotan P_C y P_D a sus proyectores y x, y a los puntos pertenecientes a C y a D respectivamente. El algoritmo comienza con cualquier punto inicial x_1 perteneciente a C ; este punto se proyecta hacia el conjunto D mediante el proyector P_D , de tal modo que se obtiene el punto y_1 , el cual, a su vez, es proyectado hacia el conjunto C con la aplicación de P_C . Este proceso se repite de manera alternada un número finito de veces hasta aproximarse a un punto en la intersección de ambos conjuntos [25] (figura 9).

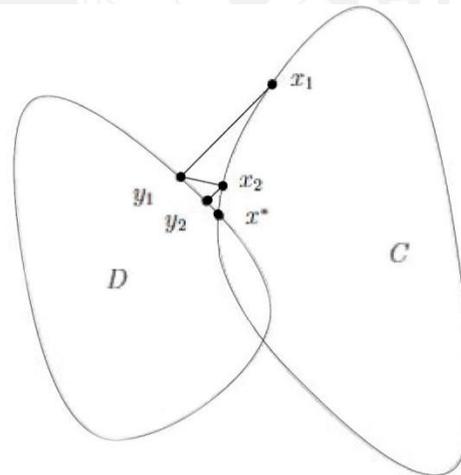


Figura 9: Aplicación de las proyecciones alternantes a dos conjuntos convexos C y D , x^* es el punto en la intersección de C y D , que el algoritmo ha encontrado.

[25].

La expresión matemática del algoritmo es la siguiente:

$$y_k = P_D(x_k), \quad x_{k+1} = P_C(y_k) \quad \dots(3.8)$$

En (3.8) k representa el número de iteración.

En este apartado se ha definido el algoritmo para el caso de dos conjuntos convexos y cerrados, sin embargo, se puede generalizar y aplicar el mismo a una cantidad finita de conjuntos [25].

3.5.3.3. Propiedades del algoritmo de las proyecciones alternantes.

El algoritmo de las proyecciones alternantes presenta las siguientes propiedades:

- Es convergente, es decir, en cada iteración, el valor proyectado a cada conjunto se encuentra más cerca de la intersección de todos los conjuntos convexos y cerrados [27].
- Permite encontrar un punto en la intersección de los conjuntos convexos y cerrados, por lo que la solución no es única [27].
- El algoritmo no necesariamente encuentra el punto deseado (solución) en una cantidad finita de iteraciones [25].
- A mayor número de iteraciones, el resultado final obtenido es mejor, ya que se aproxima más a la intersección de todos los conjuntos convexos y cerrados [25].
- Si la intersección entre dos conjuntos convexos es nula ($C \cap D = \emptyset$), el algoritmo también es útil, ya que converge, por separado, a los puntos x_k e y_k tales que la distancia entre ellos es mínima (ver figura 10) [27].

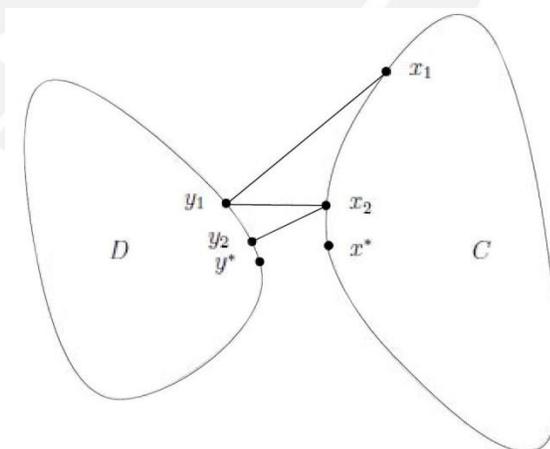


Figura 10: Aplicación de proyecciones alternantes a dos conjuntos cuya intersección es nula.

[25]

3.6. Programación del algoritmo POCS aplicado a Súper-resolución.

En 3.3.1, 3.3.2 y 3.3.3 se expusieron algunos aspectos del marco teórico de POCS y de sus propiedades, sin embargo, para poder aplicarlo, es necesario plantear el problema de súper-resolución en términos de la teoría de optimización convexa.

3.6.1. Súper-resolución con optimización convexa.

En 2.3 se obtuvo la expresión matemática que representa el modelo de degradación de una imagen digital, sin embargo, dicha expresión se re-define de la siguiente manera [12]:

$$g_k(x, y) = \sum_{m,n} W_k(x, y; m, n) f(m, n) + \epsilon_k(x, y) \quad \dots(3.9)$$

En (3.9) g_k es la imagen de baja resolución observada número k , W_k representa la PSF combinada, ϵ_k es el ruido observado para la imagen k y $f(m, n)$ la imagen digital de alta resolución deseada.

El término $f(m, n)$ representa, para el problema de optimización convexa, la variable de optimización, mientras que la solución óptima que se desea obtener es la imagen magnificada y restaurada $\hat{f}(m, n)$, por otro lado, las características o conocimiento a priori de la imagen $f(m, n)$ se deben definir en términos de conjuntos convexos cerrados.

3.6.1.1. Restricciones de la imagen deseada.

Las restricciones de la imagen deseada, las cuales representan información que se conoce previamente de la solución $f(m, n)$, son conjuntos convexos cerrados, los cuales se definen en [12], [28] y [29] del siguiente modo:

$$C_{x,y;i,k} = \{ f(m, n)_i : |r_k^{fi}(x, y)| \leq \varphi_o \} \quad \dots(3.10)$$

En (3.10) al término r_k^{xj} se le denomina residuo y se define de la siguiente manera [9]:

$$r_k^{fi}(x, y) = g_k(x, y) - \sum_{m,n} W_k(x, y; m, n) f(m, n) \quad \dots(3.11)$$

El término φ_o es la restricción de magnitud impuesta al residuo y significa que, el valor de cada pixel de cada imagen de baja resolución adquirida, no debe exceder en una cantidad φ_o a los valores de los pixeles obtenidos del proceso de filtrar la imagen

deseada $f(m, n)$ con la PSF [12]. A esta restricción se la denomina la restricción de los datos, ya que se definió en base a las imágenes adquiridas por la cámara.

La convexidad de $C_{x,y;i,k}$ se demuestra utilizando la definición de conjunto convexo en el apartado 3.4.1 pero primero, por simplicidad de notación para la demostración, se expresará la ecuación 3.11 en forma vectorial, para lo cual se reemplaza G_k en lugar de $g_k(x, y)$, W_k en lugar de $\sum_{m,n} W_k(x, y; m, n)$ y F en lugar de $f(m, n)$:

$$|G_k - W_k F| \leq \varphi_0$$

Sean los puntos $F_1, F_2 \in C_{x,y;i,k}$

Dado que F_1, F_2 son puntos cualesquiera dentro del conjunto $C_{x,y;i,k}$, se define una recta genérica $F_3 = \lambda F_1 + (1 - \lambda)F_2$, $F_3 \in C_{x,y;i,k}$, además, $\lambda \in [0,1]$

Luego,

$$|G_k - W_k F_3| = |G_k - W_k (\lambda F_1 + (1 - \lambda)F_2)|$$

$$|G_k - W_k F_3| = |G_k - \lambda W_k F_1 - W_k(1 - \lambda)F_2|$$

$$|G_k - W_k F_3| = |(\lambda G_k - \lambda W_k F_1) - \lambda G_k + (1 - \lambda)G_k - W_k(1 - \lambda)F_2 - (1 - \lambda)G_k + G_k|$$

Acomodando términos convenientemente:

$$|G_k - W_k F_3| = |\lambda(G_k - W_k F_1) + (1 - \lambda)(G_k - W_k F_2) - \lambda G_k - G_k + \lambda G_k + G_k|$$

Al simplificar los últimos cuatro términos resulta la siguiente expresión:

$$|G_k - W_k F_3| = |\lambda(G_k - W_k F_1) + (1 - \lambda)(G_k - W_k F_2)|$$

Aplicando la propiedad de desigualdad triangular a la expresión resulta:

$$|G_k - W_k F_3| \leq \lambda|(G_k - W_k F_1)| + (1 - \lambda)|(G_k - W_k F_2)|$$

Dado que F_1, F_2 pertenecen a $C_{x,y;i,k}$ se cumple las siguientes desigualdades:

$$|G_k - W_k F_1| \leq \varphi_0$$

$$|G_k - W_k F_2| \leq \varphi_0$$

Multiplicando por λ y $(1 - \lambda)$ respectivamente se obtienen las siguientes expresiones:

$$\lambda |G_k - W_k F_1| \leq \lambda \varphi_o$$

$$(1 - \lambda) |G_k - W_k F_2| \leq (1 - \lambda) \varphi_o$$

Sumando ambas expresiones se obtiene:

$$\lambda |G_k - W_k F_1| + (1 - \lambda) |G_k - W_k F_2| \leq \lambda \varphi_o + (1 - \lambda) \varphi_o$$

Reduciendo una vez más:

$$\lambda |G_k - W_k F_1| + (1 - \lambda) |G_k - W_k F_2| \leq \varphi_o$$

Por propiedad de transitividad se obtiene:

$$|G_k - W_k F_3| \leq \varphi_o, \forall \lambda \in [0,1]$$

Por lo tanto, se concluye que la recta F_3 se encuentra dentro de $C_{x,y;i,k}$ para dos puntos cualesquiera F_1, F_2 y, por ende, $C_{x,y;i,k}$ es un conjunto convexo.

Por otro lado, se define la restricción de amplitud de pixeles mediante la siguiente expresión [24]:

$$C_A = \{f(m, n) : \alpha \leq f(m, n) \leq \beta, \beta > \alpha\} \dots(3.12)$$

Esta restricción se impone a cada pixel de la imagen de alta resolución deseada según los valores α y β , los cuales deben conocerse previamente.

Para demostrar la convexidad de este conjunto se realizan los siguientes pasos algebraicos:

Sean los puntos $F_1, F_2 \in C_A$

Dado que F_1, F_2 son puntos cualesquiera dentro del conjunto C_A , se define una recta genérica $F_3 = \lambda F_1 + (1 - \lambda) F_2$, $F_3 \in C_A$, además, $\lambda \in [0,1]$

Al pertenecer F_1 a C_A , se cumple la siguiente desigualdad:

$$\alpha \leq F_1 \leq \beta$$

Al multiplicar por λ se obtiene:

$$\lambda \alpha \leq \lambda F_1 \leq \lambda \beta$$

Se procede de forma similar para F_2 pero se multiplica por $(1 - \lambda)$:

$$(1 - \lambda) \alpha \leq (1 - \lambda) F_2 \leq (1 - \lambda) \beta$$

Al sumar cada lado en ambas expresiones se obtiene:

$$\lambda \alpha + (1 - \lambda)\alpha \leq \lambda F_1 + (1 - \lambda)F_2 \leq \lambda \beta + (1 - \lambda) \beta$$

Simplificando términos y reemplazando se obtiene:

$$\alpha \leq \lambda F_1 + (1 - \lambda)F_2 \leq \beta$$

Esta expresión es igual a:

$$\alpha \leq F_3 \leq \beta, \forall \lambda \in [0,1]$$

Por lo tanto, se concluye que la recta F_3 se encuentra dentro de C_A para dos puntos cualesquiera F_1, F_2 y, por ende, C_A es un conjunto convexo.

3.6.1.2. Función Objetivo.

La función a minimizar en este problema de optimización es la distancia, o norma euclidiana, entre la imagen deseada $f(m, n)$ y los conjuntos convexos $C_{x,y;l,k}$ y C_A presentados en 3.4.1.1.

Finalmente, el problema de Súper-resolución, en base a las restricciones obtenidas y a la variable de optimización establecida, resulta expresado de la siguiente forma [29]:

$$\text{Encontrar } \hat{f} \in C = \bigcap_{1 \leq i \leq n} C_i \quad \dots \quad (3.13)$$

En (3.13) C_i son las restricciones $C_{x,y;l,k}$ y C_A .

3.6.1.3. Determinación de los proyectores.

Dado que se conocen los elementos del problema que se va a optimizar, es necesario obtener fórmulas analíticas que ejecuten las proyecciones de la imagen deseada f hacia los conjuntos convexos $C_{x,y;l,k}$ y C_A .

En [29] se aplicó el método de los multiplicadores de LaGrange para minimizar la distancia entre $C_{x,y;l,k}$ y f . El resultado de la minimización es el proyector $P_{x,y;l,k}$, el cual está representado según la siguiente fórmula:

$$P_{x,y;i,k}(f) = \begin{cases} f(m, n) + \frac{(r_k^{f_i(x,y)} - \varphi_0) W_k(x,y;m,n)}{\sum_{m,n} W_k^2(x,y;m,n)}, & r_k^{f_i(x,y)} > \varphi_0 \\ f(m, n), & -\varphi_0 < r_k^{f_i(x,y)} < \varphi_0 \\ f(m, n) + \frac{(r_k^{f_i(x,y)} + \varphi_0) W_k(x,y;m,n)}{\sum_{m,n} W_k^2(x,y;m,n)}, & r_k^{f_i(x,y)} < -\varphi_0 \end{cases} \quad \dots(3.14)$$

En (3.14), el término $W_k(x, y; m, n)$, que corresponde a la PSF combinada, va a permitir llenar pixeles en la imagen $f(m, n)$ según el valor del residuo calculado para cada pixel de cada imagen de baja resolución y teniendo en consideración el valor de restricción φ_0 para corregir el valor del residuo obtenido, de tal modo que se encuentre dentro de los límites permitidos por φ_0 . Además, si el valor del residuo se encuentra dentro del intervalo $< -\varphi_0, \varphi_0 >$, no se utiliza $W_k(x, y; m, n)$ para llenar pixeles en $f(m, n)$, ya que el residuo se encuentra dentro de los límites permitidos por φ_0 .

De forma similar, en [29] se obtuvo la siguiente expresión para el proyector de la restricción de amplitud de pixeles:

$$C_A: P_A(f) = \begin{cases} \alpha, & f(m, n) < \alpha \\ f(m, n), & \alpha \leq f(m, n) \leq \beta \\ \beta, & f(m, n) > \beta \end{cases} \quad \dots (3.15)$$

En (3.15), α y β son valores de pixel mínimo y máximo que puede tener la imagen digital de alta resolución $f(m, n)$, si un pixel de $f(m, n)$ es menor que el límite α , se corrige dicho valor de pixel reemplazándolo por α , si un pixel de $f(m, n)$ es mayor que el límite β , se corrige dicho valor de pixel reemplazándolo por β , finalmente, todos los pixeles que se encuentran dentro del intervalo $[\alpha, \beta]$, se dejan sin modificarlos.

3.7. Programación de POCS.

Dado que se determinaron los proyectores que van a generar, en cada iteración, una imagen digital de alta resolución, se procede a programar el algoritmo POCS.

Se tomaron en cuenta las siguientes consideraciones para la programación del algoritmo:

- Se seleccionó un PSF gaussiano (de dimensión 5 pixeles x 5 pixeles y desviación estándar y 1) para el llenado de pixeles de la imagen de alta resolución, ya que en [12] se estipula que las PSF de las cámaras digitales exhiben este comportamiento.

- El parámetro de desviación estándar de la PSF controla el grado de difuminación de las imágenes y en el caso del algoritmo POCS afecta al tamaño de los conjuntos convexos $C_{x,y,i,k}$, es decir, a mayor desviación estándar, mayor es la cantidad de posibles soluciones que tendrá la imagen deseada [12]. Por tal motivo, la desviación estándar de la PSF suele ser la misma que la del ruido presente en las imágenes de baja resolución, lo cual conlleva a determinar la naturaleza del ruido presente en el sistema. Sin embargo, si se desconocen las características del mismo o si no es significativo, se suele emplear desviación estándar de valor 1 [13].
- El punto inicial del algoritmo representa una imagen digital de alta resolución, sin embargo, dado que dicho punto inicial va a modificarse en cada iteración, los valores iniciales de sus pixeles pueden ser arbitrarios. Por ejemplo, el punto inicial puede ser una matriz de valores ceros o puede ser una de las imágenes de baja resolución sobre-muestreada por un factor k (valor de magnificación que se desea obtener de la imagen final).

Por otro lado, en el presente trabajo de tesis se programó una función que ejecuta el algoritmo POCS, dicha función tiene la siguiente sintaxis:

$$[y, E] = pocs(im, im_ref, X, registro, k, iteraciones) \dots(3.16)$$

En la tabla 2 se describen los parámetros de entrada de la función “pocs” y en la tabla 3 se describen sus parámetros de salida.

Tabla 2: Descripción de los parámetros de entrada de la función POCS.

Parámetros de entrada	Descripción
im	Arreglo de imágenes digitales de baja resolución que se van a procesar.
im_ref	Imagen digital de alta resolución de referencia para cálculo de error, im_ref es un parámetro opcional.
X	Punto inicial del algoritmo, representa una imagen digital de alta resolución; si no se especifica una, la función asigna por defecto una de las imágenes de baja resolución.
registro	Matriz de dimensión $n \times 2$ que contiene información del desplazamiento sub-píxel horizontal y vertical entre las imágenes de baja resolución, "n" es el número de imágenes de baja resolución con las que se cuenta.
k	Factor de magnificación.
iteraciones	Número de iteraciones que ejecutará el algoritmo POCS.

Elaboración propia.

Tabla 3: Descripción de los parámetros de salida de la función POCS.

Parámetros de salida	Descripción
y	Imagen digital de alta resolución magnificada k veces respecto de la dimensión de las imágenes digitales de baja resolución de entrada.
E	Vector que contiene el error de iteración entre la imagen que se restaura iterativamente (hasta obtener "y") y la imagen digital de referencia im_ref. Si no se especifica una imagen de referencia, el error se calcula con respecto a la iteración anterior.

Elaboración propia.

En la figura 11 se muestra el diagrama de flujo de la función POCS y en la figura 12 se muestra un gráfico que ilustra el proceso de obtención de la imagen digital de alta resolución deseada con el algoritmo POCS.

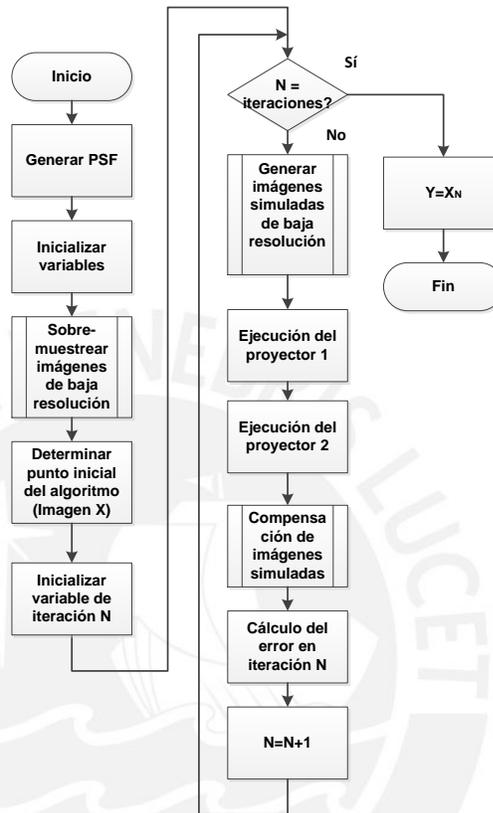


Figura 11: Diagrama de flujo de la función que ejecuta el algoritmo POCS.

En la figura 11, el bloque de generación de imágenes de baja resolución simuladas obtiene imágenes cuyos valores de píxeles se utilizan para calcular los residuos, dichos residuos entran a los bloques de ejecución de los proyectores 1 y 2 ($P_{x,y;i,k}(f)$ y $P_A(f)$ respectivamente); luego se actualizan todas las imágenes simuladas con los valores de píxeles calculados con los proyectores y , finalmente, el bloque de compensación de imágenes simuladas ensambla todas las imágenes simuladas actualizadas y genera $\hat{f}_N(m, n)$, el cual es la imagen deseada en la iteración N.

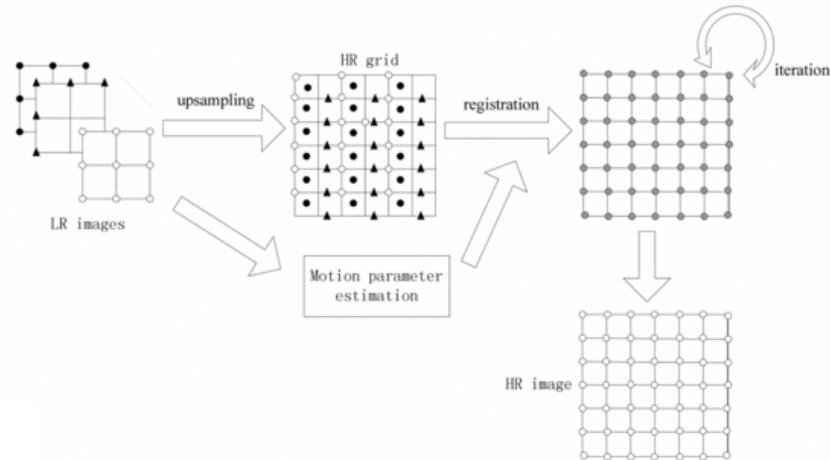


Figura 12: Aplicación de POCS a las imágenes de baja resolución.

[31]

A continuación se presentan bloques de código de la programación la función POCS con el software MATLAB:

1. Generar PSF:

```
psf=fspecial('gauss',5,1);
```

El valor de PSF se obtuvo mediante la función “fspecial”, la cual devuelve una matriz de dimensión 5 x 5, cuyos valores de elementos corresponden a una distribución gaussiana de desviación estándar 1 (ver figura 13).

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0133
0.0030	0.0133	0.0219	0.0133	0.0030

Figura 13: Valores de los elementos de la matriz que caracteriza a la PSF generada.

2. Inicialización de variables:

```
alfa=0;
beta=255;
phi=1;
```

3. Sobre-muestreo de imágenes:

```
ISA=sobre_muestrear_imagenes(im,k);
```

Se creó y se utilizó la función “sobre_muestrear_imagenes” para obtener las imágenes sobre-muestreadas que procesará el algoritmo, “im” es un arreglo que contiene las imágenes digitales de baja resolución y “k” es el factor de magnificación.

4. Determinación del punto inicial del algoritmo:

```
if(size(X)~=size(ISA{1}))
    X=zeros(size(ISA{1}));
    for i=length(im):-1:1
        coordenadas=find(ISA{i}>-1);
        X(coordenadas)=ISA{i}(coordenadas);
    end
end
X(X== -1) = 0;
```

Se determinó el punto inicial del algoritmo en función al parámetro de entrada “X” (punto inicial).

5. Inicialización de variable de iteración, inicio de iteraciones y generación de imágenes simuladas:

```
for n=1:iteraciones

    for i=1:length(imagen_simulada)
        imagen_simulada(i)=desplazamiento_subpixel(X,-k*registro(i,2),-k*registro(i,1),0);
        imagen_simulada(i)=imfilter(imagen_simulada(i),psf);
        imagen_simulada(i)=padarray(imagen_simulada(i),[4 4],0,'both');
    end
end
```

Se generan las imágenes de baja resolución simuladas con los datos obtenidos del desplazamiento sub-píxel (parámetro registro) y filtrando la imagen inicial X con el valor de PSF.

6. Ejecución del proyector 1:

```

for LR=1:length(im)

    for i=1:dim1(1)
        for j=1:dim1(2)
            if(ISA(LR)(i,j)==-1)
                continue
            end

            r=ISA(LR)(i,j)-imagen_simulada(LR)(i+4,j+4);
            if(r>phi)
                imagen_simulada(LR)(i-2+4:i+2+4,j-2+4:j+2+4)=imagen_simulada(LR)(i-2+4:i+2+4,j-2+4:j+2+4)+(r-phi).*psf./(sum(sum(psf.*psf)));

            elseif(r<-phi)
                imagen_simulada(LR)(i-2+4:i+2+4,j-2+4:j+2+4)=imagen_simulada(LR)(i-2+4:i+2+4,j-2+4:j+2+4)+(r+phi).*psf./(sum(sum(psf.*psf)));
            end
        end
    end

    imagen_simulada(LR)= imagen_simulada(LR)(5:end-4,5:end-4);
end
  
```

7. Ejecución del proyector 2:

```

X(X<alfa) = alfa;
X(X>beta) = beta;
  
```

Se aplica a cada pixel de la imagen "X" que se obtiene en cada iteración.

8. Compensación de las imágenes simuladas actualizadas:

```

imagen_simulada=alineal_imagenes(imagen_simulada,k*registro);
for i=length(imagen_simulada):-1:1
    coordenadas=find(imagen_simulada{i}~-1);
    X(coordenadas)=imagen_simulada{i}(coordenadas);
end
  
```

Se superponen las imágenes simuladas de baja resolución unas con otras, de tal modo que se ensambla una nueva versión (mejorada) de la imagen X, el desplazamiento sub-píxel existente entre ellas permite generar una imagen de alta resolución en cada iteración.

9. Cálculo del error de iteración:

```

dim2=size(X);
error= norm(reshape(im_ref,dim2(1)*dim2(2),1)-reshape(X,dim2(1)*dim2(2),1))./norm(reshape(im_ref,dim2(1)*dim2(2),1));
E(n)= error;
if (bandera==1)
    im_ref=X;
end

```

Se calcula el error de iteración, el cual se define mediante la siguiente expresión:

$$e_k \triangleq 100 \frac{\|im_ref - X_k\|}{\|im_ref\|} \quad \dots(3.17)$$

Donde *im_ref* es la imagen de referencia para calcular el error y *X_k* es la imagen de alta resolución que se mejora en cada iteración, ambos términos son parámetros de entrada de la función “pocs”.

10. Fin de las iteraciones y presentación de la imagen de alta resolución deseada:

```

end
y=X;
end

```

En anexo 1 se puede encontrar el código de la función pocs.

CAPÍTULO 4

RESULTADOS

En el presente capítulo se presentan las condiciones sobre las que se realizaron las pruebas de funcionamiento del algoritmo programado, se describen los indicadores cuantitativos seleccionados para evaluar el desempeño del mismo, se realizó la prueba del funcionamiento de POCS, se comparó el funcionamiento con los algoritmos del software MATLAB y, finalmente, se muestran las imágenes restauradas de campos de cultivo luego de aplicarse el algoritmo POCS.

4.1. Condiciones de las pruebas.

Las pruebas del algoritmo POCS se realizaron bajo las siguientes condiciones:

- Se consideraron desplazamientos sub-píxel verticales y horizontales en la etapa de registro.
- Todas las imágenes están representadas en formato de escala de grises.
- Se obtuvieron 4 escenas de campos de cultivo diferentes, de las cuales, dos de ellas contienen 4 imágenes con desplazamiento sub-píxel (figura 22).
- En algunas pruebas se generaron imágenes sintéticas con desplazamiento sub-píxel entre sí a partir de una imagen de alta definición o “ground truth”. Cabe indicar que para la generación de desplazamiento sub-píxel se utilizó el método de interpolación bicúbica, además, los valores de desplazamiento, por cada imagen sintética, oscilan entre -10 y 10 píxeles de distancia, los cuales se generaron aleatoriamente mediante una distribución uniforme.
- Se cuenta con una imagen digital genérica (no es de campo de cultivo) denominada “Lena”. Dicha imagen se utilizó como referencia (alta resolución) para las pruebas.
- Se denominó “ g ” a la imagen digital de referencia (alta resolución) y “ f ” a la imagen digital restaurada (deseada) que es el resultado final de aplicar un algoritmo de magnificación y restauración, además, debe ser lo más parecida (según los indicadores) a “ g ”.
- Las pruebas se realizaron con el software Matlab 2013a en una plataforma de hardware de 64 bits, 4GB de RAM y procesadores de múltiples núcleos.

- Los algoritmos de magnificación y restauración del software MATLAB con los que se comparó POCS son los siguientes: interpolación bilineal, interpolación bicúbica, deconvolución ciega, deconvolución Lucy-Richardson y deconvolución filtro regularizado.

4.2. Indicadores.

Para probar el funcionamiento del algoritmo POCS, según los objetivos del presente trabajo de tesis, se aplicaron los siguientes indicadores cuantitativos a las imágenes obtenidas:

4.2.1. Porcentaje de error (e_k).

Para algoritmos iterativos de súper-resolución, se define el porcentaje de error en la iteración k (e_k) según la siguiente fórmula matemática [32]:

$$e_k \triangleq 100 \frac{\|g - f_k\|}{\|g\|} \quad \dots(4.1)$$

En (4.1) f_k es la imagen deseada en la iteración k , mientras que g es la imagen de referencia y $\|\cdot\|$ indica norma vectorial euclidiana (también denominada norma vectorial L2).

4.2.2. Error cuadrático medio (ECM).

En [33] se define el error cuadrático medio (ECM) como el cuadrado de la diferencia entre la imagen de referencia g y la imagen restaurada (deseada) f . Esta operación se realiza pixel por pixel, se acumula mediante una sumatoria y se divide por la cuenta total de pixeles que corresponde a la imagen f .

La expresión matemática es la siguiente:

$$ECM = \frac{1}{M} \sum_{i=0}^M (g_i - f_i)^2 \quad \dots(4.2)$$

En (4.2) el valor M corresponde a la cantidad total de pixeles de la imagen restaurada f , mientras que g es la imagen de referencia (aquella que se conoce previamente y se asume que es de alta resolución).

4.2.3. Peak signal-to-noise ratio (PSNR).

En [34] y [35] se define PSNR como la relación entre el máximo valor posible de la imagen de referencia g y la raíz cuadrada del ECM (Error Cuadrático Medio) de la imagen restaurada f . El indicador PSNR se expresa en escala logarítmica (decibeles).

La expresión matemática es la siguiente:

$$PSNR = 20 \log_{10} \left(\frac{MAX(g)}{\sqrt{ECM}} \right) \quad \dots(4.3)$$

4.2.4. Histograma.

El histograma de una imagen digital es una función discreta que cuantifica el número de píxeles de cada valor de intensidad luminosa presente en la misma. Los histogramas de imágenes digitales proveen información estadística relevante (probabilidad de ocurrencia de nivel de intensidad en una imagen) [36].

La expresión matemática es la siguiente:

$$h(r_k) = n_k \quad \dots(4.4)$$

En (4.4) r_k representa el valor de intensidad luminosa número k y n_k es la cantidad de píxeles en la imagen que tienen valor de intensidad r_k .

Con el indicador de porcentaje de error se demostrará que el algoritmo POCS aproxima la imagen deseada f , en cada iteración, a un resultado que se ajusta a las restricciones programadas; una curva de porcentaje de error versus número de iteración (como en [32] y [37]) es útil para ilustrar el mencionado proceso.

El indicador ECM (Error Cuadrático Medio) es una medida de cuán parecidas son dos imágenes digitales; a menor valor de ECM, mayor es el parecido entre ambas. El PSNR también es un indicador del parecido entre dos imágenes digitales, sin embargo, a diferencia del ECM, a mayor valor (decibeles) de PSNR, mayor es el parecido entre ambas imágenes digitales.

El histograma no es una métrica, sin embargo, presenta información de la distribución de intensidades luminosas en toda la imagen digital; en las pruebas se generaron

gráficos de histograma para comparar las características de la imagen de referencia con la obtenida mediante el algoritmo POCS.

4.3. Prueba del funcionamiento de POCS.

En esta prueba se aplicó el algoritmo POCS para demostrar lo siguiente:

Restauración iterativa de un conjunto de 4 imágenes digitales de baja resolución que representan una misma escena (Lena) para obtener una imagen de alta resolución.

Magnificación por factor de 2 de las imágenes digitales de baja resolución de la escena que representan (Lena).

Se utilizó la imagen digital de alta resolución Lena para obtener cuatro imágenes de baja resolución, con desplazamiento sub-píxel entre sí, difuminadas y de menor tamaño, de tal modo que se aplicó el algoritmo POCS sobre dichas imágenes y se comparó el resultado final con Lena mediante el ECM y el indicador PSNR. La figura 14 ilustra el proceso de generación de las imágenes degradadas de Lena y la aplicación de POCS sobre ellas.

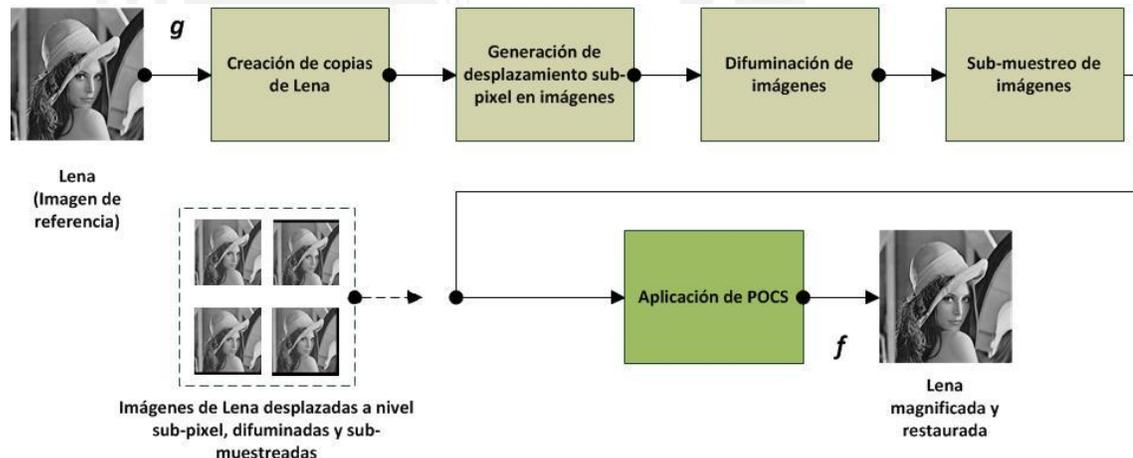


Figura 14: Generación de imágenes degradadas de Lena y aplicación de POCS.

Por otro lado, se utilizó la curva de porcentaje de error para demostrar que el algoritmo obtuvo una imagen magnificada y restaurada en cada iteración; en la figura 15 se muestra la curva de porcentaje de error partiendo desde una imagen inicial cero (todos los píxeles tienen valor de intensidad cero en la primera iteración) y en la figura 16 se muestra la curva de porcentaje de error partiendo de una imagen inicial igual a una de las imágenes de baja resolución. Las figuras 17 y 18 muestran los histogramas

correspondientes a la imagen de Lena y la imagen restaurada respectivamente, además, se puede apreciar el resultado cualitativo de la magnificación y la restauración en las figuras 19,20y 21.

Finalmente, se obtuvo el tamaño de la imagen magnificada y restaurada, de tal modo que se comparó con el tamaño de la imagen de baja resolución para comprobar que el algoritmo magnifica según el factor especificado (factor 2); la tabla 4 muestra los indicadores ECM y PSNR.

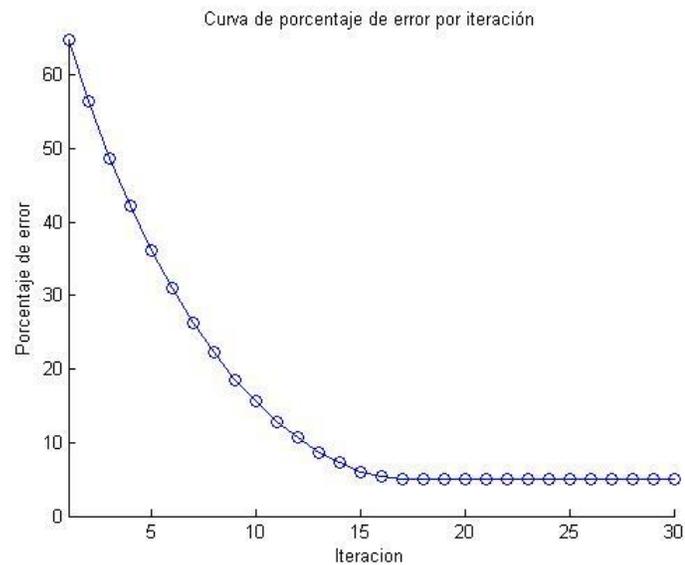


Figura 15: Curva de porcentaje de error por iteración con imagen inicial cero.

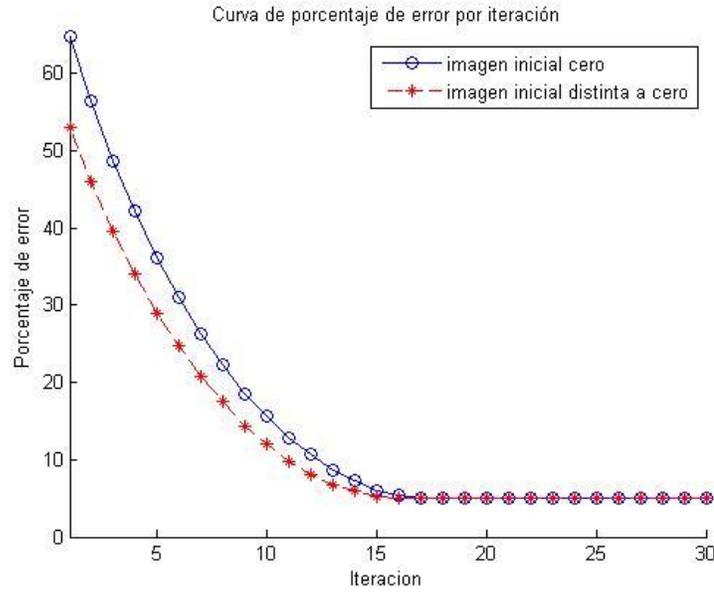


Figura 16: Comparación entre la curva de porcentaje de error por iteración con imagen inicial diferente de cero y diferente de cero.

De las figuras 15 y 16 se demuestra que el algoritmo POCS, en cada iteración genera una imagen más parecida a la de referencia, la cual se conoce de antemano que es de alta resolución y es a la que POCS debe llegar obtener.

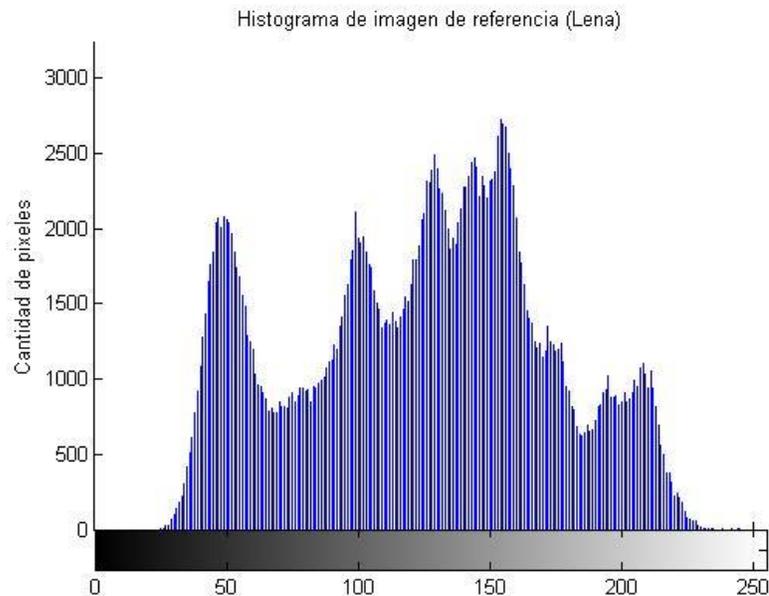


Figura 17: Histograma de Lena (imagen de referencia).

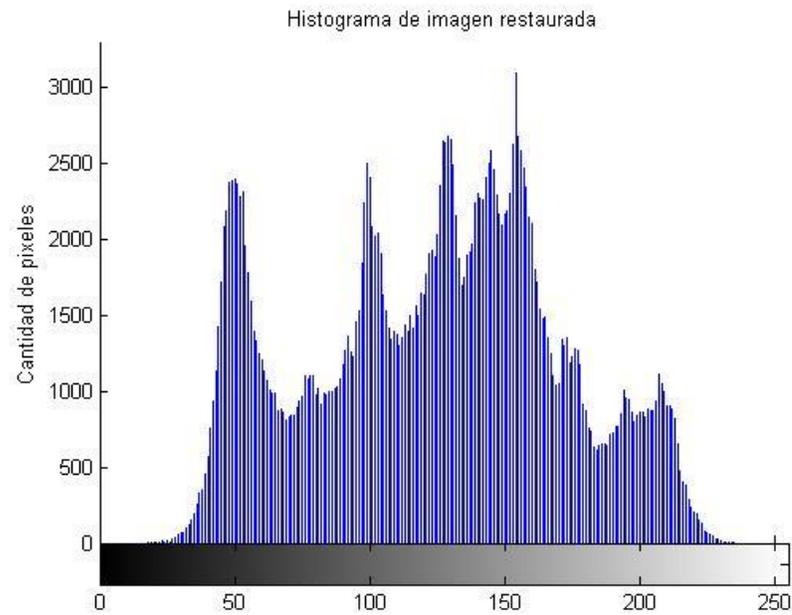


Figura 18: Histograma de la imagen restaurada de Lena.

De las figuras 17 y 18, los histogramas de la imagen de Lena y la obtenida por POCS presentan similitudes morfológicas.



Figura 19: Imágenes de Lena desplazadas a nivel sub-píxel, difuminadas y sub-muestreadas.



Figura 20: Lena magnificada y restaurada con POCS.



Figura 21: Lena original (alta resolución).

Tabla 4: Indicadores cuantitativos del proceso de magnificación y restauración de Lena.

	POCS	Lena	Lena Degradada
PSNR (Db)	32.20	∞	No aplica
ECM	36.16	0	No aplica
Tamaño Imagen (Píxeles)	512 x 512	512 x 512	256 x 256

Elaboración propia.

4.4. Comparación de POCS con los algoritmos de MATLAB.

En esta prueba se realizó lo siguiente:

- Restauración iterativa y magnificación (factor de 2) de dos escenas de campos de cultivo con el algoritmo POCS.
- Magnificación (factor de 2) de dos escenas de campos de cultivo con el algoritmo de interpolación bilineal y restauración con los algoritmos de deconvolución ciega, Lucy-Richardson y filtro regularizado.
- Magnificación (factor de 2) de dos escenas de campos de cultivo con el algoritmo de interpolación bicúbica y restauración con los algoritmos de deconvolución ciega, Lucy-Richardson y filtro regularizado.

Por cada escena de campo de cultivo se generaron 4 imágenes digitales de baja resolución para que se les pueda aplicar los algoritmos mencionados, además, en el apartado 4.1 se expuso la disposición de las escenas de campos de cultivo con las que se cuentan para las pruebas; en la figura 22 se muestra con detalle lo mencionado.

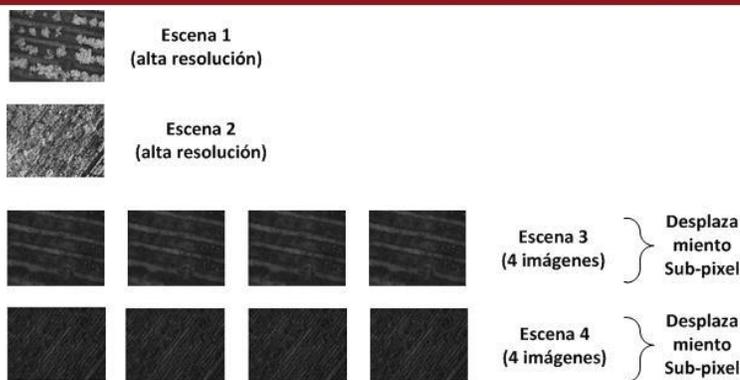


Figura 22: Disposición de escenas de campos de cultivo.

Se comparó el desempeño del algoritmo POCS con los algoritmos de magnificación y restauración de MATLAB (los de interpolación para magnificación y los de deconvolución para restauración) con los indicadores PSNR y ECM. Por otro lado, se emplearon como imágenes de referencia (alta resolución) dos imágenes de campo de cultivo (escena 1 y escena 2 en la figura 22), de tal modo que, a partir de cada una de ellas, se generaron imágenes desplazadas a nivel sub-píxel (grilla de alta definición), difuminadas y sub-muestreadas (figura 23); estas imágenes degradadas entran a los bloques de POCS, interpolación bilineal y bicúbica para ser magnificadas.

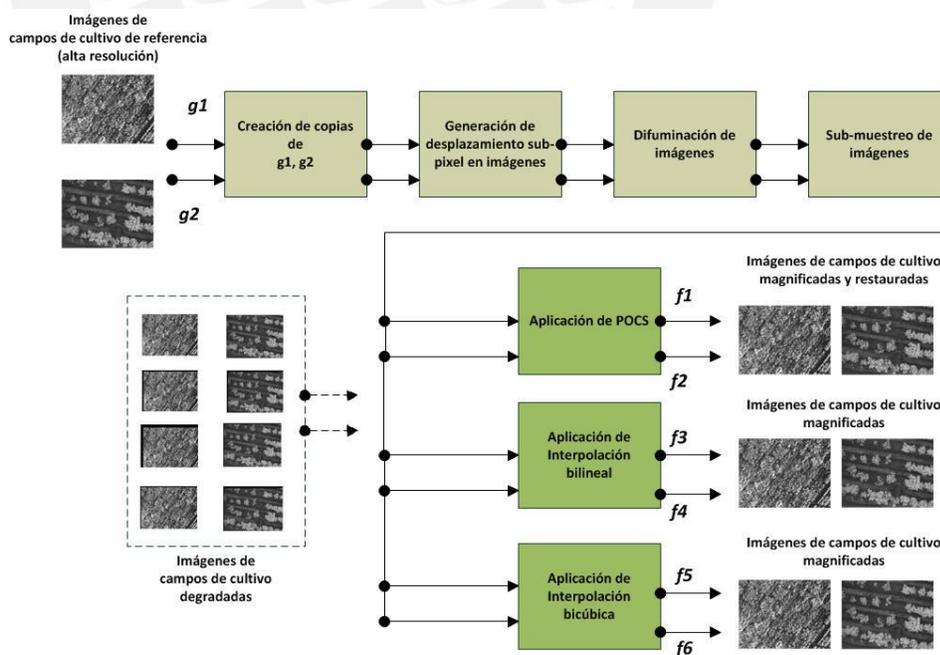


Figura 23: Generación de imágenes degradadas de campo de cultivo y aplicación de POCS, interpolación bilineal e interpolación bicúbica.

En las figuras 24 a 28 se muestran los resultados cualitativos al haberse aplicado POCS, Interpolación bilineal y bicúbica a las primeras imágenes de campo de cultivo (figura 23) y la tabla 5 muestra los resultados cuantitativos (indicadores) de la aplicación de los algoritmos mencionados.



Figura 24: Imágenes de campo de cultivo 1 degradadas.

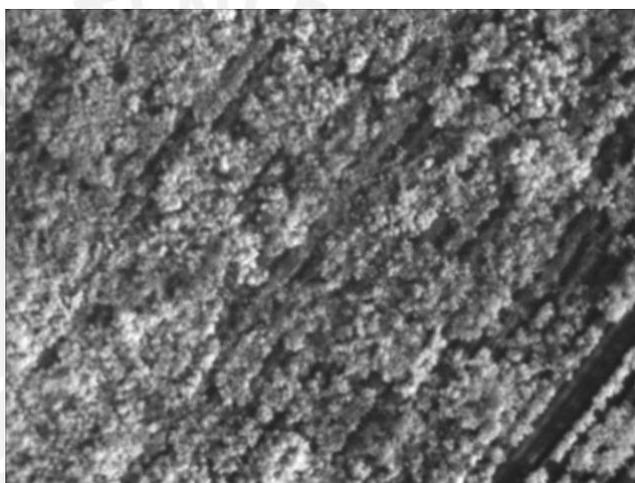


Figura 25: Imagen de campo de cultivo 1 magnificada y restaurada con POCS.

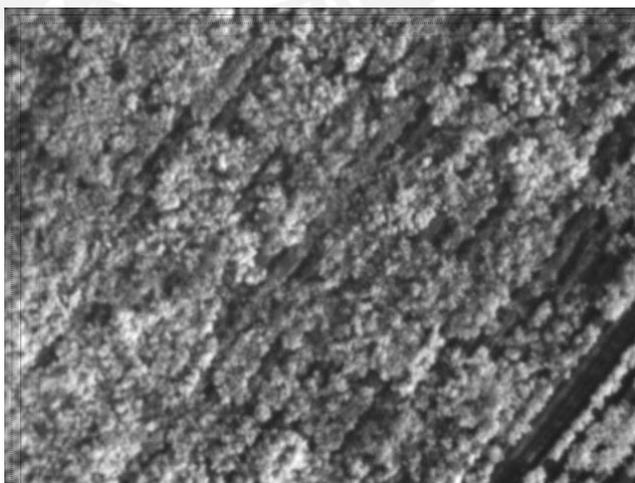


Figura 26: Imagen de campo de cultivo 1 magnificada con interpolación bilineal.

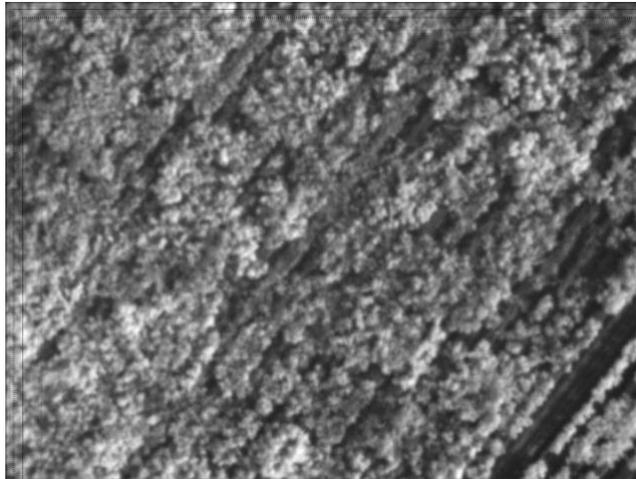


Figura 27: Imagen de campo de cultivo 1 magnificada interpolación bicúbica.

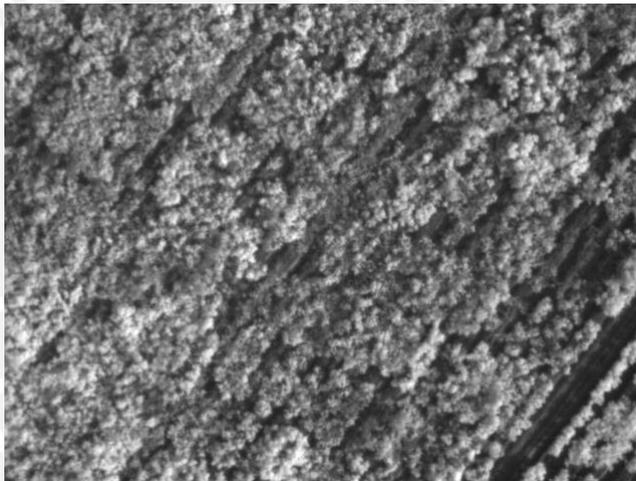


Figura 28: Imagen de campo de cultivo 1 de referencia (alta resolución).

Tabla 5: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 1. El algoritmo POCS presenta mayor valor de PSNR con respecto a los algoritmos de interpolación y menor error cuadrático medio.

	POCS	Interpolación bilineal	Interpolación bicúbica
PSNR (Db)	33.44	29.13	29.36
ECM	29.46	79.40	75.30
Tamaño Imagen (Píxeles)	480 x 640	480 x 640	480 x 640

Elaboración propia.

En las figuras 29 a 33 se muestran los resultados cualitativos al haberse aplicado POCS, Interpolación bilineal y bicúbica a las segundas imágenes de campo de cultivo (figura 23) y la tabla 6 muestra los resultados cuantitativos (indicadores) de la aplicación de los algoritmos mencionados.

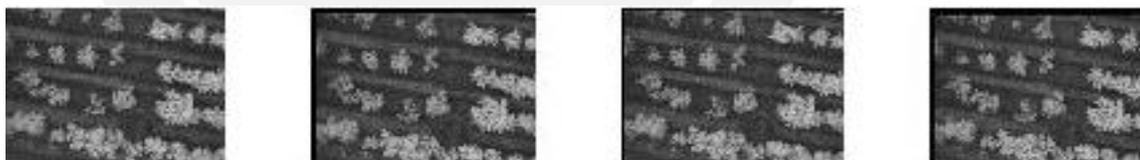


Figura 29: Imágenes de campo de cultivo 2 degradadas.

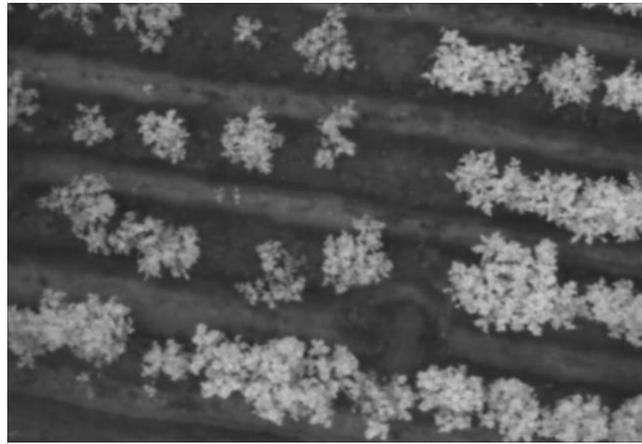


Figura 30: Imagen de campo de cultivo 2 magnificada y restaurada con POCS.

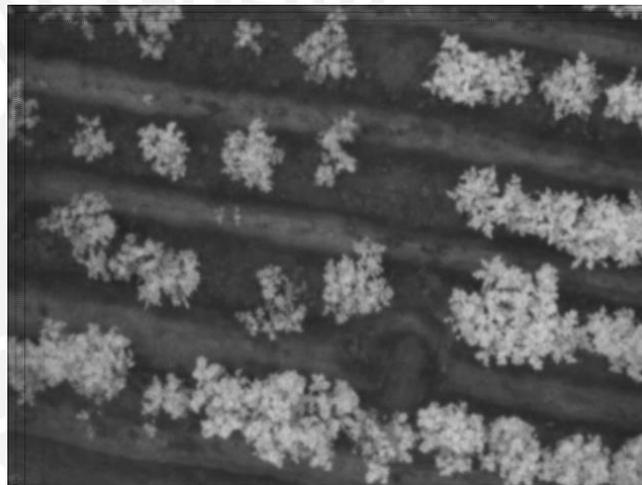


Figura 31: Imagen de campo de cultivo 2 magnificada con interpolación bilineal.

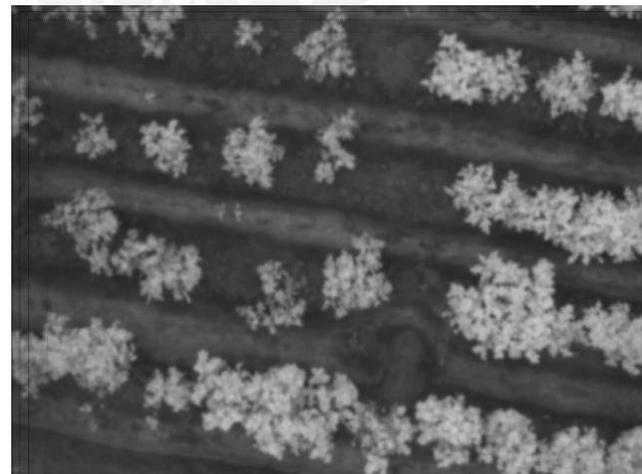


Figura 32: Imagen de campo de cultivo 2 magnificada con interpolación bicúbica.

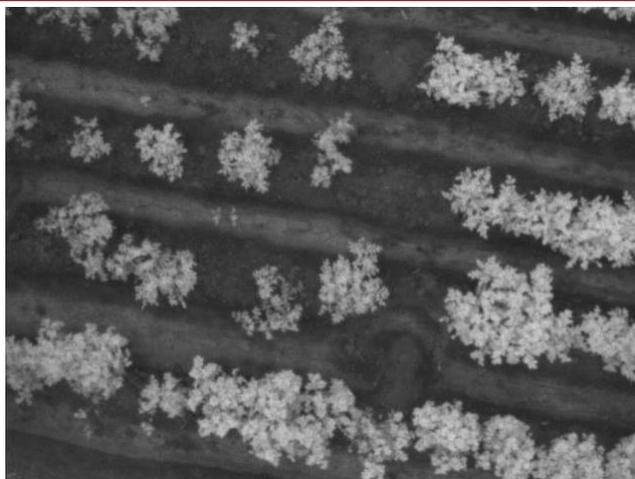


Figura 33: Imagen de campo de cultivo 2 de referencia (alta resolución).

Tabla 6: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 2. El algoritmo POCS presenta mejores resultados con respecto a los algoritmos de interpolación según los indicadores calculados de PSNR y ECM.

	POCS	Interpolación bilineal	Interpolación bicúbica
PSNR (Db)	34.15	30.15	30.16
ECM	17.28	43.41	43.28
Tamaño Imagen (Píxeles)	480 x 640	480 x 640	480 x 640

Elaboración propia.

De la tabla 6, se observa un mejor desempeño del algoritmo POCS en comparación con los algoritmos de interpolación, por otro lado, la calidad visual de las imágenes restauradas con POCS son parecidas a las de interpolación bilineal, sin embargo, la calidad de las imágenes producidas por POCS pueden mejorar aún más si se emplean más restricciones en la programación del algoritmo.

Luego de aplicar los algoritmos de interpolación para magnificar las imágenes, se aplicaron los algoritmos de restauración de MATLAB (deconvolución ciega, Lucy-Richardson y filtro regularizado) para aminorar los efectos de la difuminación.

En la figura 34 se ilustra el proceso general de restauración para las imágenes de campos de cultivo 1 y 2, las figuras 35-37 muestran los resultados de aplicar interpolación bilineal y los algoritmos de restauración a las imágenes de campo de cultivo 1, las figuras 38-40 muestran los resultados de aplicar interpolación bicúbica y los algoritmos de restauración a las imágenes de campo de cultivo 1, finalmente, las tablas 7 y 8 muestran los valores obtenidos de los indicadores y la comparación con el algoritmo POCS.

De forma similar, se aplicaron los algoritmos para las imágenes de campo de cultivo 2, las figuras 41-43 muestran los resultados para interpolación bilineal con los algoritmos de restauración de MATLAB, las figuras 44-46 muestran los resultados para interpolación bicúbica con los algoritmos de restauración mencionados y las tablas 9 y 10 muestran los valores obtenidos de los indicadores.

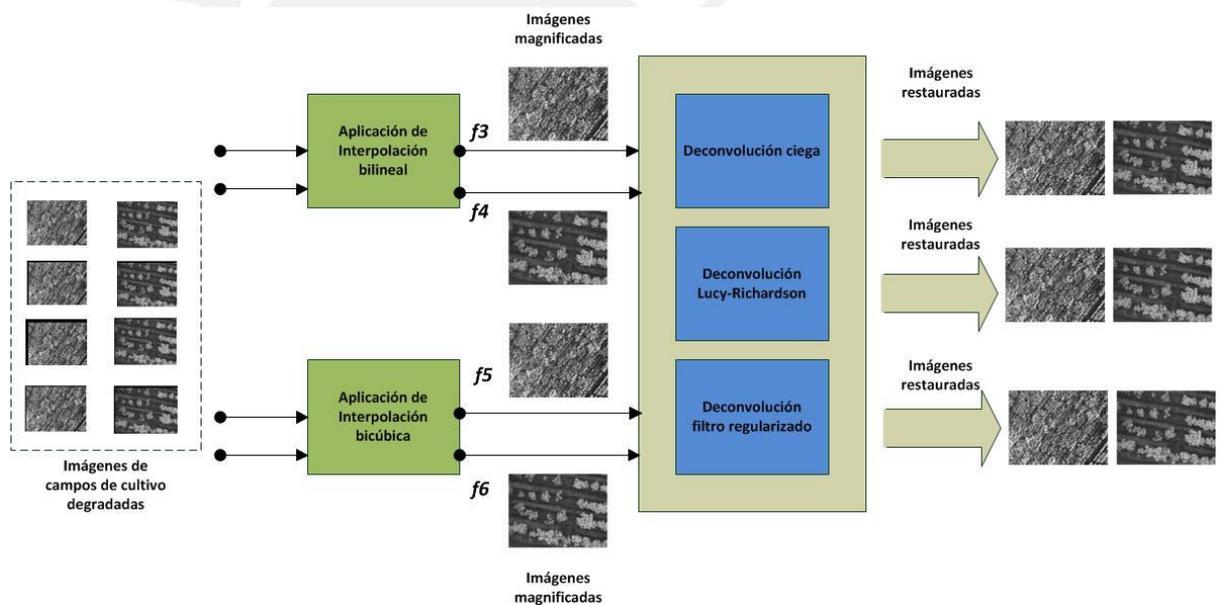


Figura 34: Aplicación de los algoritmos de restauración de MATLAB a las imágenes resultantes de los algoritmos de interpolación.

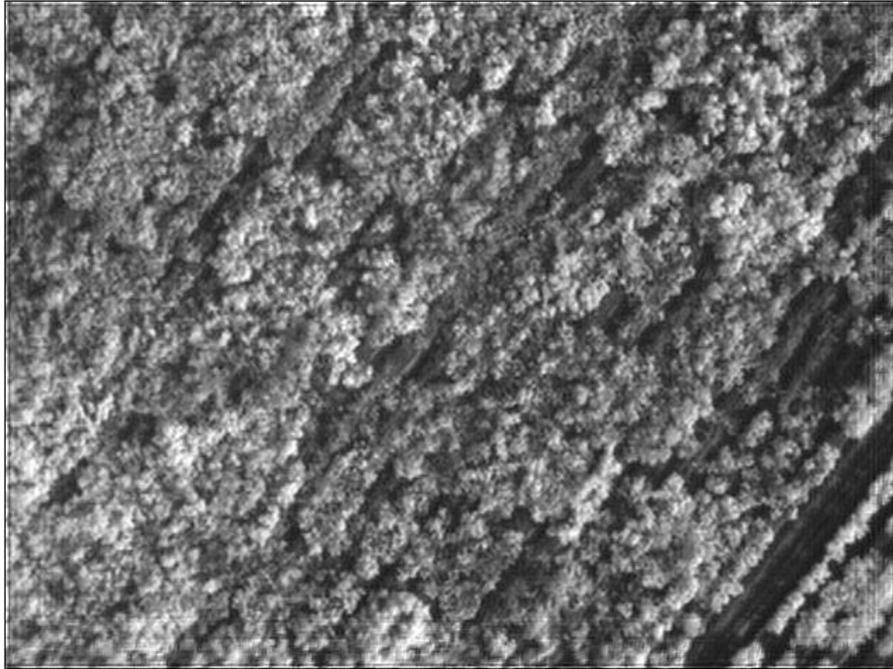


Figura 35: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución ciega para restauración a campo de cultivo 1.

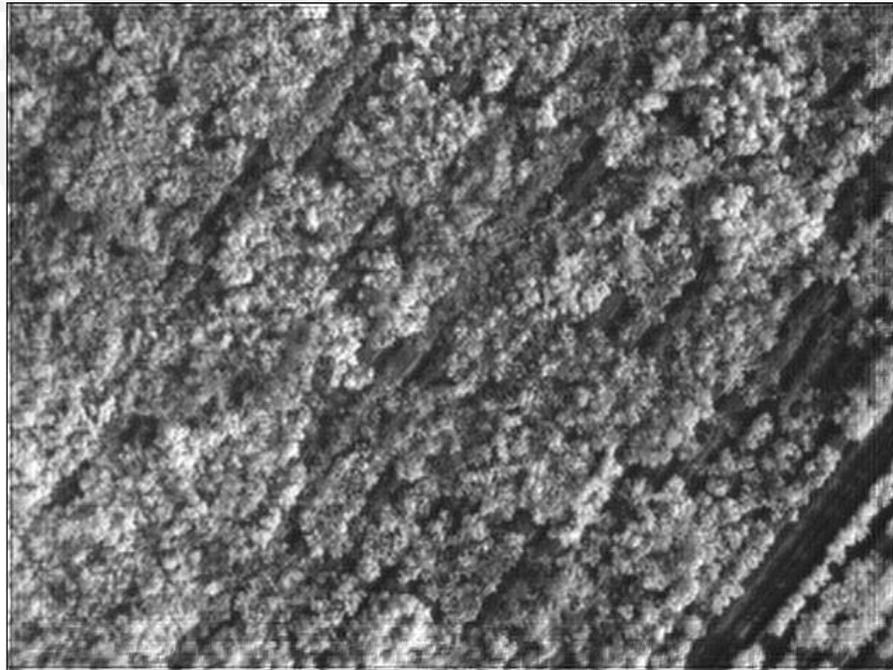


Figura 36: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución Lucy-Richardson para restauración a campo de cultivo 1.

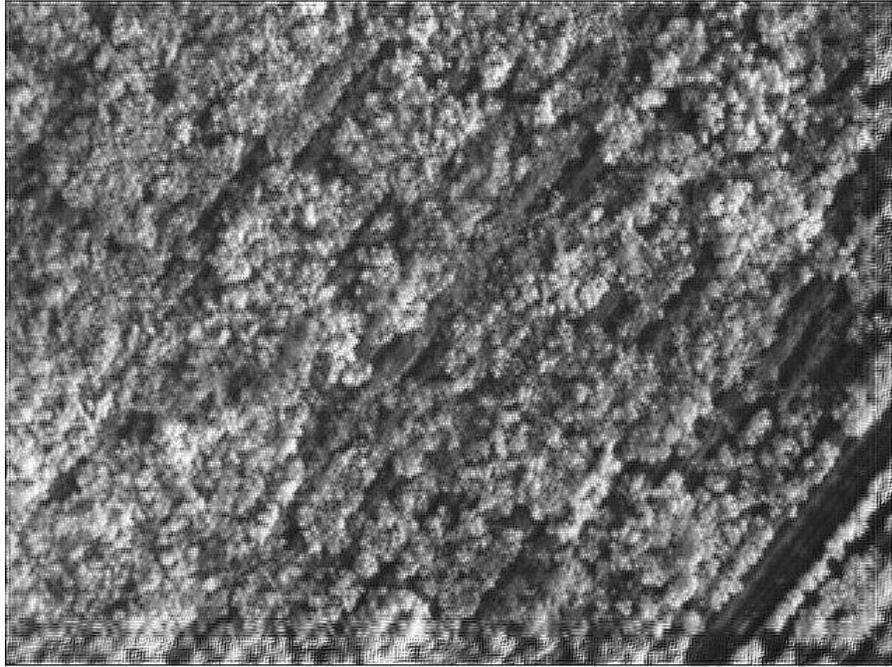


Figura 37: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución filtro regularizado para restauración a campo de cultivo 1.

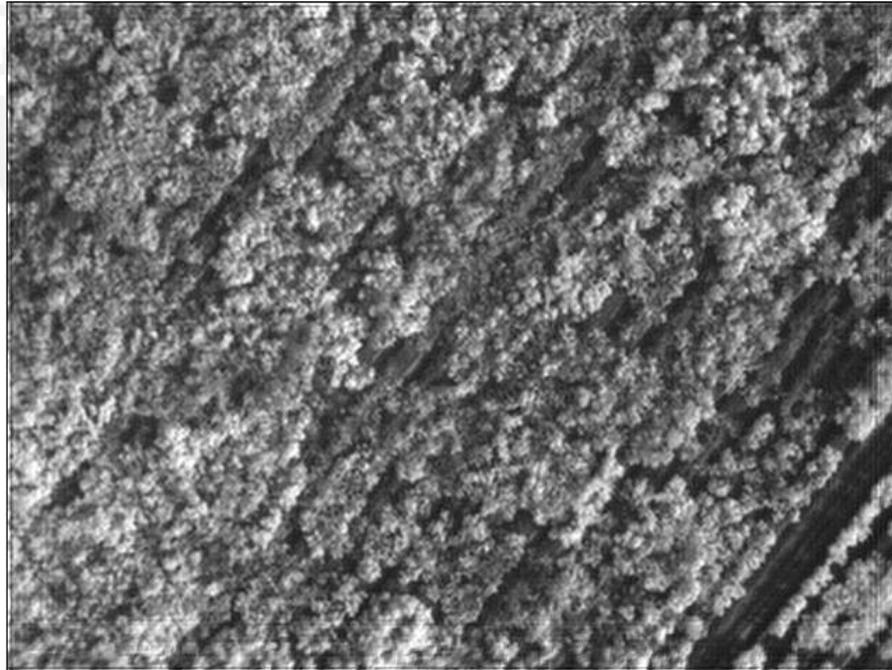


Figura 38: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución ciega para restauración a campo de cultivo 1.

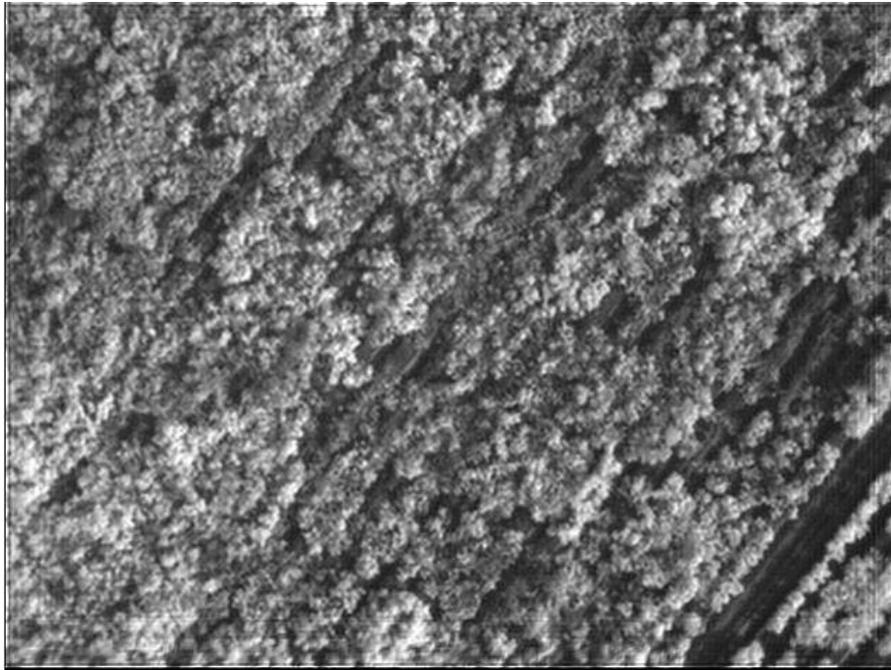


Figura 39: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución Lucy-Richardson para restauración a campo de cultivo 1.

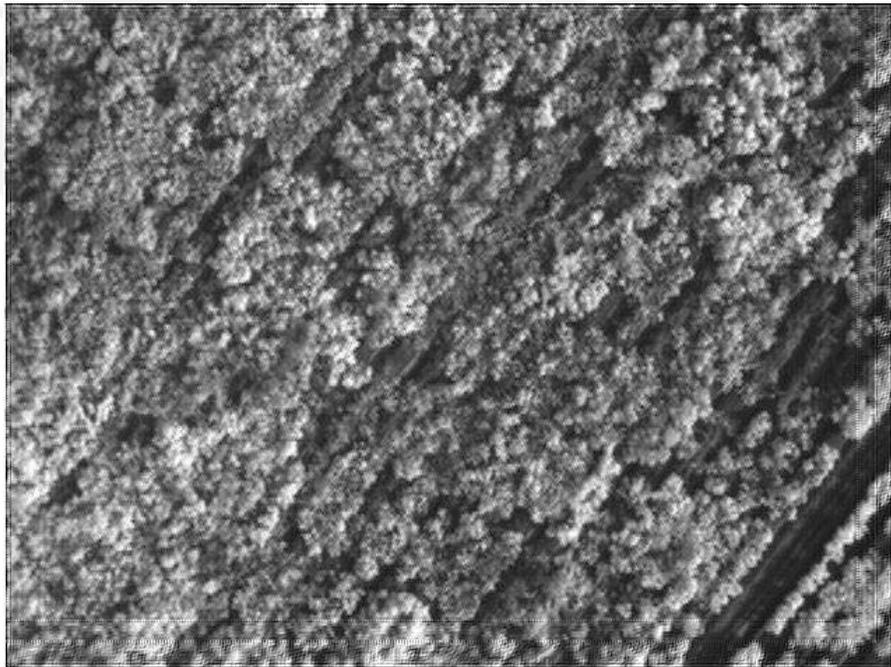


Figura 40: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución filtro regularizado para restauración a campo de cultivo 1.

Tabla 7: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 1 con los algoritmos de interpolación bilineal y de deconvolución.

	POCS	Interpolación bilineal	Interpolación bilineal y deconvolución ciega	Interpolación bilineal y deconvolución Lucy-Richardson	Interpolación bilineal y deconvolución filtro regularizado
PSNR (Db)	33.44	29.13	23.77	23.62	12.41
ECM	29.46	79.4	273.28	282.90	3732.90
Tamaño Imagen (Píxeles)	480 x 640	480 x 640	480 x 640	480 x 640	480 x 640

Elaboración propia.

Tabla 8: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 1 con los algoritmos de interpolación bicúbica y de deconvolución.

	POCS	Interpolación bicúbica	Interpolación bicúbica y deconvolución ciega	Interpolación bicúbica y deconvolución Lucy-Richardson	Interpolación bicúbica y deconvolución filtro regularizado
PSNR (Db)	33.44	29.36	23.73	23.58	12.90
ECM	29.46	75.29	275.33	285.03	3328.90
Tamaño Imagen (Píxeles)	480 x 640	480 x 640	480 x 640	480 x 640	480 x 640

Elaboración propia.

De las tablas 8 y 9, el algoritmo POCS obtuvo valores elevados de PSNR y ECM con respecto al de los algoritmos de MATLAB, tanto para los de magnificación como los de restauración, además, los resultados de deconvolución ciega, Lucy-Richardson y filtro regularizado presentan irregularidades en las zonas cercanas a los bordes de las imágenes, mientras que el algoritmo POCS no generó zonas irregulares en las imágenes a las que se aplicó.

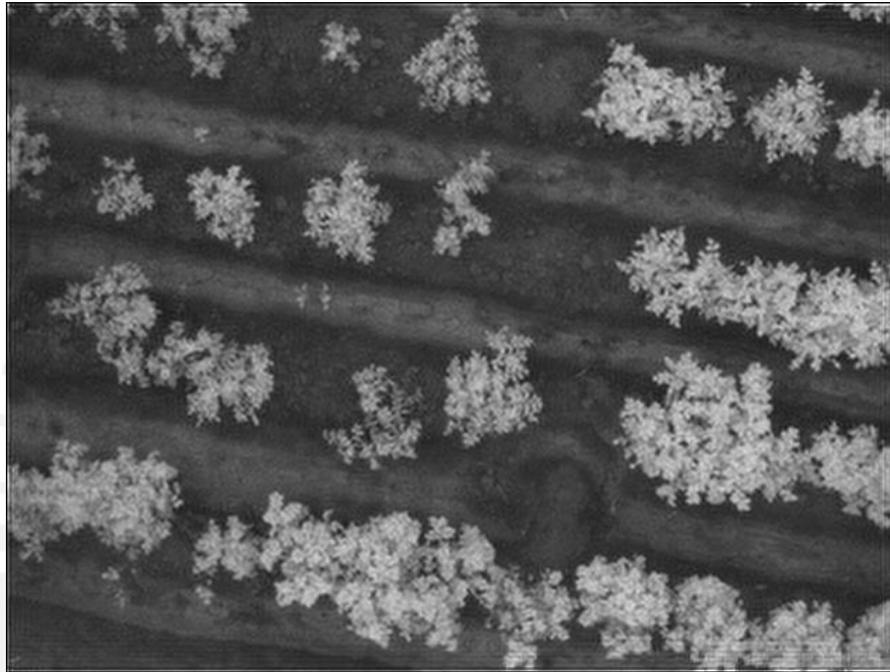


Figura 41: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución ciega para restauración a campo de cultivo 2.

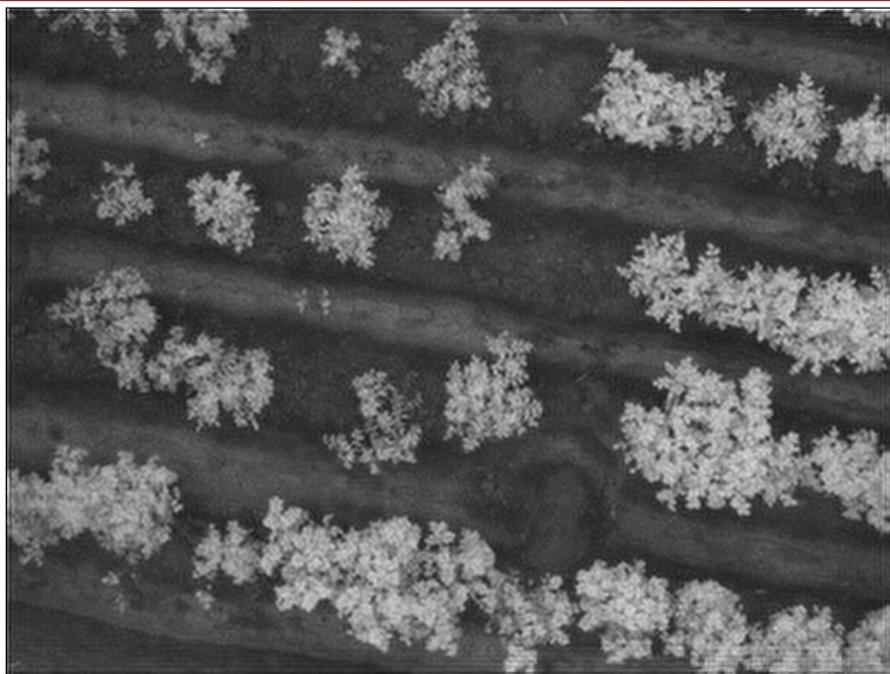


Figura 42: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución Lucy-Richardson para restauración a campo de cultivo 2.

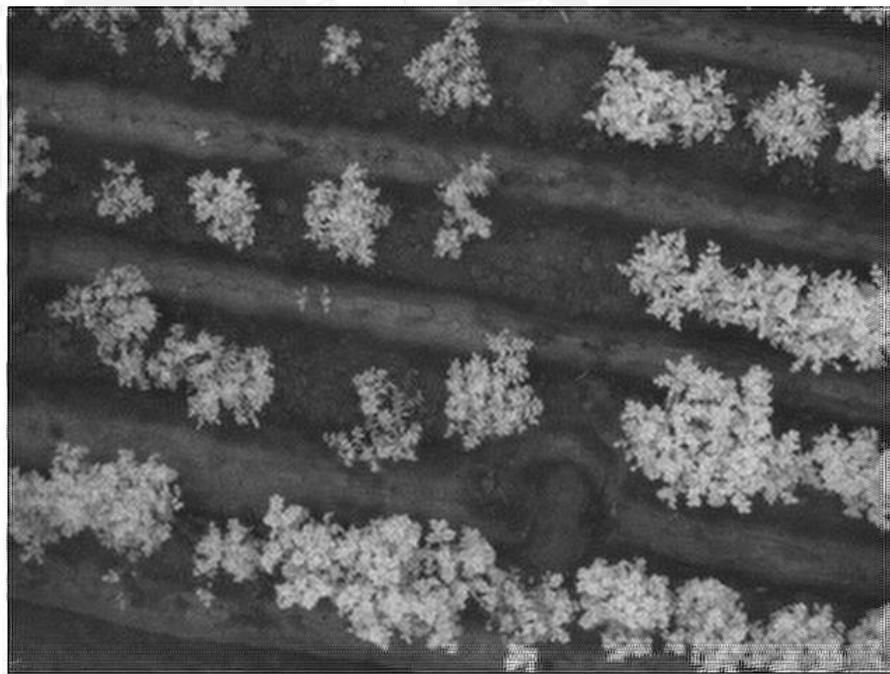


Figura 43: Aplicación de algoritmo interpolación bilineal para magnificación y deconvolución filtro regularizado para restauración a campo de cultivo 2.

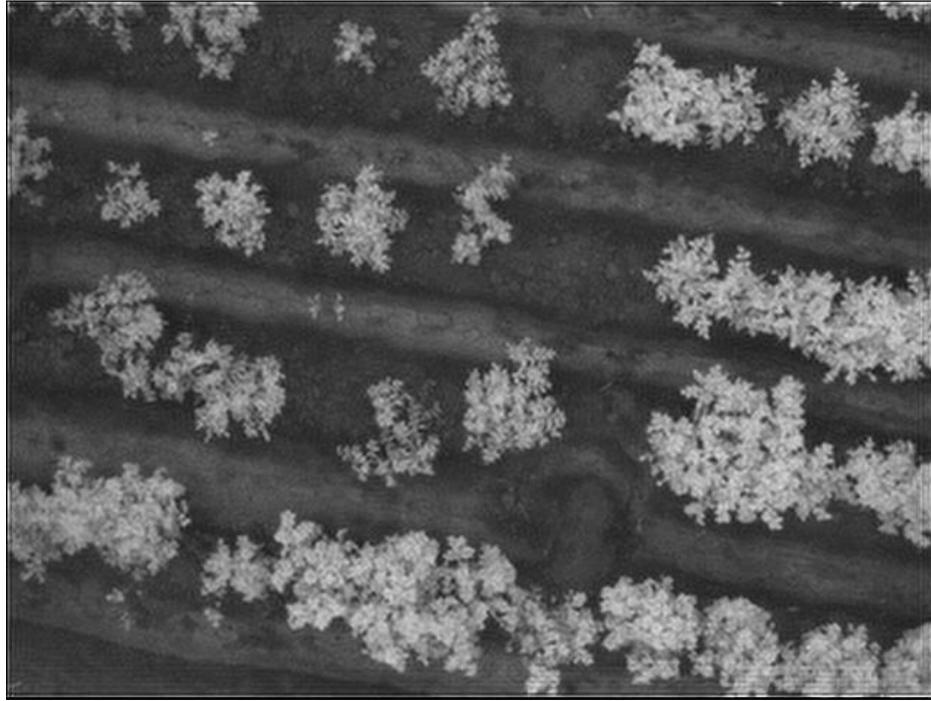


Figura 44: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución ciega para restauración a campo de cultivo 2.

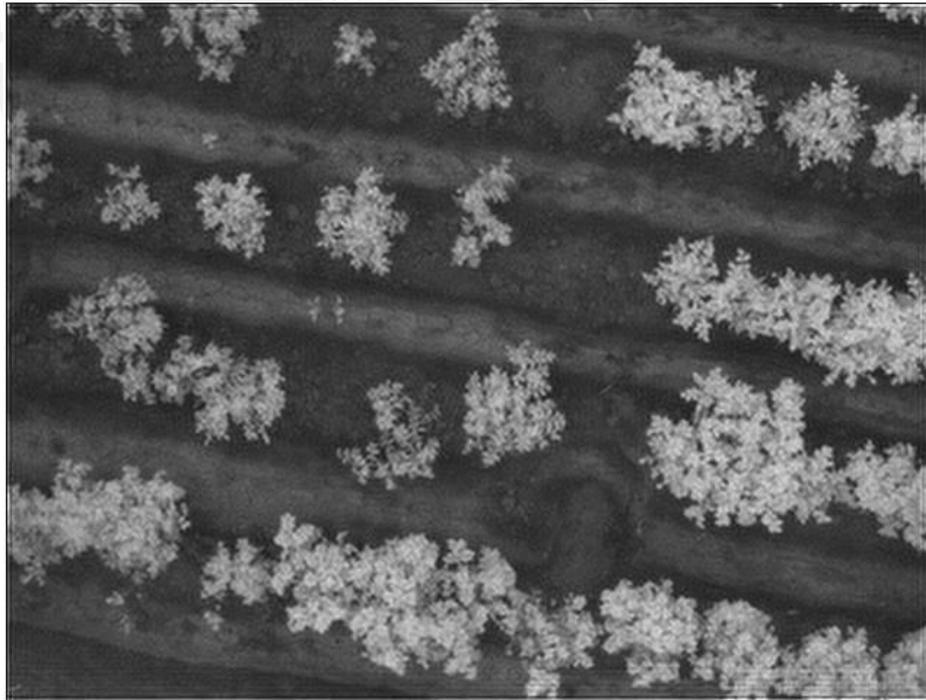


Figura 45: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución Lucy-Richardson para restauración a campo de cultivo 2.



Figura 46: Aplicación de algoritmo interpolación bicúbica para magnificación y deconvolución filtro regularizado para restauración a campo de cultivo 2.

Tabla 9: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 2 con los algoritmos de interpolación bilineal y de deconvolución.

	POCS	Interpolación bilineal	Interpolación bilineal y deconvolución ciega	Interpolación bilineal y deconvolución Lucy-Richardson	Interpolación bilineal y deconvolución filtro regularizado
PSNR (Db)	34.15	30.15	24.66	24.52	13.73
ECM	17.28	43.41	153.86	158.74	1904.90
Tamaño Imagen (Píxeles)	480 x 640	480 x 640	480 x 640	480 x 640	480 x 640

Elaboración propia.

Tabla 10: Indicadores cuantitativos del proceso de magnificación y restauración de la imagen de campo de cultivo 2 con los algoritmos de interpolación bicúbica y de deconvolución.

	POCS	Interpolación bicúbica	Interpolación bicúbica y deconvolución ciega	Interpolación bicúbica y deconvolución Lucy-Richardson	Interpolación bicúbica y deconvolución filtro regularizado
PSNR (Db)	34.15	30.16	24.64	24.51	13.77
ECM	17.28	43.28	154.25	159.09	1886.10
Tamaño Imagen (Píxeles)	480 x 640	480 x 640	480 x 640	480 x 640	480 x 640

Elaboración propia.

De las tablas 9 y 10, el algoritmo POCS presenta indicadores elevados de PSNR y bajo ECM con respecto a los algoritmos de MATLAB, además, las imágenes obtenidas después de aplicar los algoritmos de restauración deconvolución ciega, Lucy-Richardson y filtro regularizado, presentan zonas degradadas cercanas a los bordes, mientras que las imágenes obtenidas mediante POCS no presentan dichas características. Si se obvian los bordes en los cálculos de los indicadores cuantitativos, los resultados de los algoritmos basados en deconvolución mejorarían significativamente en desempeño, sin embargo, con el algoritmo POCS no es necesario tener en cuenta los bordes ni excluirlos del procesamiento, ya que, por las características de las restricciones definidas en el capítulo 3 para POCS, el procesamiento toma en cuenta todos los píxeles de la imagen. En [9] y [12] se encuentran más reportes y cálculos de desempeño de los algoritmos POCS, interpolación y de deconvolución.

4.5. Aplicación de POCS a imágenes de campos de cultivo.

En esta prueba se realizó lo siguiente:

- Restauración iterativa y magnificación (factor de 2) de imágenes digitales de baja resolución de campos de cultivo con el algoritmo POCS.

Dado que se tienen 2 escenas de campos de cultivo diferentes y cada una de ellas tiene 4 imágenes digitales desplazadas a nivel sub-píxel que la representan, se aplicó el algoritmo POCS para obtener, para cada escena, una imagen magnificada por un factor de 2 (figura 47).

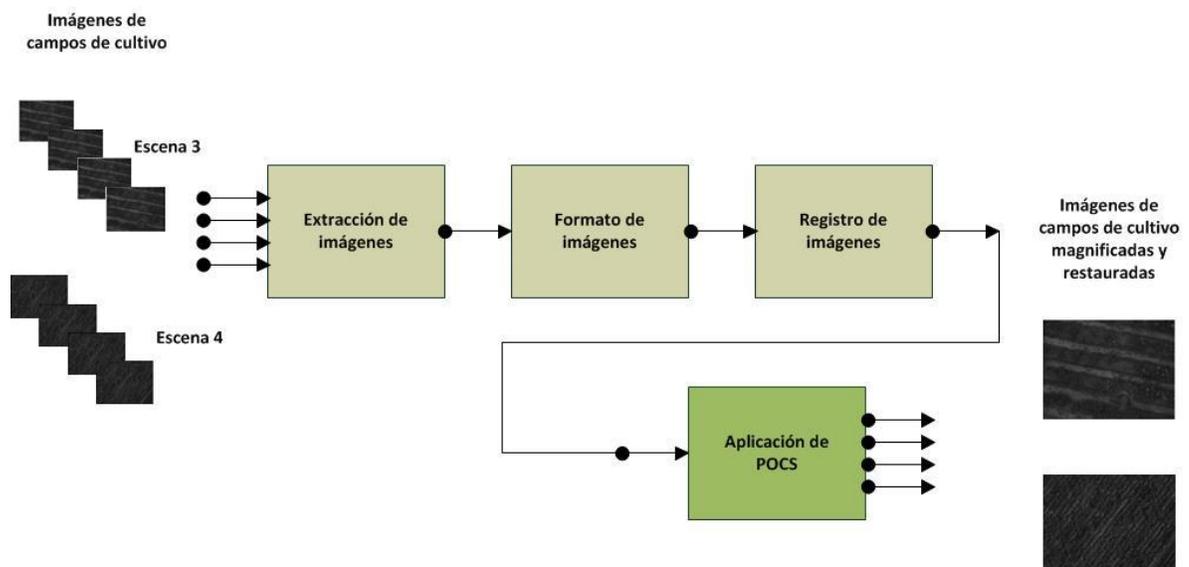


Figura 47: Aplicación de algoritmo POCS a las imágenes de campos de cultivo.

En la figura 47, en el bloque de extracción de imágenes se obtuvieron las imágenes correspondientes a las escenas de los campos de cultivo y se almacenaron en el espacio de trabajo del software MATLAB; en el bloque de formato de imágenes se transformaron las imágenes a un formato que permite aplicar operaciones aritméticas y lógicas sobre ellas; en el bloque de registro de imágenes se calcularon los valores de desplazamiento sub-píxel entre las imágenes correspondientes a cada escena; finalmente, en el bloque de aplicación de POCS, se ejecuta el algoritmo a todas las imágenes de cada escena (por separado).

En las figuras 48-49 se aprecian los resultados de la aplicación de POCS (magnificación y restauración) a las dos escenas de campos de cultivo.

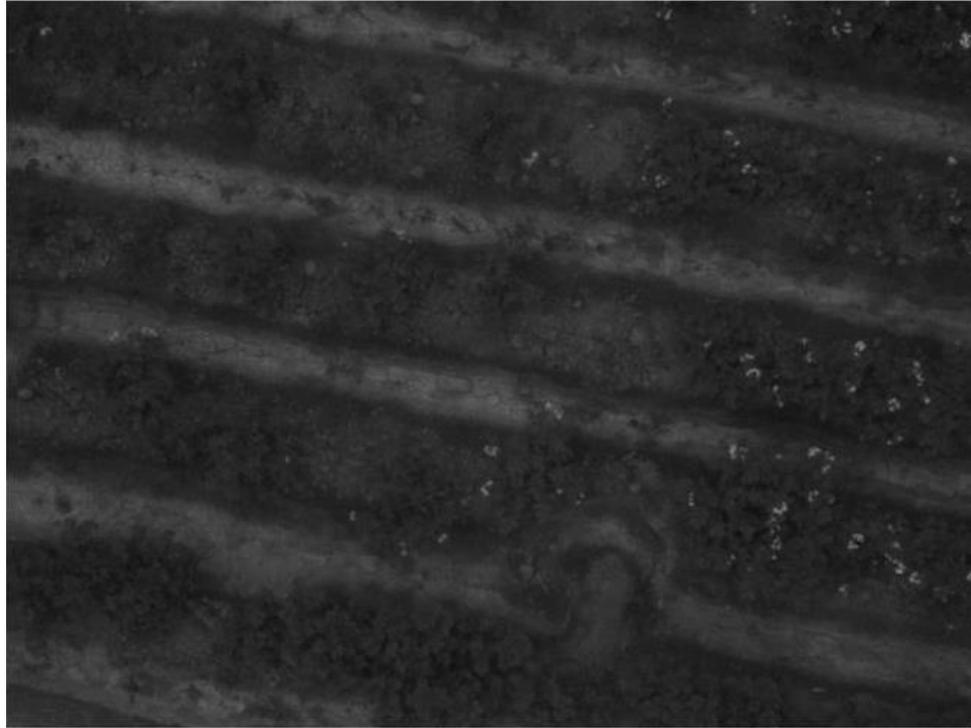


Figura 48: Aplicación de algoritmo POCS a las imágenes de campos de cultivo de la escena 3.



Figura 49: Aplicación de algoritmo POCS a las imágenes de campos de cultivo de la escena 4.

Dado que en esta prueba no se conoce una imagen de referencia de alta resolución a priori, para evaluar el desempeño de POCS en este escenario se realizó lo siguiente:

- Generación de 4 copias de las imágenes correspondientes a las escenas 3 y 4 magnificadas y restauradas con POCS.
- Desplazamiento sub-píxel de cada copia según los valores de registro calculados.
- Difuminación de las imágenes desplazadas a nivel sub-píxel.
- Sub-muestreo de las imágenes desplazadas por un factor de 1/2.
- Cálculo del ECM y el PSNR entre las 4 imágenes originales (escenas 1 y 2) y las generadas artificialmente para determinar el parecido.

La tabla 11 muestra los resultados cuantitativos obtenidos de PSNR y ECM en la presente prueba.

Tabla 11: Indicadores cuantitativos obtenidos al comparar las imágenes de baja resolución originales con las generadas artificialmente con las escenas 3 y 4.

	Escena 3	Escena 4
PSNR (Db)	38.30	38.77
ECM	2.5	1.72
Tamaño Imagen (Píxeles)	480 x 640	480 x 640

Elaboración propia.

En la tabla 11, los valores de PSNR son elevados y los de ECM muy cercanos a cero, lo que significa que la aplicación de POCS a imágenes de campos de cultivo con desplazamiento sub-píxel ofrece buenos resultados de magnificación y restauración.

En los Anexos 2 y 3 se puede encontrar más detalles del código de las funciones utilizadas y de las imágenes.

CONCLUSIONES

1. El algoritmo POCS pudo ser programado y permitió obtener una imagen de alta resolución base al desplazamiento sub-píxel existente en las imágenes de baja resolución capturadas.
2. El algoritmo POCS presentó valores elevados de PSNR y bajos valores de error cuadrático medio en comparación con los algoritmos de MATLAB que se incluyeron en las pruebas, los cuales son: interpolación bilineal, interpolación bicúbica, deconvolución ciega, deconvolución Lucy-Richardson y deconvolución por filtro regularizado, lo que significa que POCS obtuvo mejores resultados que todos los algoritmos de MATLAB.
3. El algoritmo POCS permitió aumentar la resolución de imágenes de campos de cultivo de manera satisfactoria, además, también presentó buenos resultados para imágenes que no son de campos de cultivo.
4. El beneficio del algoritmo POCS es que puede restaurar y magnificar imágenes de baja resolución sin necesidad de invertir recursos económicos en la adquisición de hardware óptico para mejorar la resolución de las imágenes, además, el algoritmo tiene la capacidad de ser reprogramado para mejorar la calidad de las imágenes que se restauran con él en la medida en que se conozca más información de los procesos y características que determinaron la captura de las mencionadas imágenes.

RECOMENDACIONES

1. Incorporar al algoritmo base de POCS, más restricciones de las que se programaron en el presente trabajo; ello conllevará a obtener una mejor imagen de alta resolución.
2. Incorporar en la programación de POCS un mecanismo de salida del bucle de iteraciones cuando el porcentaje de error por iteración sea igual a un valor pre-establecido, de tal modo que el algoritmo se lleve cabo en menor tiempo.
3. Incorporar en la etapa de registro un algoritmo que tenga en consideración rotaciones relativas entre las imágenes de baja resolución, ello conllevará a tener más información de la escena que se necesita reconstruir.
4. Realizar pruebas de evaluación del algoritmo POCS en presencia de ruido, ya que permitirán conocer el comportamiento del algoritmo en dicho escenario, con lo cual permitirá, a su vez, programar restricciones o variantes del algoritmo que mejoren el desempeño frente al ruido.

FUENTES

[1] Emiliano García. “Agricultura de Precisión”

[consultado 1/04/2015]

<http://www.palermo.edu/ingenieria/downloads/pdfwebc&T8/8CyT12.pdf>

[2] SGS. “Agricultura de Precisión”

[consultado 5/04/2015]

<http://www.sgs.pe/es-ES/Agriculture-Food/Seed-and-Crop/Soil-Leaf-and-Water-Services/Precision-Farming.aspx>

[3] Scan Terra. “Agricultura de Precisión”

[consultado 5/04/2015]

http://www.scanterra.com.ar/agricultura_precision_inf.html

[4] Miguel Herrera. “Banderillero Satelital”

[consultado 5/04/2015]

<https://mecanizacionagricolafca.files.wordpress.com/2012/09/banderillero-satelital.pdf>

[5] Geo Agro. “Imágenes satelitales I: radiografías del agro”

[consultado 6/04/2015]

<https://www.geoagro.com/blog/im%C3%A1genes-satelitales>

[6] S. C. Park, M. K. Park, and M. G. Kang, “Super-resolution image reconstruction: A technical overview”, IEEE Signal Processing Magazine, vol.20(3), Mayo 2003, pp. 21-36.

[7] T. Komatsu, K. Aizawa, T. Igarashi, and T. Saito, “Signal-processing based method for acquiring very high resolution image with multiple cameras and its theoretical analysis,” Proc. Inst. Elec. Eng., vol. 140, no. 1, pt. I, pp. 19-25, Febrero. 1993.

[8] RODRIGUEZ, Roberto.

2012 Procesamiento y análisis digital de imágenes. Primera edición. C.V., México: Alfa Omega Grupo Editor.

[9] Xiaoqing Su, Shutao Li, "Multi-frame Image Super-resolution Reconstruction based on Sparse Representation and POCS International Journal of Digital Content Technology and its Applications", International Journal of Digital Content Technology and its Applications. Volumen 5, número 8, Agosto 2011.

[10] Hafner, M.; Liedgruber, M.; Uhl, A., "POCS-based super-resolution for HD endoscopy video frames," in Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on , vol., no., pp.185-190, 20-22 Junio 2013.

[11] Yao Wang. "Image Sampling and Resizing"

[consultado 16/05/2015]

http://eeweb.poly.edu/~yao/EL5123/lecture8_sampling.pdf

[12] Xie Wei; Zhang Feiyan; Chen Hao; Qin Qianqing, "Blind Super-Resolution Image Reconstruction Based on POCS Model," Measuring Technology and Mechatronics Automation, 2009. ICMTMA '09. International Conference, vol.1, no., pp.437, 440, 11-12 Abril 2009.

[13] Elad, M.; Feuer, A., "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," in Image Processing, IEEE Transactions on , vol.6, no.12, pp.1646-1658, Diciembre 1997.

[14] R.Y. Tsai and T.S. Huang, "Multipleframe image restoration and registration," in Advances in Computer Vision and Image Processing. Greenwich, CT: JAI Press Inc., 1984, pp. 317-339.

[15] J. J. Clark, M R. Palmer, P.D. Laurence, "A transformation method for the reconstruction of functions from nonuniformly spaced samples," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-33, pp. 1151-1165, 1985.

[16] Siddique, M., Khan, R., Khan, K., & Ahmad, N. (2012). Performance Evaluation of Super-Resolution Reconstruction Algorithms Based On Linear Magnifications. International Journal Of Computer Science Issues (IJCSI), 9(3), 207-215.

[17] Zhengwei Yang, Rick Mueller, UNBIASED HISTOGRAM MATCHING QUALITY MEASURE FOR OPTIMAL RADIOMETRIC NORMALIZATION, ASPRS 2008 Annual Conference Portland, Oregon , Abril 28 - Mayo 2, 2007.

[18] BOYD, Stephen.

2004 Convex Optimization. C.V., Cambridge: Cambridge University Press.

[19] David Joyce. "Convexity"

[consultado 16/05/2015]

<http://www.quora.com/Why-might-a-set-of-points-equidistant-from-a-center-point-form-a-round-shape>

[20] Mathematics dictionary. "Conjunto Cerrado"

[consultado 16/05/2015]

<http://www.mathematicsdictionary.com/spanish/vmd/full/c/closedset.htm>

[21] Bertsekas, Dimitri P.

2003 Convex analysis and optimization. Belmont, Mass.: Athena Scientific.

[22] Lay, David.

2012 Linear algebra and its applications. Cuarta Edición. Boston: Addison-Wesley.

[23] John K. Hunter. "Applied Analysis"

[consultado 16/05/2015]

<https://www.math.ucdavis.edu/~hunter/book/ch6.pdf>

[24] Paul Klein. "Stockholm doctoral program in economics".

[consultado 16/05/2015]

<http://paulklein.ca/newsite/teaching/projections.pdf>

[25] BOYD, Stephen. "Alternating Projections".

[consultado 16/05/2015]

http://web.stanford.edu/class/ee392o/alt_proj.pdf

[26] DATTORRO. "Convex Optimization".

[consultado 16/05/2015]

http://www.convexoptimization.com/dattorro/orthogonal_and_nonorthogonal_projection.html

[27] W. Cheney and A. Goldstein. Proximity maps for convex sets. Proceedings of the AMS, 10:448–450, 1959.

[28] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration", presented at IEEE International Conference of Acoustics, Speech and Signal Processing, San Francisco, California, USA, 1992.

[29] D. C. Youla and H. Webb, "Image restoration by the method of convex projections: Part 1-Theory," this issue, pp. 81-94.

[30] D.C. Youla, "Generalized image restoration by the method of alternating projections," IEEE Trans. Circuits Syst., vol. CAS-25, Sept. 1978.

[31] Cheng Yan, "Study of fast super-resolution reconstruction method," Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on , vol.4, no., pp.156,159, 10-12 Junio 2011.

[32] Sezan, M.I.; Stark, H., "Image Restoration by the Method of Convex Projections: Part 2-Applications and Numerical Results," Medical Imaging, IEEE Transactions on , vol.1,no.2,pp.95,101,Oct.1982.

doi: 10.1109/TMI.1982.4307556.

[33] Tom Distler. "Image Quality Assessment (IQA)", 2011

[consultado 10/05/2015]

<http://tdistler.com/iqa/algorithms.html>

[34] National Instruments. "Peak Signal-to-Noise Ratio as an Image Quality Metric", 2013

[consultado 10/05/2015]

<http://www.ni.com/white-paper/13306/en/>

[35] S.S, Panda; M.S.R.S Prasad; G, Jena., "POCS Based Super-Resolution Image Reconstruction Using an Adaptive Regularization Parameter," IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 2, Septiembre 2011 ISSN (Online): 1694-0814.

[36] GONZALEZ, Rafael.

2008 Digital image processing. Tercera edición. Upper Saddle River, NJ: Pearson Education.

[37] Stark, H.; Oskoui, P., "High-resolution image recovery from image-plane arrays, using convex projections," J. Opt. Soc. Am. A6, 1715-1726(1989). doi: 10.1364/JOSAA.6.001715.