

Fusión de Imágenes Multi-Foco con Ventanas Variables

Felix Calderon*, Adan Garnica-Carrillo, Juan J. Flores

División de Estudios de Posgrado. Facultad de Ingeniería Eléctrica. Universidad Michoacana de San Nicolas de Hidalgo. Santiago Tapia 403. Col. Centro. C.P. 58000. Morelia Michoacan México

Resumen

En este artículo presentamos el Algoritmo Combinación Lineal de Imágenes con Ventanas Variables (CLI-VV) para la fusión de imágenes multi-foco. A diferencia del Algoritmo CLI-S presentado en un trabajo anterior, el algoritmo CLI-VV permite determinar automáticamente el tamaño óptimo de la ventana en cada píxel para la segmentación de las regiones con la mayor nitidez. También presentamos la generalizado el Algoritmo CLI-VV para la fusión de conjuntos de imágenes multi-foco con más de dos imágenes. A este nuevo algoritmo lo denominamos Fusión Multi-foco con Ventanas Variables (FM-VV). El Algoritmo CLI-VV se probó con 21 pares de imágenes sintéticas y 29 pares de imágenes multi-foco reales, y el Algoritmo FM-VV sobre 5 tríos de imágenes multi-foco. En todos los ejemplos se obtuvo un porcentaje de acierto competitivos, producidos en tiempos de ejecución menores a los reportados en la literatura.

Palabras Clave:

Fusión de imágenes multi-foco, Ventanas deslizantes, Imágenes integrales

Multi Focus Image Fusion with variable size windows

Abstract

In this paper we present the Linear Image Combination Algorithm with Variable Windows (CLI-VV) for the fusion of multi-focus images. Unlike the CLI-S algorithm presented in a previous work, the CLI-VV algorithm allows to automatically determine the optimal size of the window in each pixel for the segmentation of the regions with the highest sharpness. We also present the generalized CLI-VV Algorithm for the fusion of sets of multi-focus images with more than two images. This new algorithm is called Variable Windows Multi-focus Fusion (FM-VV). The CLI-VV Algorithm was tested with 21 pairs of synthetic images and 29 pairs of real multi-focus images, and the FM-VV Algorithm on 5 trios of multi-focus images. In all the tests a competitive accuracy was obtained, with execution times lower than those reported in the literature.

Keywords:

Multi-focus image fusion, Sliding windows, Incremental images.

1. Introducción

Hay muchas aplicaciones en las que se involucran imágenes, por ejemplo, en sistemas de seguridad, en fotografía, en apoyo al diagnóstico con imágenes médicas, en imágenes satelitales y en algunos ambientes de investigación. En ellas es deseable que las imágenes obtenidas tengan suficiente claridad o nitidez en todos los objetos de la escena. Sin embargo, lo anterior no siempre es posible pues el rango de distancias medidos desde la lente hasta los objetos que se desean capturar no es constante y las cámaras, microscopios y otros objetos que

usan lentes para la captura de las imágenes, tienen limitaciones físicas que impiden que todos los objetos de la escena tengan la misma claridad en la imagen digital generada. A dicha limitación se le conoce como profundidad de campo, término que hace referencia al rango de distancias que la lente de una cámara, con una apertura dada, es capaz de reproducir de manera nítida (Kuthirummal et al., 2011).

En términos ideales, la luz reflejada por cada punto de la escena debería incidir sobre un único punto en el sensor de la cámara, sin embargo, dada la curvatura de la lente, la luz reflejada por puntos que están a una distancia fuera del rango abarca-

*Autor para correspondencia: calderon@umich.mx

do por la profundidad de campo inciden sobre varias posiciones en el sensor, generando pequeños círculos en lugar de generar un sólo punto. Dichos círculos son conocidos como círculos de confusión y están definidos en (Sezan et al., 1991).

El valor de color o nivel de gris de los píxeles de la imagen generada que son afectados por los círculos de confusión es el resultado de una mezcla de colores que corresponden a varios puntos de la escena en lugar de ser el resultado del color de un sólo punto, esta situación provoca que esos píxeles aparezcan borrosos o desenfocados.

Al no poder capturar de forma nítida todos los objetos de una escena en una sola fotografía, se tiene que tomar la decisión de qué objeto se desea tener más claro en la fotografía y enfocar la cámara para que dicho objeto quede dentro del rango abarcado por la profundidad de campo de la lente. Variando el nivel de enfoque de la cámara podemos llegar a tener un conjunto $I = \{I_1, I_2, \dots, I_N\}$ con N imágenes en las que los detalles de los objetos de interés de la escena, se aprecien de forma nítida en por lo menos una de las imágenes del conjunto. A dicho conjunto de imágenes obtenidas mediante la variación del nivel de enfoque, se le conoce como imágenes multi-foco. El problema de fusión de imágenes multi-foco consiste en extraer de la k -ésima imagen I_k , aquellas regiones que son más nítidas que en cualquier otra imagen del conjunto, con el objetivo de generar una imagen final I_F compuesta por las regiones nítidas.

En la Figura 1 se muestra un ejemplo de tres imágenes multi-foco, en las que aparecen tres tazas. En la Figura 1(a) se muestra una imagen donde el enfoque de la cámara fue ajustado para obtener de forma clara la taza que aparece a la izquierda de la escena. La Figura 1(b) muestra la imagen donde el enfoque fue ajustado para obtener los detalles de la taza que está en el centro de la escena y finalmente en la Figura 1(c), el nivel de enfoque se ajustó para obtener de forma nítida los detalles de la taza que aparece a la derecha de la escena. Para este ejemplo la fusión de imágenes multi-foco consiste en extraer de cada una de las tres imágenes mostradas en la Figura 1, la taza que aparece nítida en cada imagen y posteriormente combinar la información extraída para formar una nueva imagen I_F , en la que se puedan observar de forma clara los detalles de cada una de las tazas.



Figura 1: Ejemplo de imágenes multi-foco

Definimos una imagen digital como una representación discreta de una escena, en la que idealmente cada punto de la escena mapea a un punto en la imagen. Dicha imagen I_k es un arreglo bidimensional de valores, con n_r renglones y n_c columnas en donde $I_k(r, c)$ representa el nivel de color en las coordenadas (r, c) de la k -ésima imagen. También definimos un mapa de decisión P como un arreglo bidimensional de tamaño $n_r \times n_c$. Cada uno de los elementos de este arreglo $P(r, c)$ tendrá un valor $k-1$ donde k corresponde al índice de la imagen cuyo píxel tiene la mayor nitidez en las coordenadas (r, c) . En el caso de conjuntos

multi-foco con dos imágenes el mapa de decisión $P(r, c)$ tomará valores binarios y solamente en este caso puede considerarse una matriz de pesos.

No sabemos de la existencia de un valor de referencia o repositorio de imágenes multi-foco que tenga valores de referencia de los mapas de decisión. Por tal motivo generamos nuestras propias imágenes sintéticas en las que tenemos un valor de referencia del mapa de decisión y en consecuencia se puede estimar la eficiencia del algoritmo de fusión que desarrollamos. Sin embargo, obtuvimos a mano el mapa de decisión de referencia de las imágenes reales (mediante el programa Gimp) para utilizar métricas que permitan dar un valor cuantitativo de los resultados obtenidos.

El resto de este artículo está organizado de la siguiente manera, en la Sección 1 se hace una definición del problema de fusión de imágenes multi-foco. La Sección 2 menciona algunas estrategias que se han utilizado para tratar de resolver el problema de forma eficiente. La sección 3 presenta el Algoritmo CLI-VV el cual es una mejora del Algoritmo CLI-S propuesto en (Calderon et al., 2016), además se presenta la generalización para más de dos imágenes en el Algoritmo FM-VV. La Sección 4 presenta los resultados con 50 pares de imágenes, de las cuales 21 pares son sintéticos y 29 son pares de imágenes utilizados en trabajos publicados previamente. También presenta la aplicación del algoritmo FM-VV sobre 5 tríos de imágenes multi-foco. Dichos ejemplos son mostrados con el fin de probar el desempeño del algoritmo con diferentes conjuntos de imágenes. Finalmente, en la Sección 5, se presentan las conclusiones de este trabajo.

2. Antecedentes

Debido a los diversos campos de aplicación de la fusión de imágenes multi-foco, este tema se ha vuelto popular entre los investigadores del área de procesamiento de imágenes. Li *et al.* en (Li et al., 2017) presentan una gráfica que muestra el incremento en la cantidad de artículos relacionados con este tema en los últimos años. Dentro de la literatura relacionada con el tema encontramos propuestas de solución que van desde aquellas en las que se propone calcular la imagen fusionada como un promedio de las imágenes fuente, hasta técnicas en las que se propone el uso de evolución diferencial (Aslantas and Kurban, 2010) y otros trabajos que utilizan redes neuronales artificiales para el cálculo del mapa de decisión (Li et al., 2002; Zhou et al., 2006; Ma et al., 2011; Pagidimaray and Babu, 2011).

Para calcular la imagen fusionada como un promedio pesado de las dos imágenes fuente, se puede utilizar (1), en donde los pesos aplicados pueden ser $w_1(r, c) = w_2(r, c) = 0.5$.

$$I_F(r, c) = w_1(r, c)I_1(r, c) + w_2(r, c)I_2(r, c) \quad (1)$$

Aunque la técnica planteada en (1) es simple, se puede decir que algunas técnicas de la literatura están basados en esta idea, ya que en la mayoría de estas técnicas se calcula un peso $w_k(r, c)$ variable, que se asigna a cada píxel de la imagen I_k . En algunos casos dichos pesos son binarios y en general son valores reales entre 0 y 1.

En el cálculo de los pesos que se asignan a cada píxel de las imágenes se han utilizado diferentes técnicas, de las cuales

podemos mencionar algunas que han tenido la mayor aceptación. Existen propuestas en las que se utilizan redes neuronales para la fusión de imágenes (Li et al., 2002; Zhou et al., 2006; Ma et al., 2011; Pagidimarry and Babu, 2011). En algunos otros trabajos se propone tratar el problema como un proceso de optimización para el cálculo de los pesos que se asocian a cada píxel de las imágenes (Calderon and Garnica, 2014) y (Calderon et al., 2016).

En otros trabajos se propone usar la transformada Wavelet para determinar las regiones de cada imagen que deben conformar la imagen final (Li et al., 1995; Zhang and Blum, 1999; Pajares and de la Cruz, 2004; Shi et al., 2005; Li and Yang, 2008a; Malviya and Bhirud, 2009; Li et al., 2010; Tian and Chen, 2010; Yang, 2011; Shah et al., 2013; Yang et al., 2014).

También existen enfoques en los que se utiliza la transformada Contourlet (Zhang and long Guo, 2009; Shah et al., 2013; Yang et al., 2015). Orozco (Orozco, 2013) propone el cálculo de una matriz de pesos calculada con ayuda del algoritmo de segmentación ECQMMFM propuesto en (Rivera et al., 2007). Cao et al. (Cao et al., 2015) proponen el uso de la Transformada Discreta Coseno y un procesamiento por bloques en el proceso de fusión.

2.1. Trabajos basados en mapas de decisión

Existen algunas propuestas en la literatura donde se calcula el mapa de decisión. El primer trabajo del que tenemos referencia donde se estima un mapa de decisión usando transformada Wavelet, es el presentado por Li et al. en (Li et al., 1995). Zhang and Blum en (Zhang and Blum, 1999) hacen una descomposición multi-escala y utilizan la transformada discreta Wavelet y un procesamiento por ventanas para determinar los pesos asociadas a cada imagen. Li et al. en (Li et al., 2010) calculan un mapa de decisión usando la Transformada Redundante Wavelet.

Li et al. en (Li et al., 2013b) propone calcular un mapa de decisión inicial y aplicar operadores morfológicos para evitar la aparición de objetos no deseados en las regiones de transición entre áreas enfocadas y desenfocadas.

Yang et al. en (Yang et al., 2014) realizan el cálculo del mapa de decisión usando transformada Wavelet y calculando la energía en una vecindad con un kernel de Sobel. Zhou et al. en (Zhou et al., 2014), también proponen el cálculo del mapa de decisión usando la información del gradiente a diferentes escalas.

Yang et al. (Yang et al., 2015) calculan un mapa de decisión usando la transformada Contourlet y al final aplican una verificación de consistencia con procesamiento morfológico. Calderón et al. en (Calderon and Garnica, 2014) y (Calderon et al., 2016) proponen también un mapa de decisión el cual es calculado planteando el problema como un proceso de optimización con restricciones.

En trabajos publicados por (Li et al., 1995, 2010, 2013b; Zhou et al., 2014; Calderon and Garnica, 2014; Yang et al., 2015) se calcula un mapa de decisión. Se propone el mejoramiento de dicho mapa usando técnicas de relleno de regiones no conectadas, operadores morfológicos o técnicas que garanticen la coherencia espacial.

Uno de los trabajos publicados recientemente por Zhan et al. plantea un método de solución para el problema de fusión de imágenes multi-foco (Zhang et al., 2016a) y lo compara con los

métodos propuestos por (Chai et al., 2011), (Li et al., 2013a), (Duan et al., 2014), (Luo et al., 2012). A pesar de reportar mejores resultados el tiempo de ejecución es mayor. Posteriormente Zhan et al., en (Zhang et al., 2016b) propone un método donde las imágenes son descompuestas para extraer la información de la textura y utiliza un sistema de ventanas deslizantes para realizar la fusión de las imágenes. Los resultados reportados son buenos, sin embargo, los tiempos reportados no permiten la aplicación del algoritmo en tiempo real. En (Kong and Lei, 2017) se usa un sistema de ventanas deslizantes y diccionarios de descomposición espacial para lograr la fusión de imágenes. En dicho trabajo no se reportan los mapas de decisión para las imágenes y los tiempos de ejecución reportados son superiores a 7 minutos. Uno de los trabajos más prominentes en la actualidad es el reportado en (Liu et al., 2017a) donde se obtiene un mapa de decisión inicial al que se aplica una verificación de consistencia para obtener el mapa de decisión final. Los tiempos de ejecución reportados en este trabajo son satisfactorios (Liu et al., 2017a), aunque no superan la calidad de la imagen fusionada reportada en otros trabajos que hemos revisado. En (Liu et al., 2017b) se realiza un proceso de descomposición de la imagen para encontrar las regiones enfocadas y realizar la fusión, sin embargo, los tiempos de ejecución son superiores a 19 segundos para las imágenes reportadas. En (Aslantas and Toprak, 2017) se ve el problema como un proceso de optimización en el que se trata de encontrar las regiones desenfocadas de cada imagen fuente, aunque los resultados son buenos en varios casos, son superados por otras propuestas y se reportan tiempos de ejecución superiores a 1 segundo para imágenes de 256×256 .

De los trabajos listados en esta sub-sección, hay propuestas que ofrecen soluciones con muy buena exactitud, pero a costa de una alta complejidad en tiempo mientras que otras propuestas ofrecen encontrar la solución en un tiempo razonablemente corto, pero sacrificando parte de la exactitud a cambio de un mejor desempeño en tiempo. En general, en muchos de los problemas en los que la solución involucra el uso de recursos computacionales se tiene un compromiso entre el tiempo requerido para encontrar la solución y la exactitud de la misma.

2.2. Trabajos basados en procesamiento por bloques o por píxeles

En las propuestas de solución para el problema de fusión de imágenes multi-foco se distinguen dos técnicas de procesamiento de las imágenes la primera técnica plantea procesar cada píxel de las imágenes de manera independiente; (Piella, 2003; Lewis et al., 2007; Aslantas and Toprak, 2014) son algunos de ellos y otros más se analizan en (Li et al., 2017). La segunda técnica propone resolver el problema tomando bloques de píxeles en lugar de tratarlos de forma independiente (Zhang and Blum, 1999; Li et al., 2001, 2002; Zhao et al., 2008; Aslantas and Kurban, 2010; Pagidimarry and Babu, 2011; De and Chanda, 2013; Qu et al., 2014; Bai et al., 2015; Cao et al., 2015; Calderon et al., 2016).

Otra técnica que se ha planteado en la fusión de imágenes consiste en procesar las imágenes segmentándolas primero en regiones homogéneas que pertenecen al mismo objeto de la escena y por ende se deben extraer de la misma imagen multi-foco, (Li and Yang, 2008b; Pramanik et al., 2013).

Existen propuestas como las publicadas en (De and Chanda, 2013) y (Bai et al., 2015), en donde se implementa un Quad-Tree. La técnica de Quad-Tree consiste en dividir la imagen en cuatro bloques, procesar cada uno de ellos y medir el nivel de enfoque de cada bloque. Si el nivel de enfoque no supera un umbral se repite el proceso hasta llegar a un tamaño mínimo de bloque. El uso de Quad-Trees es equivalente a tener tamaños de bloque variables, procesando con bloques grandes en algunas regiones de la imagen y con bloques pequeños en las regiones donde no se tiene la certeza de que la fusión sea correcta. A diferencia de Quad-Trees donde se utilizan bloques proponemos resolver el problema de fusión de imágenes a nivel de píxeles. En nuestra propuesta cada píxel de la imagen tendrá una ventana de tamaño variable la cual se mejora en un proceso iterativo con información de bordes para limitar su tamaño. Dichas diferencias nos permiten lograr una mejor definición en los bordes y en consecuencia una mejor segmentación.

Las soluciones basadas en el tratamiento de los píxeles de forma independiente ofrecen buena definición en los bordes de los objetos que aparecen en la escena, pero se sacrifica parte de la coherencia espacial o consistencia de la solución, por lo que es común hacer un proceso de verificación de consistencia sobre el mapa de decisión calculado. Por otro lado, las soluciones en las que se hace el tratamiento de las imágenes a nivel de bloque, no tienen el problema de la pérdida de coherencia espacial pero a cambio se sacrifica la exactitud de la segmentación en los bordes de los objetos.

2.3. Medición del nivel de enfoque de una imagen

En la literatura es común medir el nivel de enfoque de las imágenes I_1 e I_2 a través de una función F y generar una imagen fusionada I_F píxel a píxel utilizando (2).

$$I_F(r, c) = \begin{cases} I_1(r, c) & \text{Si } F(I_1(r, c)) > F(I_2(r, c)) \\ I_2(r, c) & \text{En caso contrario} \end{cases} \quad (2)$$

La regla dada por (2), puede dar lugar a una forma de calcular una matriz de pesos o mapa de decisión como se indica en (3). Estos pesos son atómicos, de tal suerte que píxeles adyacentes correspondientes a un mismo objeto pueden presentar valores diferentes de pesos. Por esta razón es necesario extender la regla a regiones o bloques de la imagen.

$$P(r, c) = \begin{cases} 1 & \text{Si } F(I_1(r, c)) > F(I_2(r, c)) \\ 0 & \text{En otro caso} \end{cases} \quad (3)$$

Cuando se hace un procesamiento a nivel de bloque, la regla de fusión es similar a la mostrada en (4).

$$C_i = \begin{cases} A_i & \hat{F}(A_i) > \hat{F}(B_i) + TH \\ B_i & \hat{F}(B_i) > \hat{F}(A_i) + TH \\ \frac{A_i+B_i}{2} & \text{En otro caso.} \end{cases} \quad (4)$$

donde C_i es el i -ésimo bloque de la imagen fusionada, A_i y B_i son los i -ésimos bloques de píxeles correspondientes a las imágenes I_1 e I_2 respectivamente, TH es un umbral (en algunos casos es 0) y \hat{F} es una función que calcula el nivel de enfoque de un bloque.

Algunos trabajos como (Li et al., 2001; Pajares and de la Cruz, 2004; Zhou et al., 2006; Li and Yang, 2008b; Cao et al.,

2015), proponen el uso de la frecuencia espacial para calcular el nivel de actividad o enfoque de un bloque o píxel. Dicha técnica de medición del nivel de actividad fue propuesta por Eskicioglu and Fisher en (Eskicioglu and Fisher, 1995).

La frecuencia espacial y algunos otros métodos para medir el nivel de enfoque de una imagen están basados en primeras y segundas derivadas o en los momentos centrales. Dichos métodos miden de alguna u otra manera el nivel de variación de la imagen en una región, tomando esa medición como nivel de enfoque o nivel de actividad. Esta teoría está basada en la premisa de que la información contenida en las regiones enfocadas de una imagen es de alta frecuencia y por ende es altamente variante mientras que la información contenida en las regiones desenfocadas, es de baja frecuencia y por ende hay poca variación entre un píxel y sus vecinos.

Algunos filtros pasa-altas también son utilizados como medidores de enfoque de una imagen. Riaz *et al.* (Riaz et al., 2008) plantean que los filtros pasa-altas son muy efectivos para medir el nivel de enfoque de una imagen. También explican que una forma de filtrar las frecuencias altas es utilizar el operador Laplaciano y sus variantes como la diferencia de Gaussianas (DOG), la cual de acuerdo a lo planteado en (Assirati et al., 2014) es una muy buena aproximación al Laplaciano. La diferencia de Gaussianas es usada en (Orozco, 2013) como medida de nivel de enfoque de las imágenes.

Una forma de aproximar el Laplaciano es utilizar el kernel de convolución mostrado en (5). Dicho kernel puede ser visto como la medición del nivel de actividad o cálculo de las primeras derivadas de la imagen en las 8 direcciones posibles. Este kernel se presenta en la literatura como una discretización del Laplaciano y se utiliza para medir la calidad o enfoque de la imagen, (Riaz et al., 2008; Calderon and Garnica, 2014; Calderon et al., 2016).

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (5)$$

Aunque existen diversas maneras de medir la claridad o nitidez de una imagen, las más comunes en la literatura son aquellas basadas en: frecuencia espacial, energía del Laplaciano, derivadas, varianza, momentos centrales y entropía.

En este trabajo decidimos utilizar la representación del Laplaciano discretizado por la simplicidad en su cálculo y por ser un buen estimador del nivel de enfoque.

A diferencia de los trabajos citados en ésta sección, en nuestra propuesta: utilizamos un medidor simple de nitidez, realizamos el procesamiento con ventanas de tamaño adaptable que integra las ventajas de procesar por bloque y píxel a píxel, eliminamos la necesidad de hacer la verificación de consistencia sobre el mapa de decisión, obtenemos un resultado que podría permitir su aplicación en tiempo real y presentamos la generalización para fusionar conjuntos con más de dos imágenes multi-foco.

3. El Algoritmo CLI-VV

Calderón *et al.* en (Calderon et al., 2016) presentan una técnica para crear un par de imágenes multi-foco sintéticas (I_1

e I_2), a partir de una imagen completamente nítida I_0 y un mapa de decisión binario P_0 que tiene 1 en las regiones que se desea desenfocar para generar I_1 y 0 en aquellos píxeles que se desea desenfocar en I_2 . El proceso planteado en (Calderon et al., 2016) consiste en generar una imagen completamente desenfocada J_0 convolucionando una Gaussiana $G_\sigma(r, c)$ con media cero y varianza σ (6) con la imagen nítida I_0 y generar las imágenes I_1 e I_2 como una combinación de la imagen desenfocada J_0 y la enfocada I_0 aplicando (7). Con esta técnica, las imágenes generadas contienen regiones nítidas y regiones desenfocadas de forma complementaria, por lo que los objetos de interés aparecerán nítidos en por lo menos una de las imágenes.

$$J_0(r, c) = I_0(r, c) * G_\sigma(r, c) \quad (6)$$

$$\begin{aligned} I_1(r, c) &= J_0(r, c)P_0(r, c) + I_0(r, c)(1 - P_0(r, c)) \\ I_2(r, c) &= J_0(r, c)(1 - P_0(r, c)) + I_0(r, c)P_0(r, c) \end{aligned} \quad (7)$$

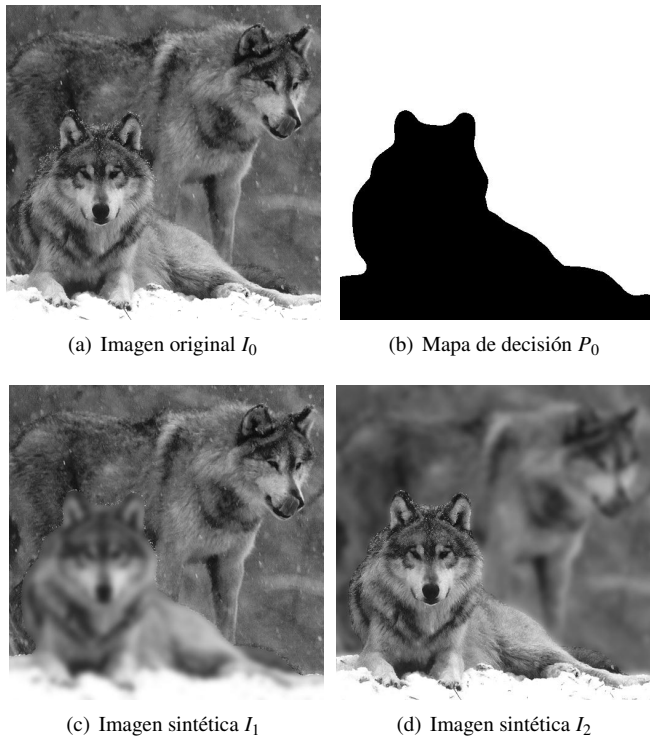


Figura 2: Simulación de pérdida de nitidez dada una imagen I_0 y un mapa de decisión P_0

En la Figura 2 se muestran un ejemplo de imágenes multi-foco sintéticas generadas con (7). La Fig. 2(a) es la imagen completamente nítida I_0 , la Fig. 2(b) es el mapa de decisión P_0 utilizado y las Figs. 2(c) y 2(d) son las imágenes sintéticas generadas I_1 e I_2 respectivamente.

Calderón *et al.* definen el problema de fusión de imágenes multi-foco como un proceso inverso al de creación de imágenes sintéticas en el que se requiere encontrar un mapa de decisión que nos permita recortar las regiones enfocadas de cada una de las imágenes multi-foco del conjunto. En la Figura 3 se muestra de manera gráfica como se realiza el proceso de fusión cuando se tiene dos imágenes multi-foco I_1 e I_2 (Figs. 3(a) y 3(d), respectivamente) así como un mapa de decisión P y su complemento P' (Figs. 3(b) y 3(e)). Las Figs. 3(c) y 3(f) muestran la

parte nítida segmentada de las imágenes I_1 e I_2 y la Fig. 3(g) el resultado de unir las imágenes de las Figuras 3(c) y 3(f) lo cual se puede representar matemáticamente con (8).

$$I_F(r, c) = P(r, c)I_1(r, c) + (1 - P(r, c))I_2(r, c) \quad (8)$$

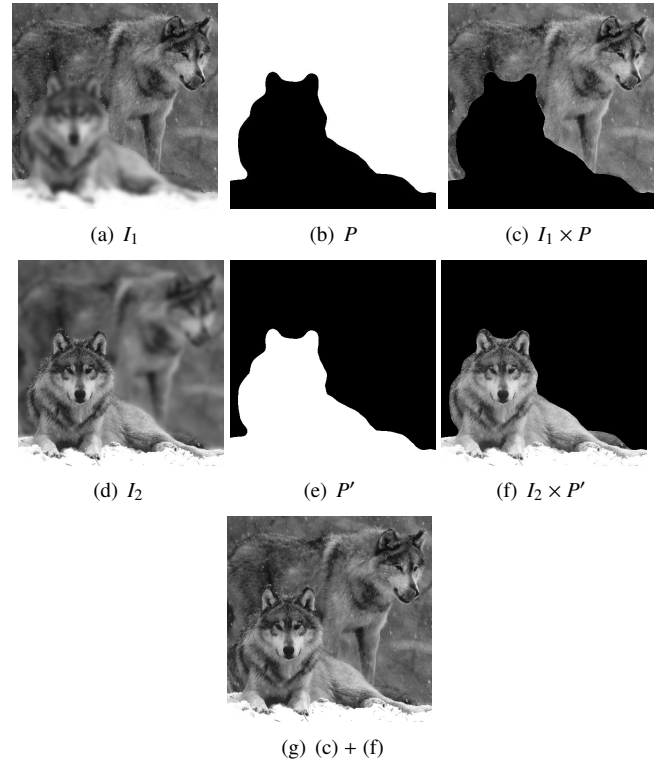


Figura 3: Ejemplo de Fusión de dos imágenes multi-foco dado un mapa de decisión

El problema de fusión de imágenes multi-foco puede ser planteado como un proceso inverso al de creación de las imágenes sintéticas. Entonces el principal problema para la fusión de imágenes multi-foco es calcular el mapa de decisión P que nos permita realizar la combinación de las mismas.

Para calcular el valor de del mapa de decisión $P(r, c)$ en cada una de las coordenadas (r, c) de la imagen, agregando restricciones de coherencia espacial, (Calderon et al., 2016) proponen optimizar el sistema dado por (9) en una región o ventana de tamaño $(2w + 1) \times (2w + 1)$ para valores de $w \in \{0, 1, 2, 3 \dots\}$.

$$\operatorname{argmax}_{P(r, c)} E_s(P(r, c)) = P(r, c)S(r, c, w) \quad (9)$$

$$\text{s.a} \\ 0 \leq P(r, c) \leq 1$$

$$S(r, c, w) = \sum_{k=-w}^w \sum_{l=-w}^w \Delta F(r+k, c+l) \quad (10)$$

donde $E_s(P(r, c))$ es la función objetivo, $S(r, c, w)$ es la suma de $\Delta F(r, c) = F_1(r, c) - F_2(r, c)$ alrededor de una ventana y $F_k(r, c)$ es una medida de la nitidez de la imagen I_k en las coordenadas (r, c) .

Algoritmo 1 CLI-S(I_1, I_2, w)

- 1: $F_k = |I_k * h|, \forall k \in \{1, 2\}$
- 2: $\Delta F(r, c) = F_1(r, c) - F_2(r, c)$
 $\forall(r, c) \in \{1, \dots, n_r\} \times \{1, \dots, n_c\}$
- 3: $\Delta F_{inc}(r, c) = \sum_{k=1}^r \sum_{l=1}^l \Delta F(k, l)$
 $\forall(r, c) \in \{1, \dots, n_r\} \times \{1, \dots, n_c\}$
- 4: **para** $r = 1$ **hasta** n_r **hacer**
- 5: **para** $c = 1$ **hasta** n_c **hacer**
- 6: $S(r, c, w) = \Delta F_{inc}(r + w, c + w) - \Delta F_{inc}(r + w, c - w - 1) - \Delta F_{inc}(r - w - 1, c + w) + \Delta F_{inc}(r - w - 1, c - w - 1)$
- 7: **si** $S(r, c, w) \geq 0$ **entonces**
- 8: $P(r, c) = 1$
- 9: **si no**
- 10: $P(r, c) = 0$
- 11: **fin si**
- 12: **fin para**
- 13: **fin para**
- 14: Construir I_F utilizando (8)
- 15: **devolver** I_F

La solución para (9), se lleva a cabo utilizando el Algoritmo 1 llamado Combinación Lineal de Imágenes Simple (CLI-S) publicado en (Calderon et al., 2016) y para calcular $S(r, c, w)$ (10) se utiliza imágenes integrales (Viola and Jones, 2001). El Algoritmo CLI-S, recibe como parámetros las dos imágenes a fusionar y el valor para w , el cual es constante en todas las coordenadas de la imagen.

En las Figuras 4(a), 4(b), 4(c) y 4(d), se muestran los mapas de decisión calculados al aplicar el Algoritmo CLI-S, con $w = 0, w = 15, w = 30$ y $w = 45$ respectivamente, para el ejemplo de las imágenes sintéticas mostradas en la Fig. 2. En las imágenes de la Figura 4(a) se observa que con tamaño de ventana pequeño se logra una buena definición en los bordes de segmentación pero se tiene poca coherencia espacial, en la Figura 4(b) se tiene suficiente coherencia espacial con bordes que se aproximan a los bordes reales en las imágenes, mientras que con tamaño de ventana grande Figs. 4(c) y 4(d), se tiene coherencia espacial pero deslocalización de los bordes del mapa de decisión con respecto a los bordes de las imágenes originales.

Podemos notar, que el mapa de decisión de la Figura 4, tiene diferentes formas para diferentes valores de w y que debe existir un valor w tal que minimice las diferencias entre el mapa de decisión calculado P y el mapa de decisión original P_0 , con el que fueron generadas las imágenes I_1 e I_2 . Sin embargo, en el Algoritmo CLI-S el valor de w es seleccionado manualmente por el usuario, hasta tener el mejor mapa de decisión.

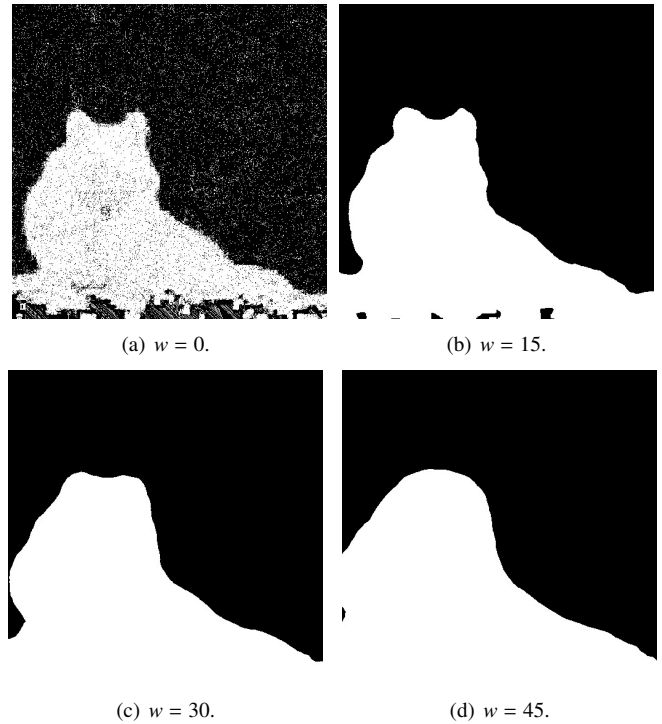


Figura 4: Mapas de decisión calculados con el algoritmo CLI-S para cuatro diferentes tamaños de ventana.

En la Figura 5 se muestra el comportamiento del porcentaje de aciertos obtenidos con el algoritmo CLI-S (Alg. 1) para el ejemplo de las imágenes de los lobos. Note que el valor que maximiza dicho porcentaje es $w = 15$. Sin embargo, este valor w óptimo sólo puede ser obtenido cuando se tiene un valor de referencia P_0 , que en conjuntos de imágenes multi-foco obviamente no existe. Este valor óptimo para w cambiará para cada par de imágenes multi-foco.

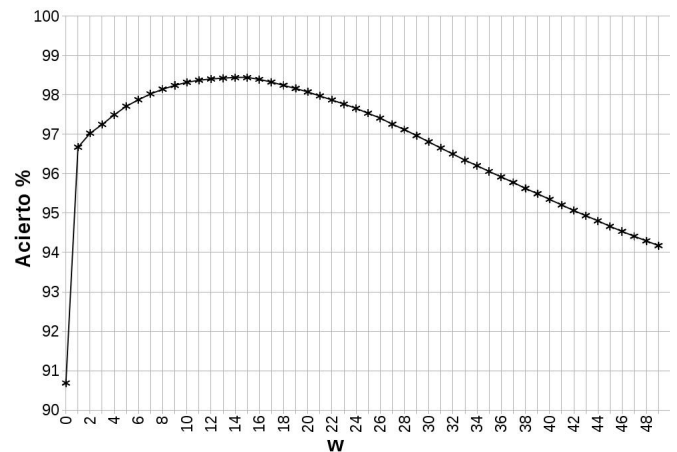


Figura 5: Porcentaje de acierto del Algoritmo CLI-S variando el valor de w .

Lo anterior nos lleva a pensar que lo ideal es calcular automáticamente el tamaño de las ventanas en cada uno de los píxeles. Este tamaño debe ser una función de la distancia del píxel a los bordes del mapa de decisión, los cuales deberían coincidir con las fronteras entre las regiones nítidas y borrosas de la imagen.

3.1. Cálculo eficiente del tamaño de la ventana

De acuerdo con lo mencionado anteriormente, proponemos estimar el tamaño óptimo de la ventana a nivel píxel, tal que cumpla con los objetivos de garantizar la coherencia espacial en las regiones pertenecientes a un mismo objeto de la escena y la correcta segmentación en los bordes de las regiones nítidas.

Definiremos una ventana $V(W(r, c))$ como el conjunto de las coordenadas (r, c) asociadas a cada píxel dentro de una región cuadrada. Esta ventana estará centrada en las coordenadas (r, c) con área $A(W(r, c)) = (2W(r, c) + 1)^2$ y el valor $W(r, c)$ es un número entero positivo que debemos calcular.

Definimos un arreglo B , con la información de los bordes del mapa de decisión P . El valor de B en cada una de las coordenadas (r, c) , será igual a 1 en caso de existir un borde y 0 si no existe; así dado P podemos calcular B mediante (11). Inicialmente asumiremos que el mapa de decisión contiene ceros $P^{(0)}(r, c) = 0$ y por lo tanto la matriz de bordes también tendrá ceros $B^{(0)}(r, c) = 0$ en cada una de las coordenadas de la imagen.

$$B(r, c) = \begin{cases} 1 & |P(r, c) - P(r-1, c)| > 0 \\ 1 & |P(r, c) - P(r, c-1)| > 0 \\ 0 & \text{Si no.} \end{cases} \quad (11)$$

Dada una matriz de bordes B , nuestra propuesta es maximizar el área de la ventana A asociada a cada coordenada (r, c) , limitada a que la cantidad de puntos del borde dentro de la ventana sea menor a un valor T y que el área de la ventana sea menor o igual a $(2w_{max} + 1)^2$. Esto lo podemos plantear como un proceso de optimización dado por (12). Permitir que el valor mínimo de $W(r, c)$ sea 0 en (12), hace posible la creación de ventanas de 1×1 en las regiones cercanas a los bordes, lo cual es equivalente a hacer un procesamiento a nivel píxel y limitar $W(r, c) \leq w_{max}$ evita que las ventanas crezcan indefinidamente ya que inicialmente no se tiene información de los bordes de segmentación.

$$\operatorname{argmax}_{W(r, c)} A(W(r, c)) = (2W(r, c) + 1)^2 \quad (12)$$

s.a

$$0 \leq W(r, c) \leq w_{max} \quad (13)$$

$$\sum_{\langle k, l \rangle \in V(W(r, c))} B(k, l) < T \quad (14)$$

Una forma de encontrar los valores que maximizan (12), es aplicar un proceso iterativo, donde $W(r, c)$ crece desde un valor inicial 0 hasta un valor w_{max} mientras no se rebasa la tolerancia T en los bordes del mapa de decisión (14). El valor final de $W(r, c)$ óptimo será aquel que maximiza el área de la ventana, sin que los bordes sobrepasen la tolerancia T , tal como se muestra en la imagen de la Figura 6. En dicha figura se muestran dos ventanas centradas en las coordenadas (r, c) , con áreas $(2w_{max} + 1)^2$ y $(2W(r, c) + 1)^2$. Se puede observar que la ventana mayor cumple con la restricción dada por (13), pero abarca gran cantidad de puntos del borde, violando la restricción dada por (14). La ventana interna, es la ventana más grande que se puede crear sin violar ninguna de las restricciones.

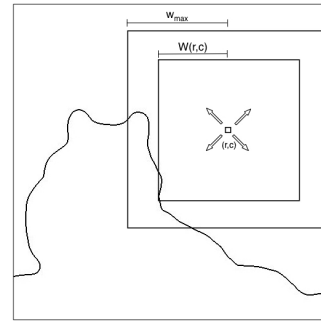


Figura 6: Ejemplo del proceso de maximización del área de la ventana a partir de un punto en las coordenadas (r, c)

Para hacer eficiente la búsqueda de $W(r, c)$ que optimiza (12), proponemos utilizar búsqueda binaria y calcular la sumatoria de la restricción dada por (14) utilizando imágenes integrales (Viola and Jones, 2001) de acuerdo con (15) y (16).

$$B_{inc}(r, c) = \sum_{k=1}^r \sum_{l=1}^c B(k, l) \quad (15)$$

$$SB(r, c, w) = B_{inc}(r+w, c+w) - B_{inc}(r+w, c-w-1) - B_{inc}(r-w-1, c+w) + B_{inc}(r-w-1, c-w-1) \quad (16)$$

donde $B_{inc}(r, c)$ es la suma de todos los puntos de la matriz de bordes desde $(1, 1)$ hasta (r, c) y $SB(r, c, w)$ es la suma de los bordes dentro de la ventana.

En el Algoritmo 2 llamado de Ventana Óptima (VO) se resume lo planteado en los párrafos anteriores. El Algoritmo VO recibirá como parámetros: el valor máximo para w_{max} , la tolerancia T y un mapa de decisión P y regresará una matriz W con los valores que optimizan (12) para todas las coordenadas $(r, c) \in \{1, \dots, n_r\} \times \{1, \dots, n_c\}$.

Algoritmo 2 VO(w_{max}, T, P)

- 1: Hacer $a = 0$ y $b = w_{max}$.
 - 2: Calcular B , $\forall (r, c) \in \{1, \dots, n_r\} \times \{1, \dots, n_c\}$ con (11)
 - 3: Calcular B_{inc} con (15)
 - 4: **para** $r = 1$ **hasta** n_r **hacer**
 - 5: **para** $c = 1$ **hasta** n_c **hacer**
 - 6: **mientras** $b - a > 1$ **hacer**
 - 7: $w \leftarrow \lfloor \frac{a+b}{2} \rfloor$
 - 8: **si** $SB(r, c, w) \leq T$ **entonces**
 - 9: $a \leftarrow w$
 - 10: **si no**
 - 11: $b \leftarrow w$
 - 12: **fin si**
 - 13: **fin mientras**
 - 14: $W(r, c) \leftarrow w$
 - 15: **fin para**
 - 16: **fin para**
 - 17: **devolver** W
-

En la Figura 7 se muestran algunas ventanas calculadas con el Algoritmo VO centradas en sus píxeles correspondientes sobre un fondo negro para el par multi-foco de la Figura 2. Note,

para este ejemplo, que se obtuvieron ventanas de tamaño variable que resultan ser grandes en regiones homogéneas y pequeñas en regiones cercanas a los bordes. Las ventanas se traslapan ya que dos píxeles vecinos tienen ventanas de tamaño igual o diferente.

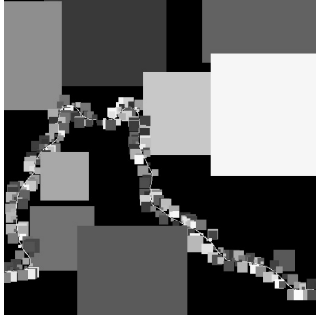


Figura 7: Tamaños de ventana obtenidos con el Algoritmo VO (Alg. 2).

Al inicio no se tiene información del mapa de decisión $P^{(0)}$, por lo que el Algoritmo VO dará como resultado ventanas con área $(2w_{max} + 1)^2$. Con esta información podemos calcular un mapa de decisión $P^{(1)}$, sin embargo, dicho mapa tendrá poca exactitud en los bordes. Para afinar nuestro mapa, proponemos aplicar nuevamente el Algoritmo VO, utilizando los bordes de $P^{(1)}$ y obtener los tamaños de ventana adecuados, que nos permiten calcular un nuevo mapa de decisión $P^{(2)}$, el cual deberá tener una mayor precisión en los bordes de segmentación. En general podemos repetir el proceso para calcular $P^{(k+1)}$ en función de los bordes de $P^{(k)}$, hasta que la diferencia entre $P^{(k)}$ y $P^{(k+1)}$ sea menor a un umbral o se supere un número máximo de iteraciones. El Algoritmo 3 al que denominaremos Combinación Lineal de Imágenes con Ventanas Variables (CLI-VV), es una modificación del Algoritmo 1 (CLI-S). Los principales cambios que se han hecho son: dar un valor inicial a $P^{(0)}$ (línea 4) e iterar el proceso de cálculo de $W^{(k)}$ con la información de $P^{(k-1)}$ utilizando el Algoritmo VO (Alg. 2 líneas 5 y 6), para posteriormente recalcular $P^{(k)}$ utilizando los valores de $W^{(k)}$.

Algoritmo 3 CLI-VV($I_1, I_2, w_{max}, T, N_{iter}$)

- 1: $F_k = |I_k * h|, \forall k \in \{1, 2\}$
 - 2: $\Delta F(r, c) = F_1(r, c) - F_2(r, c)$
 $\forall (r, c) \in \{1, \dots, n_r\} \times \{1, \dots, n_c\}$
 - 3: $\Delta F_{inc}(r, c) = \sum_{k=1}^r \sum_{l=1}^c \Delta F(k, l)$
 $\forall (r, c) \in \{1, \dots, n_r\} \times \{1, \dots, n_c\}$
 - 4: $P^{(0)}(r, c) \leftarrow 0 \forall (r, c) \in \{1, \dots, n_r\} \times \{1, \dots, n_c\}$
 - 5: **para** $k = 1$ **hasta** N_{iter} **hacer**
 - 6: $W^{(k)} \leftarrow VO(w_{max}, T, P^{(k-1)})$ (Alg. 2).
 - 7: **para** $r = 1$ **hasta** n_r **hacer**
 - 8: **para** $c = 1$ **hasta** n_c **hacer**
 - 9: **si** $S(r, c, W^{(k)}(r, c)) \geq 0$ **entonces**
 - 10: $P^{(k)}(r, c) = 1$
 - 11: **si no**
 - 12: $P^{(k)}(r, c) = 0$
 - 13: **fin si**
 - 14: **fin para**
 - 15: **fin para**
 - 16: **fin para**
 - 17: Calcular la imagen fusionada I_F aplicando (8)
 - 18: **devolver** $\{I_F, P^{(N_{iter})}\}$
-

El algoritmo CLI-VV($I_1, I_2, w_{max}, T, N_{iter}$) requiere de cinco parámetros, donde w_{max} debe ser un valor suficientemente grande para generar ventanas que abarquen entre un 25 % y un 50 % del tamaño total de la imagen, el valor de T indicará cuantos puntos del borde permitiremos por ventana y N_{iter} es el número de iteraciones que se harán para mejorar P . Es importante mencionar que se utilizan imágenes integrales en el Algoritmo 3 (línea 9), lo cual permite que el tamaño de la ventana, no influya en el tiempo de ejecución.

3.2. Fusión de conjuntos con más de dos imágenes multi-foco.

En la literatura, lo más común es encontrar trabajos en los que se presenta la fusión de dos imágenes multi-foco, sin embargo, existen situaciones en las que es necesario aplicar el procedimiento para conjuntos de imágenes multi-foco con más de dos imágenes $N > 2$. Con el propósito de fusionar conjuntos con más de dos imágenes, desarrollamos el Algoritmo 4, al que denominamos Fusión Multi-foco con Ventanas Variables (FM-VV). El Algoritmo FM-VV inicia fusionando las imágenes I_1 e I_2 con el Alg. CLI-VV para obtener $I_F^{(1)}$, posteriormente fusiona las imágenes $I_F^{(1)}$ e I_3 para obtener $I_F^{(2)}$ repitiendo el proceso hasta haber fusionado todas las imágenes del conjunto I . El mapa de decisión final se puede obtener sumando todos los mapas de decisión parciales $P_F = P_F^{(1)} + \dots + P_F^{(k)}$

Algoritmo 4 FM-VV($\{I_1, I_2, \dots, I_N\}, w_{max}, T, N_{iter}$)

- 1: $I_F^{(0)} \leftarrow I_1$.
 - 2: **para** $k = 1$ **hasta** $N - 1$ **hacer**
 - 3: $\{I_F^{(k)}, P_F^{(k)}\} \leftarrow \text{CLI-VV}(I_F^{(k-1)}, I_{(k+1)}, w_{max}, T, N_{iter})$.
 - 4: **fin para**
 - 5: $P_F = \sum_{k=1}^{N-1} P_F^{(k)}$.
 - 6: **devolver** $\{I_F^{(N-1)}, P_F\}$
-

3.3. Análisis de Complejidad

En esta sección presentamos el análisis de complejidad asintótica de los algoritmos propuestos en este artículo. En el análisis de complejidad suponemos que las imágenes a fusionar son de tamaño $n_r \times n_c$ píxeles.

El Algoritmo 1, CLI-S, recibe las dos imágenes y el tamaño de ventana calculada y regresa la imagen fusionada. Las líneas 1, 2 y 3 son todas de complejidad $O(n_r n_c)$, dado que barren matrices del mismo tamaño que las imágenes originales y se utilizan imágenes integrales para su cálculo, en la línea 3. El ciclo de la línea 4 se ejecuta $O(n_r)$ veces y el de la línea 5 $O(n_c)$ veces. El cuerpo del ciclo interno (líneas 6 a 11), es de complejidad constante, i.e, $O(1)$. Por consiguiente la complejidad del ciclo externo (líneas 4 a 13) es de complejidad $O(n_r n_c)$. La línea 14 puede ejecutarse en $O(n_r n_c)$. Esto da como resultado que la complejidad de CLI-S sea $O(n_r n_c)$.

El Algoritmo 2, VO, toma como argumentos el tamaño máximo de ventana, la tolerancia T (escalar), y el mapa de decisión P de tamaño $n_r \times n_c$. El Algoritmo VO regresa la matriz W de tamaño $n_r \times n_c$, la cual indica los tamaños de ventana óptima para cada píxel de las imágenes a fusionar. La línea 1 se puede ejecutar en tiempo constante $O(1)$. Las líneas 2 y 3 tienen una complejidad $O(n_r n_c)$. Los ciclos de las líneas 4 y 5 se ejecutan en $O(n_r)$ y $O(n_c)$ veces, respectivamente. El ciclo de las líneas 6 a 13 implementa la búsqueda binaria del tamaño óptimo de

ventana para el píxel en cuestión, lo cual tiene un costo computacional de $O(\log w_{max})$. El costo total del ciclo de 4 a 16 es de $O(n_r n_c \log w_{max})$. Por lo tanto, la complejidad total de VO es $O(n_r n_c \log w_{max})$.

El Algoritmo 3, CLI-VV toma como argumentos dos imágenes, I_1 e I_2 , el tamaño máximo de la ventana w_{max} , la tolerancia T y el número máximo de iteraciones permitidas para refinar los tamaños de ventana N_{iter} ; calcula y regresa la imagen fusionada y el mapa de decisión. Las primeras cuatro líneas son todas de complejidad $O(n_r n_c)$. El ciclo externo (línea 5) se ejecuta $O(N_{iter})$ veces. La línea 6 ejecuta el Algoritmo VO, el cual ya se demostró que es de complejidad $O(n_r n_c \log w_{max})$. Los ciclos de las líneas 7 y 8 se ejecutan $O(n_r)$, $O(n_c)$ veces, respectivamente. El cuerpo del ciclo interno es de complejidad constante, esto hace que el ciclo de las líneas 7 a 15 sea de complejidad $O(n_r n_c)$ y el ciclo de las líneas 5 a la 16 sea $O(N_{iter} n_r n_c \log w_{max})$. La línea 17 es de complejidad $O(n_r n_c)$. La complejidad del Algoritmo CLI-VV es $O(N_{iter} n_r n_c \log w_{max})$.

El Algoritmo FM-VV toma como argumentos un arreglo de N imágenes a fusionar, w_{max} , T y N_{iter} y regresa las imágenes fusionadas y el mapa de decisión. La línea 1 es de complejidad $O(n_r n_c)$. El ciclo de las líneas 2 a 4 se repite $O(N)$ veces y su cuerpo (línea 3) tiene una complejidad de $O(N_{iter} n_r n_c \log w_{max})$ por lo que la complejidad de dicho ciclo es $O(N N_{iter} n_r n_c \log w_{max})$. La línea 5 es de complejidad $O(N n_r n_c)$. La complejidad computacional de FM-VV es $O(N N_{iter} n_r n_c \log w_{max})$.

Si consideramos que w_{max} es una constante independientemente del tamaño de las imágenes, al igual que N_{iter} la cual se ve en los resultados que no pasa de 10 y N es mucho menor que el tamaño de las mismas, podemos considerar que la complejidad de VO, CLI-VV y FM-VV son todas $O(n_r n_c)$.

Si $n_p = n_r \times n_c$ es el número de píxeles en la imagen, la complejidad de todos los Algoritmos presentados en este artículo es lineal con el número de píxeles $O(n_p)$, es decir con el tamaño del problema.

4. Resultados

En esta sección se presentan los resultados de aplicar los algoritmos CLI-VV y FM-VV en pares y tríos de imágenes multi-foco. Primero presentamos la fusión de 21 pares de imágenes multi-foco sintéticas con el fin de hacer una medición cuantitativa del resultado obtenido con el Algoritmo CLI-VV y verificar su efectividad respecto al Algoritmo CLI-S. Posteriormente presentamos los resultados con 29 pares de imágenes multi-foco reales con el propósito de que el lector califique el desempeño del algoritmo CLI-VV de manera cualitativa y cuantitativa. Finalmente se presentan las soluciones obtenidas con el algoritmo FM-VV para 5 tríos de imágenes multi-foco.

Para fusionar la imagen final mediante (8) y dado que los mapas de decisión terminan siendo una etiqueta del píxel más nítido, utilizaremos cuatro métricas de segmentación para determinar la precisión de los Algoritmos CLI-VV y FM-VV. Las métricas que reportamos son Precisión a Nivel Pixel (Pixel Accuracy PA), Media de la Precisión a Nivel Pixel (Mean Pixel Accuracy MPA), Media de la Intersección sobre la Unión (Mean Intersection over Union MIU) y Frecuencia Pesada de la

Intersección sobre la Unión (Frequency Weighted Intersection over Union FWIU), las cuales define (Long et al., 2014) como (17).

$$\begin{aligned} PA &= \frac{\sum_{i=1}^N n_{i,i}}{\sum_{i=1}^N t_i} \\ MPA &= \frac{1}{N} \sum_{i=1}^N \frac{n_{i,i}}{t_i} \\ MIU &= \frac{1}{N} \sum_{i=1}^N \frac{n_{i,i}}{t_i + \sum_{j=1}^N n_{j,i} - n_{i,i}} \\ FWIU &= \left(\sum_{k=1}^N t_k \right)^{-1} \sum_{i=1}^N \frac{t_i n_{i,i}}{t_i + \sum_{j=1}^N n_{j,i} - n_{i,i}} \end{aligned} \quad (17)$$

donde $n_{i,j}$ es el número de píxeles de la clase i que fueron predichos como clase j y $t_i = \sum_{j=1}^N n_{i,j}$ es el total de los píxeles de la clase i

4.1. Resultados del Algoritmo CLI-VV en pares de imágenes multi-foco

Para hacer la evaluación cuantitativa del Algoritmo CLI-VV se creó un conjunto de 21 pares de imágenes sintéticas y se bajaron de la red 29 imágenes reales de las cuales no existe un mapa de decisión. Para las 21 imágenes sintéticas 4 fueron creadas usando mapas de decisión generados por nosotros y 17 fueron tomadas de la página Pascal VOC. La página Pascal VOC tiene imagen de referencia y un mapa de segmentación que los usuarios utilizan para probar sus algoritmos de Segmentación. En el caso de las imágenes sintéticas a priori sabemos el mapa de decisión y en el caso de las imágenes reales se determinaron los mapas de decisión de referencia (ground truth) de manera manual, separando las regiones nítidas en cada una de las imágenes multi-foco (utilizando el programa Gimp).

4.1.1. Pares de Imágenes Sintéticas

Para mostrar la mejora en el desempeño del algoritmo CLI-VV sobre el algoritmo CLI-S creamos cuatro pares de imágenes multi-foco sintéticas a partir de las imágenes de los lobos (Fig. 2(a)) y Lena (Fig. 10(a)). Para crear estos pares de imágenes sintéticas utilizamos tres diferentes mapas de decisión, el primero coincide con el contorno de uno de los dos lobos (Fig. 8(a)), el segundo con el contorno de Lena (Fig. 8(b)) y el tercero es una estrella de 18 picos (Fig 8(c)). El mapa de decisión de la estrella se utilizó tanto en la imagen de los lobos como en el de Lena para que el lector pueda comparar el desempeño de los algoritmos con un mapa de decisión que no coincide con los bordes de los objetos en las imágenes.

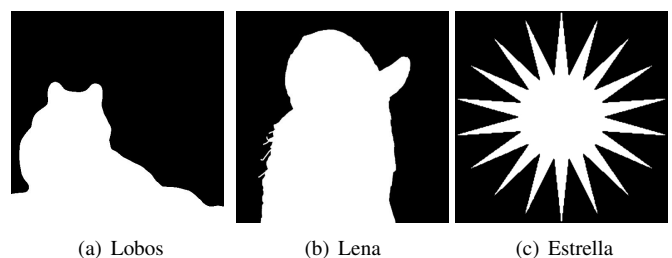


Figura 8: Mapas de decisión utilizados para generar pares sintéticos de imágenes multi-foco

En las imágenes de la Figura 9 se muestran algunas de las iteraciones del Algoritmo CLI-VV (Alg. 3) aplicado a las imágenes sintéticas de la Figura 2. Los parámetros utilizados fueron $w_{max} = 128$, $T = 1$ y $N_{iter} = 10$. En las Figuras 9(a-d) se muestran los bordes obtenidos en cada iteración, en las Figuras 9(e-h) se muestran los valores de W para cada iteración, ilustrados con tonos de gris, donde el valor máximo corresponde a color blanco y el mínimo al color negro y finalmente las Figuras 9(i-l) muestran la evolución del mapa de decisión P . Cada columna de la Figura corresponde a las iteraciones 1, 2, 3 y 10, respectivamente. Dado que inicialmente no se tiene información de los bordes de segmentación, los tamaños de la ventana tomarán un valor constante w_{max} en todas las coordenadas de la imagen. Podemos notar que en la primer iteración el mapa de decisión será equivalente a la solución del Algoritmo 1, (Calderon et al., 2016) para un valor $w = w_{max}$.

A medida que se itera, $W(r, c)$ tomará valores pequeños cerca de los bordes, permitiendo mejorar la segmentación y valores cercanos a w_{max} en las regiones homogéneas, los cuales maximizan la coherencia espacial. Esto mejora el mapa de decisión y en consecuencia la nitidez de la imagen fusionada.

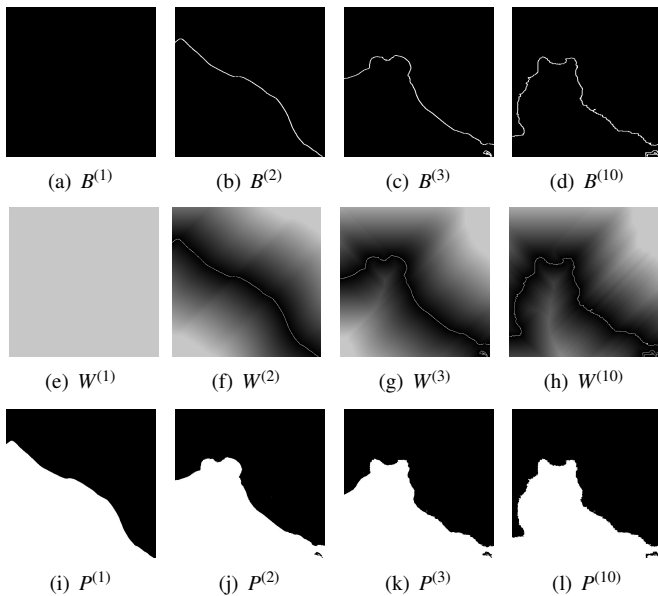


Figura 9: Evolución de los bordes B , tamaños de ventana W y del mapa de decisión P para las imágenes sintéticas de los lobos.

En la Figura 10, se presenta un segundo par de imágenes sintéticas generado a partir de la imagen mostrada en la Figura 10(a), utilizando (7) y el mapa mostrado en 10(b).

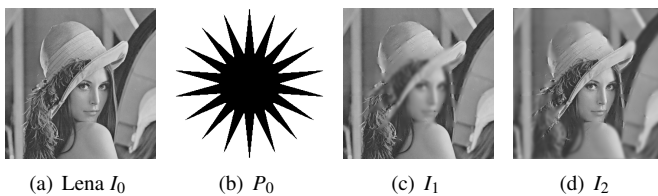


Figura 10: Simulación de pérdida de nitidez utilizando la imagen de Lena y un mapa de decisión con forma de estrella

Se ha seleccionado un mapa de decisión P_0 con forma de es-

trella con 18 picos con el objetivo de mostrar el desempeño del Algoritmo CLI-VV con mapas de decisión con regiones puntiagudas.

En la Figura 11 se muestran los resultados de algunas iteraciones del Algoritmo CLI-VV cuando se utilizan las imágenes mostradas en las Figuras 10(c) y 10(d). Para la Fig 11 en el primer renglón (Figs. 11 (a-d)) se muestra la evolución de los bordes, el segundo renglón (Figs. 11 (e-h)) imágenes con los valores de W y el último renglón (Figs. 11 (i-l)) corresponde al mapa de decisión calculado en las iteraciones 1, 2, 3 y 10. Cabe hacer notar que en cada iteración el mapa de decisión es mejorado hasta llegar al valor de PA =99.37 %.

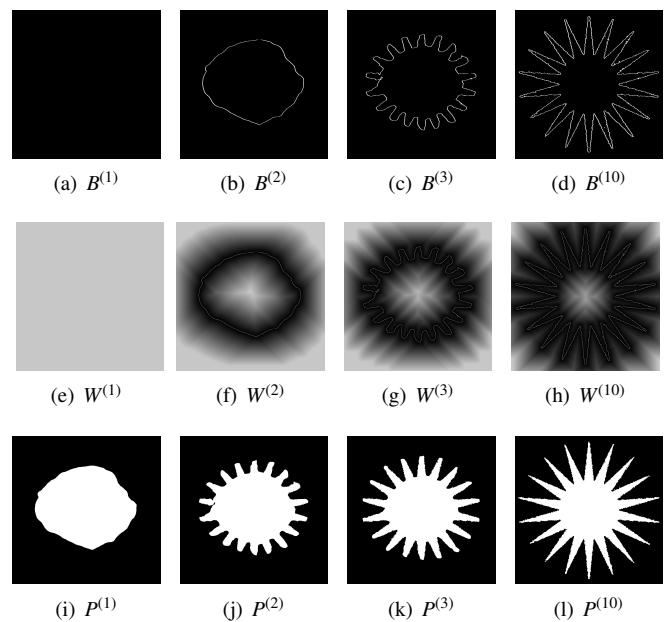


Figura 11: Evolución de los bordes B , tamaños de ventana W y del mapa de decisión P , para la imagen de lena con un mapa de decisión en forma de estrella.

Para el siguiente experimento el parámetro utilizado en el algoritmo CLI-S fue $w = 15$, el cual es el óptimo de acuerdo con la gráfica de la Figura 5. Para el caso del algoritmo CLI-VV se utilizaron como parámetros $w_{max} = 58$, $T = 5$ y $N_{iter} = 10$. Los parámetros se asumieron fijos para hacer notar el comportamiento de los algoritmos bajo diferentes condiciones de imágenes y mapas de decisión.

En la Figura 12 se muestran las diferencias entre los mapas de decisión dados (Fig. 8) y los mapas de decisión calculados con los algoritmos CLI-S y CLI-VV, respectivamente. Para estas imágenes blanco significa que hay diferencia y negro que no hay. Podemos notar de manera cualitativa que el algoritmo CLI-VV presenta menores diferencias que el algoritmo CLI-S.

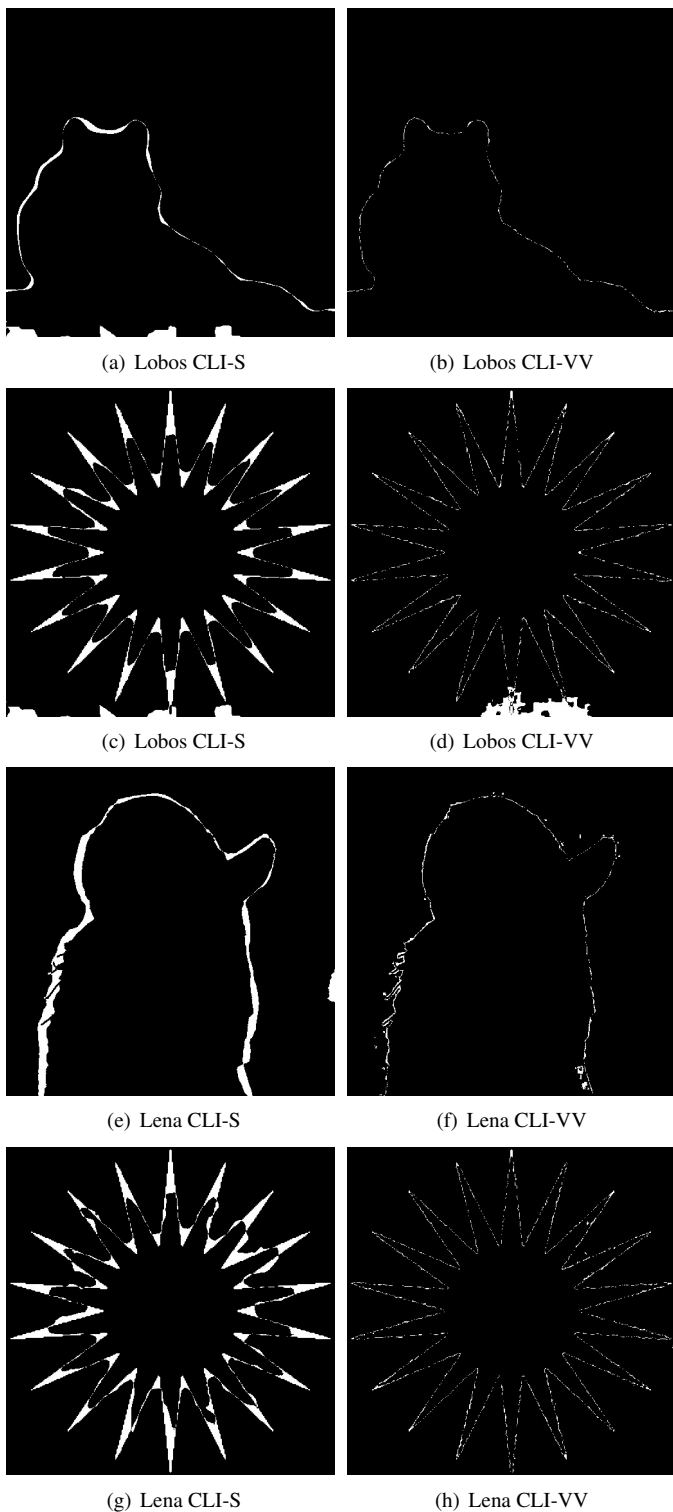


Figura 12: Diferencias calculadas entre el mapa de decisión dado y el calculado con los algoritmos CLI-S y CLI-VV

Los resultados del porcentaje de PA se muestran en la Tabla 1; en la primera columna se muestra el nombre de la imagen utilizada para calcular el par multi-foco sintético, en la segunda columna la Figura utilizada como mapa de decisión, en las columnas 3 y 4 los porcentajes de PA y el tiempo en segundos para el algoritmo CLI-S y finalmente en las columnas 5 y 6 el porcentaje PA y el tiempo para el algoritmo CLI-VV. En la Tabla 1 podemos notar que en todos los casos el Algoritmo CLI-VV

supera al Algoritmo CLI-S, aunque el tiempo de ejecución es considerablemente mayor.

Tabla 1: Precisión a Nivel Pixel (PA) obtenidos con los algoritmos CLI-S y FM-VV.

I_0	P_0	CLI-S		CLI-VV	
		PA	seg.	PA	seg.
Lobos	Fig. 8(a)	98.4	0.016	99.8	0.256
	Fig. 8(c)	93.4	0.016	98.1	0.255
Lena	Fig. 8(b)	96.4	0.016	99.3	0.244
	Fig. 8(c)	93.0	0.016	98.7	0.239

La Figura 13 muestra los promedios de tiempo de ejecución en segundos del Algoritmo CLI-VV requerido para fusionar pares de imágenes con diferentes tamaños $n_p = n_r \times n_c$. El Algoritmo CLI-VV se ejecutó 50 veces para cada par y se calculó el promedio de tiempo. Los tiempos promedio se graficaron en la Fig. 13, como puntos sobre la línea de mínimos cuadrados para resaltar el comportamiento lineal del Algoritmo CLI-VV.

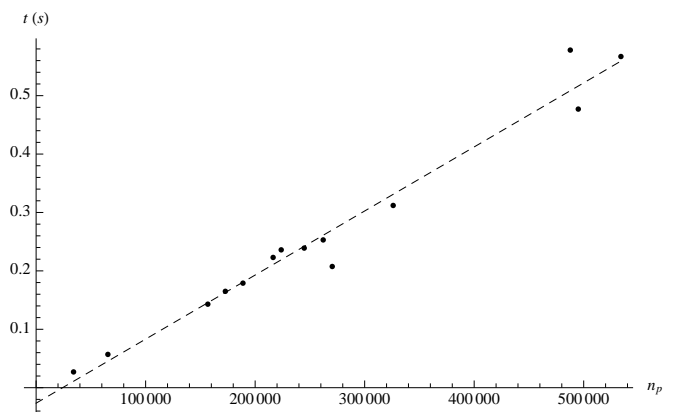


Figura 13: Gráfica del tiempo de ejecución t (s) del Algoritmo CLI-VV contra el número de píxeles n_p de las imágenes.

En la Tabla 2 se presenta el tiempo de ejecución para los Algoritmos de De and Chanda (De and Chanda, 2013), Bai et al. (Bai et al., 2015) y CLI-VV. En la Tabla se puede notar que el Algoritmo CLI-VV tiene tiempos de ejecución muy por debajo de los tiempos reportados en estos trabajos para un par de imágenes sintéticas de tamaño 512×512 .

Tabla 2: Comparación de tiempo de ejecución en propuestas similares.

Tiempo (s.)		
(De and Chanda, 2013)	(Bai et al., 2015)	CLI-VV
29	1.029	0.253

Para mostrar el desempeño del algoritmo CLI-VV, adicionalmente a las imágenes sintéticas de la Fig. 8, utilizamos algunas de las imágenes de la página PASCAL VOC para generar 17 pares de imágenes multi-foco. En la Fig. 14 columnas 1 y 2 se muestran 4 de los 17 pares generados y los bordes del mapa de decisión en blanco sobre la imagen fusionada se muestran en la tercer columna.

Para las 21 imágenes sintéticas calculamos el error entre el mapa de decisión dado y el calculado mediante los algoritmos CLI-S y CLI-VV, utilizando las métricas dadas por (17).

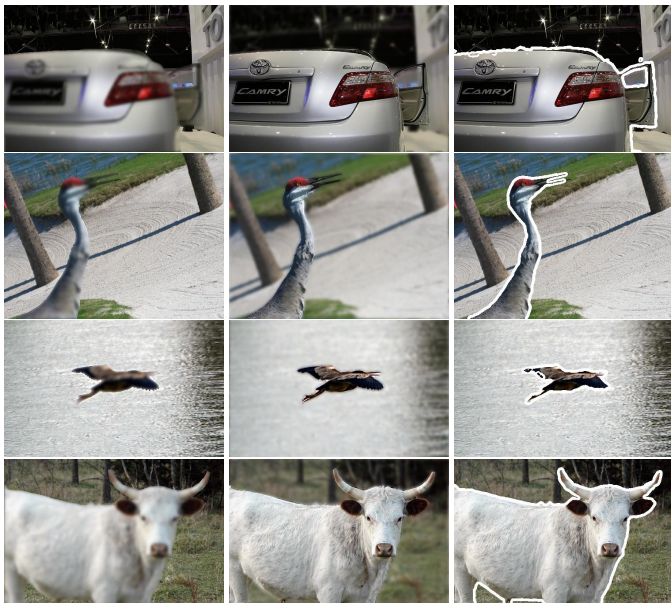


Figura 14: Ejemplos de la aplicación del Algoritmo CLI-VV a 20 imágenes de 512×512 píxeles utilizando los mismos parámetros

En la Tabla 3 se muestran los promedios de las métricas de segmentación para los algoritmos CLI-S y CLI-VV; note en esta tabla que en promedio tenemos una precisión a nivel píxel PA del 99.19 % con el algoritmo CLI-VV contra 95.7 % del Algoritmo CLI-S.

Tabla 3: Índices promedio calculados para 21 pares de imágenes multi-foco sintéticas utilizando los algoritmos CLI-S y CLI-VV

Algoritmo	PA	MPA	MIU	FWIU
CLI-S	95.70	93.91	88.73	91.94
CLI-VV	99.19	98.90	97.65	98.39

4.1.2. Pares de Imágenes Reales

En la Figura 15 se muestran los resultados de aplicar el Algoritmo CLI-VV a 4 pares de imágenes de un conjunto de 9 pares de imágenes multi-foco obtenidas de internet y de algunos artículos relacionados con el tema, todas ellas de tamaño diferente. En esta Figura se muestran tres columnas correspondientes a los pares de imágenes utilizadas así como los resultados obtenidos. En la primera y segunda columna se muestran los pares de imágenes multi-foco I_1 e I_2 y en la tercer columna se muestra los bordes del mapa de segmentación sobre la imagen fusionada. Los casos de la líneas 1 y 3 se seleccionaron como muestra, por ser pares muy utilizados en la literatura. Cabe resaltar los casos mostrados en los renglones 2 y 4, de dicha figura, en donde se puede notar que el algoritmo CLI-VV permite segmentar regiones pequeñas, contenidas dentro de regiones más grandes. Por ejemplo, en las figuras correspondientes a la escultura y el edificio mostradas en el segundo renglón, se logró segmentar adecuadamente las regiones que están entre los pies

de los personajes de la escultura. Dado que las imágenes utilizadas tienen diferentes tamaños, los parámetros utilizados fueron seleccionados hasta obtener la mejor imagen fusionada. Cabe mencionar que solamente el parámetro del tamaño de la ventana se fijó en un porcentaje dado por $w_{max} = 0.3 \max(nr, nc)$.

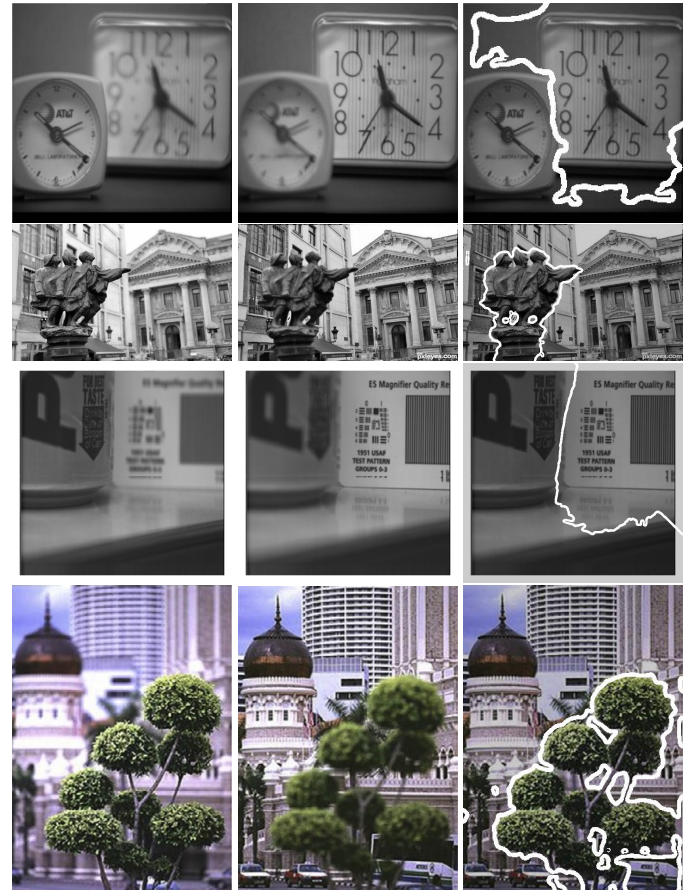


Figura 15: Resultado de aplicar el Algoritmo CLI-VV a 4 pares de imágenes con diferentes tamaños y utilizando diferentes parámetros.

Con el siguiente experimento queremos mostrar que el Algoritmo CLI-VV es capaz de obtener resultados razonables con un conjunto de 20 pares imágenes multi-foco utilizando los mismos parámetros. En la Figura 16 se muestran los resultados obtenidos en tres de las veinte imágenes, en la columna 1 y 2 se muestran los pares y en la columna 3 la imagen de los bordes del mapa de decisión sobre la imagen fusionada. Mostramos estos ejemplos debido a la calidad de la imágenes fusionadas y complejidad de par multifoco de las imágenes del primer y tercer renglón. Estas imágenes fueron tomadas de la red y algunas de ellas son utilizadas por (Nejati et al., 2015). Los 20 pares de imágenes son de tamaño 512×512 y los parámetros para el Algoritmo CLI-VV fueron $T = 20$, $w_{max} = 52$ y $N_{iter} = 10$. Es posible observar en estas imágenes que el mapa de decisión tiene la suficiente coherencia espacial, que se logra una segmentación adecuada de las partes nítidas, la cual coincide con los bordes de los objetos y que todos los objetos de la imagen fusionada lucen nítidos (columna 3).

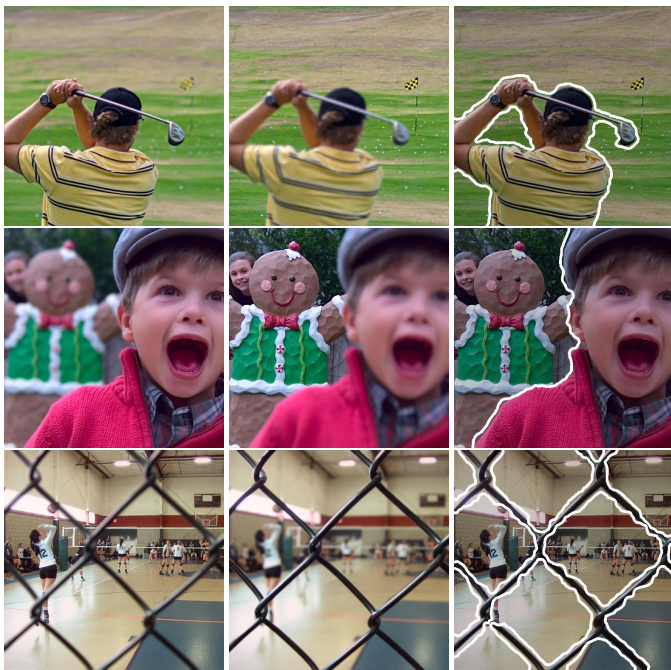


Figura 16: Ejemplos de la aplicación del Algoritmo CLI-VV a 3 pares de un conjunto de 20 pares de imágenes de 512×512 píxeles utilizando los mismos parámetros

En la Tabla 4 se presentan los promedios de las métricas dadas por (17) para el mapa de decisión calculado para los 29 pares de imágenes reales utilizando el algoritmo CLI-VV. En esta tabla el primer renglón corresponde a la información para los 9 pares de imágenes con diferentes tamaños (con diferentes parámetros) y el segundo renglón a 20 pares de imágenes con el mismo tamaño (donde se utilizaron los mismos parámetros). Note que tenemos una precisión a nivel píxel promedio del 95.61 %.

Tabla 4: Métrica promedio calculados para los 29 pares de imágenes multi-foco reales

Parámetros	Total	PA	MPA	MIU	FWIU
Diferentes	9	92.77	92.42	85.86	87.06
Iguales	20	96.88	97.01	92.93	94.20
Promedio	29	95.61	95.58	90.74	91.99

Con el conjunto de 20 pares de imágenes multi-foco reales, se realizó la fusión utilizando el algoritmo CLI-VV con diferentes implementaciones del Laplaciano utilizando: Laplaciano Discretizado, Diferencias de Gaussinas (DoG) y segundas derivadas de Gaussinas $\nabla^2 G_\sigma$. Para DoG se utilizó $G_{\sigma_1} - G_{\sigma_2}$ con $\sigma_1 = 0$ y $\sigma_2 = 0.7$. En el caso de las segundas derivadas de Gaussiana se utilizó $\sigma = 1$. Los resultados se muestran en la Tabla 5 note que los resultados no son significativamente diferentes pero tanto DoG como $\nabla^2 G_\sigma$ son muy sensibles a la varianza de la Gaussiana σ . En la Tabla 5 se reporta el promedio de las métricas dadas por (17) y los tiempos requeridos para obtener la imagen fusionada con el algoritmo CLI-VV. Los valores de σ , σ_1 y σ_2 se calcularon de manera manual hasta lograr el máximo desempeño y los parámetros de CLI-VV utilizados fueron $T = 20$, $w_{max} = 52$ y $N_{iter} = 10$. Todos los parámetros

fueron los mismos para el conjunto de los 20 pares de imágenes.

Tabla 5: Promedio de métricas y tiempos para el algoritmo CLI-VV utilizando diferentes implementaciones del Laplaciano

Implementación del Laplaciano	Promedio Métricas	Tiempo Segundos
Discretizado	95.26	0.2093
DoG	95.25	0.2178
$\nabla^2 G_\sigma$	94.14	0.2604

4.2. Resultado de aplicar el algoritmo FM-VV a conjuntos con mas de tres imágenes multi-foco

La Figura 17 muestra los resultados de aplicar el Algoritmo FM-VV a tres imágenes multi-foco. La Figura 17(c) muestra el resultado de fusionar las imágenes mostradas en las Figs. 17(a) y 17(b) correspondientes a I_1 e I_2 , respectivamente y la Figura 17(e) la fusión de las imágenes de las Figs. 17(c) y 17(d) correspondientes a las imágenes $I_F^{(1)}$ e I_3 , respectivamente. En la Figura 17(f) se muestra el mapa de decisión resultante de aplicar el Algoritmo FM-VV (Alg 4). En esta figura las regiones marcadas en color blanco se deberán tomar de la imagen I_1 , las marcadas en gris deberán tomarse de la imagen I_2 y las marcadas con negro se deberán tomar de I_3 . Se podría pensar que el fondo de estas imágenes hubiese sido mal segmentado, sin embargo, es indistinto tomar la información de I_1 , I_2 o I_3 , dado que es una superficie sin textura que no se ve afectada por el cambio de foco. Los parámetros utilizados por el Algoritmo FM-VV para este ejemplo fueron $w_{max} = 25$ y $T = 13$.

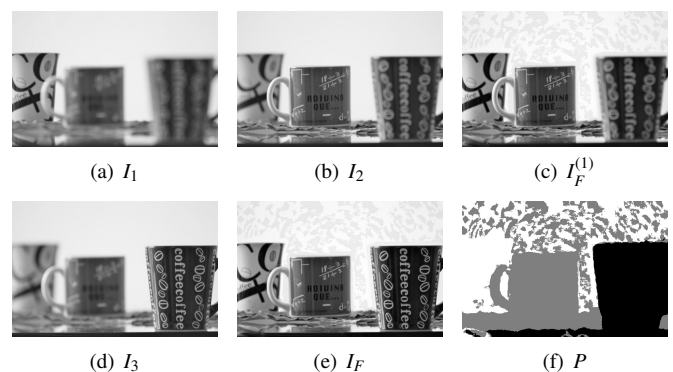


Figura 17: Ejemplo del procedimiento para fusionar tres imágenes multi-foco correspondientes a unas tazas.

Los ejemplos que se muestran a continuación corresponden a 4 conjuntos con 3 imágenes multi-foco, obtenidas de la red. Las escenas corresponden a un buzo en la playa, unos lobos marinos, un teclado de computadora y unas baterías en fila. Se han fijado los valores de los parámetros en $T = 30$ y $w_{max} = 104$ (20 % de las dimensiones de las imágenes). En la Figura 18 tenemos en cada renglón el conjunto de las tres imágenes multi-foco y los bordes del mapa de decisión en color blanco sobre la imagen fusionada calculada con el algoritmo FM-VV. Puede observarse en dicha figura que los bordes del mapa de decisión obtenido, se ajustan a los bordes reales de las imágenes y que además se logra la coherencia espacial en zonas sin bordes. En la Tabla 6 se muestran de manera cuantitativa las 4 métricas

para el mapa de decisión calculado con FM-VV y el mapa de decisión de referencia realizado a mano (utilizando el programa GIMP). El promedio de PA fue de 92.19 % para las cuatro imágenes.

Tabla 6: Métricas calculadas para los 4 conjuntos de 3 imágenes multi-foco utilizando los mismos parámetros.

Desc.	PA	MPA	MIU	FWIU
Buzo	92.63	93.93	82.28	87.26
Lobos Marinos	92.01	93.19	84.96	85.51
Teclado	97.81	97.55	95.40	95.71
Baterías	90.33	89.88	81.86	82.58
Promedio	93.19	93.64	86.12	87.76

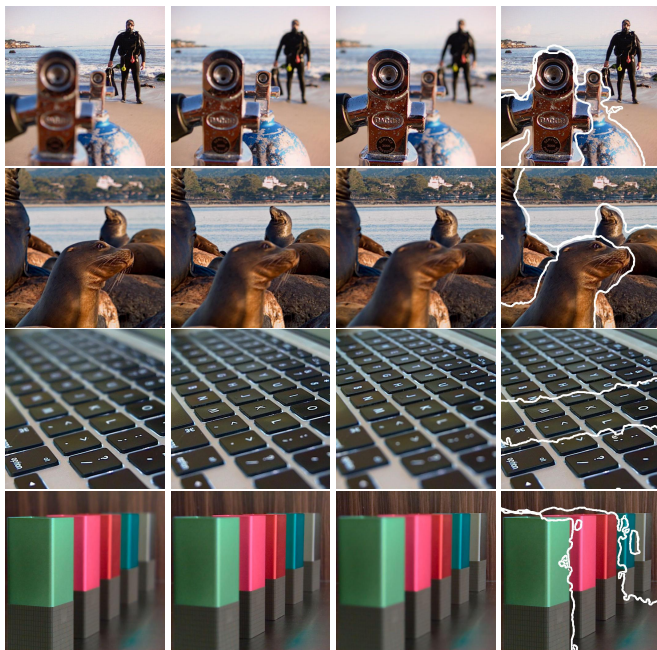


Figura 18: Ejemplos de la aplicación del Algoritmo FM-VV a 4 imágenes utilizando los mismos parámetros

5. Conclusiones

En este trabajo presentamos los Algoritmos CLI-VV y FM-VV para la fusión de imágenes multi-foco. El Algoritmo CLI-VV fue desarrollado para mejorar el Algoritmo CLI-S publicado en (Calderon et al., 2016) y el Algoritmo FM-VV es la generalización del CLI-VV para más de un par de imágenes.

Modificamos al Algoritmo CLI-S con el fin de mejorar la definición en los bordes del mapa de segmentación manteniendo coherencia espacial en las regiones donde se requiere. El Algoritmo CLI-VV obtuvo una precisión PA del 99.19 % en las 21 imágenes sintéticas mostradas en la sección de resultados superando al algoritmo CLI-S en todos los casos. Mostramos resultados del algoritmo CLI-VV con 20 pares de imágenes reales donde los parámetros se mantuvieron constantes para mostrar que puede desempeñarse satisfactoriamente independientemente sin tener que cambiar los parámetros para cada par de imágenes, en este caso el promedio de PA fue de 96.88 %.

Las imágenes integrales se utilizaron para detectar el total de bordes y la suma de las diferencias en una ventana dentro del Algoritmo CLI-VV, logrando que el tiempo de evaluación en la ventana sea constante, independientemente del tamaño de ésta. Haber utilizados las imágenes integrales no afecta los desempeños de los algoritmos CLI-VV y FM-VV, pero mejora sustancialmente el tiempo de cómputo. Debido a la naturaleza iterativa del Algoritmo CLI-VV su tiempo de ejecución es mayor que el tiempo del Algoritmo CLI-S, sin embargo, la calidad de los resultados del algoritmo CLI-VV es superior con tiempos de ejecución por debajo de algunos reportados en el estado del arte. Adicionalmente, el único parámetro usado por CLI-VV es el tamaño máximo de ventana y sus resultados no son sensibles a él para un tamaño dado de la imagen.

En virtud de la rapidez de los Algoritmos CLI-VV y FM-VV estos serán utilizados para la creación de fotografías nítidas a partir de un conjunto de imágenes.

Aunque consideramos que los resultados obtenidos son suficientemente buenos, existen mejoras que se pueden aplicar al algoritmo. Por ejemplo, probar con otras técnicas para medir el nivel de enfoque, aplicar el proceso en un esquema de pirámides con el fin de tener representaciones multi-escala y aplicar programación paralela para mejorar su eficiencia en tiempo.

Referencias

- Aslantas, V., Kurban, R., 2010. Fusion of multi-focus images using differential evolution algorithm. *Expert Systems with Applications* 37 (12), 8861 – 8870.
DOI: <http://dx.doi.org/10.1016/j.eswa.2010.06.011>
- Aslantas, V., Toprak, A. N., 2014. A pixel based multi-focus image fusion method. *Optics Communications* 332, 350 – 358.
DOI: <http://dx.doi.org/10.1016/j.optcom.2014.07.044>
- Aslantas, V., Toprak, A. N., 2017. Multi-focus image fusion based on optimal defocus estimation. *Computers and Electrical Engineering*.
DOI: <http://dx.doi.org/10.1016/j.compeleceng.2017.02.003>
- Assirati, L., Silva, N. R., Berton, L., Lopes, A. A., Bruno, O. M., 2014. Performing edge detection by difference of gaussians using q-gaussian kernels. *Journal of Physics: Conference Series* 490 (1), 012020.
- Bai, X., Zhang, Y., Zhou, F., Xue, B., 2015. Quadtree-based multi-focus image fusion using a weighted focus-measure. *Information Fusion* 22, 105 – 118.
DOI: <http://dx.doi.org/10.1016/j.inffus.2014.05.003>
- Calderon, F., Garnica, A., 2014. Multi focus image fusion based on linear combination of images. *IEEE*, pp. 1–7.
DOI: 10.1109/ROPEC.2014.7036340
- Calderon, F., Garnica-Carrillo, A., Flores, J. J., 2016. Fusión de imágenes multi foco basado en la combinación lineal de imágenes utilizando imágenes incrementales. *Revista Iberoamericana de Automática e Informática Industrial (RIAI)* 13 (4), 450 – 461.
DOI: <http://dx.doi.org/10.1016/j.riai.2016.07.002>
- Cao, L., Jin, L., Tao, H., Li, G., Zhuang, Z., Zhang, Y., Feb 2015. Multi-focus image fusion based on spatial frequency in discrete cosine transform domain. *Signal Processing Letters, IEEE* 22 (2), 220–224.
DOI: 10.1109/LSP.2014.2354534
- Chai, Y., Li, H., Li, Z., 2011. Multifocus image fusion scheme using focused region detection and multiresolution. *Optics Communications* 284 (19), 4376 – 4389.
DOI: <http://dx.doi.org/10.1016/j.optcom.2011.05.046>
- De, I., Chanda, B., 2013. Multi-focus image fusion using a morphology-based focus measure in a quad-tree structure. *Information Fusion* 14 (2), 136 – 146.
DOI: <http://dx.doi.org/10.1016/j.inffus.2012.01.007>
- Duan, J., Meng, G., Xiang, S., Pan, C., 2014. Multifocus image fusion via focus segmentation and region reconstruction. *Neurocomputing* 140, 193 – 209.
DOI: <http://dx.doi.org/10.1016/j.neucom.2014.03.023>
- Eskicioglu, A., Fisher, P., Dec 1995. Image quality measures and their performance. *Communications, IEEE Transactions on* 43 (12), 2959–2965.
DOI: 10.1109/26.477498

- Kong, W., Lei, Y., 2017. Multi-focus image fusion using biochemical ion exchange model. *Applied Soft Computing* 51, 314 – 327.
DOI: <http://dx.doi.org/10.1016/j.asoc.2016.11.033>
- Kuthirummal, S., Nagahara, H., Zhou, C., Nayar, S., Jan 2011. Flexible depth of field photography. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33 (1), 58–71.
DOI: 10.1109/TPAMI.2010.66
- Lewis, J. J., O’Callaghan, R. J., Nikolov, S. G., Bull, D. R., Canagarajah, N., 2007. Pixel- and region-based image fusion with complex wavelets. *Information Fusion* 8 (2), 119 – 130, special Issue on Image Fusion: Advances in the State of the Art.
DOI: <http://dx.doi.org/10.1016/j.inffus.2005.09.006>
- Li, H., Chai, Y., Li, Z., 2013a. Multi-focus image fusion based on nonsubsampled contourlet transform and focused regions detection. *Optik - International Journal for Light and Electron Optics* 124 (1), 40 – 51.
DOI: <http://dx.doi.org/10.1016/j.ijleo.2011.11.088>
- Li, H., Chai, Y., Li, Z., 2013b. A new fusion scheme for multifocus images based on focused pixels detection. *Machine vision and applications* 24 (6), 1167–1181.
- Li, H., Manjunath, B., Mitra, S., 1995. Multisensor image fusion using the wavelet transform. *Graphical Models and Image Processing* 57 (3), 235 – 245.
DOI: <http://dx.doi.org/10.1006/gmip.1995.1022>
- Li, S., Kang, X., Fang, L., Hu, J., Yin, H., 2017. Pixel-level image fusion: A survey of the state of the art. *Information Fusion* 33, 100 – 112.
DOI: <http://dx.doi.org/10.1016/j.inffus.2016.05.004>
- Li, S., Kwok, J. T., Wang, Y., 2001. Combination of images with diverse focuses using the spatial frequency. *Information Fusion* 2 (3), 169 – 176.
DOI: [http://dx.doi.org/10.1016/S1566-2535\(01\)00038-0](http://dx.doi.org/10.1016/S1566-2535(01)00038-0)
- Li, S., Kwok, J. T., Wang, Y., 2002. Multifocus image fusion using artificial neural networks. *Pattern Recognition Letters* 23 (8), 985 – 997.
DOI: [http://dx.doi.org/10.1016/S0167-8655\(02\)00029-6](http://dx.doi.org/10.1016/S0167-8655(02)00029-6)
- Li, S., Yang, B., 2008a. Multifocus image fusion by combining curvelet and wavelet transform. *Pattern Recognition Letters* 29 (9), 1295–1301.
- Li, S., Yang, B., 2008b. Multifocus image fusion using region segmentation and spatial frequency. *Image and Vision Computing* 26 (7), 971 – 979.
DOI: <http://dx.doi.org/10.1016/j.imavis.2007.10.012>
- Li, X., He, M., Roux, M., August 2010. Multifocus image fusion based on redundant wavelet transform. *Image Processing, IET* 4 (4), 283–293.
DOI: 10.1049/iet-ivr.2008.0259
- Liu, Y., Chen, X., Peng, H., Wang, Z., 2017a. Multi-focus image fusion with a deep convolutional neural network. *Information Fusion* 36, 191 – 207.
DOI: <http://dx.doi.org/10.1016/j.inffus.2016.12.001>
- Liu, Z., Chai, Y., Yin, H., Zhou, J., Zhu, Z., 2017b. A novel multi-focus image fusion approach based on image decomposition. *Information Fusion* 35, 102 – 116.
DOI: <http://dx.doi.org/10.1016/j.inffus.2016.09.007>
- Long, J., Shelhamer, E., Darrell, T., 2014. Fully convolutional networks for semantic segmentation. *CoRR abs/1411.4038*.
- Luo, X., Zhang, J., Dai, Q., 2012. A regional image fusion based on similarity characteristics. *Signal Processing* 92 (5), 1268 – 1280.
DOI: <http://dx.doi.org/10.1016/j.sigpro.2011.11.021>
- Ma, Y., Zhan, K., Wang, Z., service), S. O., 2011. Applications of pulse-coupled neural networks.
- Malviya, A., Bhirud, S., Dec 2009. Wavelet based multi-focus image fusion. In: *Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*. pp. 1–6.
DOI: 10.1109/ICM2CS.2009.5397990
- Nejati, M., Samavi, S., Shirani, S., 2015. Multi-focus image fusion using dictionary-based sparse representation. *Information Fusion* 25, 72 – 84.
DOI: <http://dx.doi.org/10.1016/j.inffus.2014.10.004>
- Orozco, R. I., 2013. Fusión de imágenes multifoco por medio de filtrado de regiones de alta y baja frecuencia. Master’s thesis, División de Estudios de Postgrado. Facultad de Ingeniería Eléctrica. UMSNH, Morelia Michoacan Mexico.
- Pagidimarri, M., Babu, K. A., 2011. An all approach for multi-focus image fusion using neural network. *Artificial Intelligent Systems and Machine Learning* 3 (12), 732–739.
- Pajares, G., de la Cruz, J. M., 2004. A wavelet-based image fusion tutorial. *Pattern Recognition* 37 (9), 1855 – 1872.
- Piella, G., 2003. A general framework for multiresolution image fusion: from pixels to regions. *Information Fusion* 4 (4), 259 – 280.
DOI: [http://dx.doi.org/10.1016/S1566-2535\(03\)00046-0](http://dx.doi.org/10.1016/S1566-2535(03)00046-0)
- Pramanik, S., Prusty, S., Bhattacharjee, D., Bhunre, P. K., 2013. A region-to-pixel based multi-sensor image fusion. *Procedia Technology* 10, 654 – 662.
DOI: <http://dx.doi.org/10.1016/j.protcy.2013.12.407>
- Qu, X., Hou, Y., Lam, F., Guo, D., Zhong, J., Chen, Z., 2014. Magnetic resonance image reconstruction from undersampled measurements using a patch-based nonlocal operator. *Medical Image Analysis* 18 (6), 843 – 856, sparse Methods for Signal Reconstruction and Medical Image Analysis.
- Riaz, M., Park, S., Ahmad, M., Rasheed, W., Park, J., 2008. Generalized laplacian as focus measure. In: *Bubak, M., van Albada, G., Dongarra, J., Sloot, P. (Eds.), Computational Science ICCS 2008. Vol. 5101 of Lecture Notes in Computer Science. Springer Berlin Heidelberg*, pp. 1013–1021.
- Rivera, M., Ocegueda, O., Marroquin, J., Dec 2007. Entropy-controlled quadratic markov measure field models for efficient image segmentation. *Image Processing, IEEE Transactions on* 16 (12), 3047–3057.
DOI: 10.1109/TIP.2007.909384
- Sezan, M., Pavlovic, G., Tekalp, A., Erdem, A., Apr 1991. On modeling the focus blur in image restoration. In: *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*. pp. 2485–2488 vol.4.
DOI: 10.1109/ICASSP.1991.150905
- Shah, P., Merchant, S. N., Desai, U. B., 2013. Multifocus and multispectral image fusion based on pixel significance using multiresolution decomposition. *Signal, Image and Video Processing* 7 (1), 95–109.
DOI: 10.1007/s11760-011-0219-7
- Shi, W., Zhu, C., Tian, Y., Nichol, J., 2005. Wavelet-based image fusion and quality assessment. *International Journal of Applied Earth Observation and Geoinformation* 6 (3-4), 241 – 251.
- Tian, J., Chen, L., Sept 2010. Multi-focus image fusion using wavelet-domain statistics. In: *Image Processing (ICIP), 2010 17th IEEE International Conference on*. pp. 1205–1208.
DOI: 10.1109/ICIP.2010.5651791
- Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. pp. I-511–I-518 vol.1.
DOI: 10.1109/CVPR.2001.990517
- Yang, Y., 2011. A novel {DWT} based multi-focus image fusion method. *Procedia Engineering* 24 (0), 177 – 181, international Conference on Advances in Engineering 2011.
- Yang, Y., Huang, S., Gao, J., Qian, Z., 2014. Multi-focus image fusion using an effective discrete wavelet transform based algorithm. *Measurement Science Review* 14 (2), 102 – 108.
- Yang, Y., Tong, S., Huang, S., Lin, P., 2015. Multifocus image fusion based on nscf and focused area detection. *IEEE Sensors Journal* 15 (5), 2824–2838.
- Zhang, B., Lu, X., Pei, H., Liu, H., Zhao, Y., Zhou, W., 2016a. Multi-focus image fusion algorithm based on focused region extraction. *Neurocomputing* 174, 733 – 748.
DOI: <http://dx.doi.org/10.1016/j.neucom.2015.09.092>
- Zhang, Q., long Guo, B., 2009. Multifocus image fusion using the nonsubsampled contourlet transform. *Signal Processing* 89 (7), 1334 – 1346.
DOI: <http://dx.doi.org/10.1016/j.sigpro.2009.01.012>
- Zhang, Y., Chen, L., Zhao, Z., Jia, J., 2016b. Multi-focus image fusion based on cartoon-texture image decomposition. *Optik - International Journal for Light and Electron Optics* 127 (3), 1291 – 1296.
DOI: <http://dx.doi.org/10.1016/j.ijleo.2015.10.098>
- Zhang, Z., Blum, R., Aug 1999. A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application. *Proceedings of the IEEE* 87 (8), 1315–1326.
DOI: 10.1109/5.775414
- Zhao, H., Li, Q., Feng, H., 2008. Multi-focus color image fusion in the {HSI} space using the sum-modified-laplacian and a coarse edge map. *Image and Vision Computing* 26 (9), 1285 – 1295.
DOI: <http://dx.doi.org/10.1016/j.imavis.2008.03.007>
- Zhou, L., Ji, G., Shi, C., Feng, C., Nian, R., 2006. A Multi-focus Image Fusion Method Based on Image Information Features and the Artificial Neural Networks. Vol. 344. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 747–752.
- Zhou, Z., Li, S., Wang, B., 2014. Multi-scale weighted gradient-based fusion for multi-focus images. *Information Fusion* 20 (0), 60 – 72.