

DISEÑO DE UNA ARQUITECTURA EN HARDWARE PARA OBTENER EL CANAL OSCURO EN IMÁGENES CON HAZE

Design of a Hardware Architecture to Obtain the Dark Channel on Hazy Images

Miguel Ángel Moncada Malagón^{1*}, José Eduardo Gaspar Badillo¹,
Juan Manuel Ramos Arreguín¹, Jesús Carlos Pedraza Ortega¹,
Marco Antonio Aceves Fernández¹

¹Universidad Autónoma de Querétaro

*Autor de correspondencia:
moncada.mma1@gmail.com

RESUMEN

La presencia de grandes cantidades de partículas acumuladas en la atmósfera produce baja visibilidad en las imágenes. A este efecto se le conoce como hazing. Uno de los métodos más eficientes encontrados en la literatura que utiliza una imagen única como entrada, es decir, que no requiere información adicional de la imagen, es el algoritmo del Dark Channel Prior (DCP), el cual está basado en obtener el canal oscuro o dark channel para remover este efecto. En este trabajo se presenta la metodología y arquitectura en VHDL para obtener solamente del canal oscuro en una tarjeta Nexys 4 de Xilinx que posteriormente puede ser usado para implementar el algoritmo de dehazing DCP. El sistema embebido resultante de la metodología permite almacenar las imágenes tanto en memoria Flash como en bloques de memoria RAM. Las imágenes resultantes se muestran mediante el puerto VGA del FPGA hacia un monitor con una resolución de 640x480 pixeles con 12 bits de color.

Palabras clave: dehazing, canal oscuro, FPGA, VHDL, visión por computadora, procesamiento de imágenes

ABSTRACT

The presence of large amounts of particles accumulated in the atmosphere results in low visibility in images. This effect is known as hazing. One of the most efficient methods found in the literature, which uses a single image as an input, that is, that does not require additional image information, is the Dark Channel Prior (DCP) algorithm, which is based on obtaining the dark channel to remove this effect. This paper presents the methodology and architecture in VHDL for obtaining only the dark channel on a Nexys 4 FPGA by Xilinx which can later be used to implement the DCP dehazing algorithm. The embedded system resulting from the methodology allows the images to be stored

both in Flash memory and in blocks of RAM, the resulting images are displayed through the FPGA VGA port to a monitor with a resolution of 640 x 480 pixels with 12 color bits.

Keywords: dehazing, dark channel prior, FPGA, VHDL, computer vision, image processing

INTRODUCCIÓN

En la actualidad, el procesamiento digital de imágenes es un área de interés para los investigadores debido a su inmenso campo de acción, el cual va desde la mejora de la calidad de una imagen hasta el procesamiento del más alto nivel como lo es la clasificación de objetos, y a las situaciones a las que se enfrentan. En ocasiones la presencia de grandes cantidades de partículas acumuladas en la atmósfera produce baja visibilidad en las imágenes. A este efecto se le conoce como hazing. Esta baja visibilidad en las imágenes afecta la precisión de las técnicas de visión por computadora como la detección de objetos, el seguimiento de rostros, el reconocimiento de matrículas, entre otras, así como en tareas relacionadas con sistemas de vigilancia y sistemas avanzados de asistencia al conductor (ADAS, por sus siglas en inglés). Es por esto que surge la necesidad de métodos para resolver este problema, que se desarrollan en años recientes y son nombrados como algoritmos de dehazing.

En la literatura, diversos métodos han sido propuestos para tratar este problema. En 2008 Tan observa que una imagen libre de hazing debería tener un contraste más alto por lo que remueve el haze maximizando el contraste local de la imagen. Los resultados son visualmente buenos, pero pueden llegar a ser no válidos físicamente y el algoritmo generalmente produce imágenes sobresaturadas [1]. En 2011 He propone un algoritmo de dehazing basado en el Dark Channel Prior (DCP) para estimar el mapa de transmisión de la imagen y obtiene resultados satisfactorios (véase Figura 1). Su principal desventaja



es que el proceso es lento y no puede ser usado en sistemas de tiempo real, además sus resultados sufren de halos alrededor de bordes [2]. Más adelante, para corregir estos halos en 2012, He lleva a cabo una variación del DCP donde toma en cuenta que los dispositivos de captura realizan ajustes automáticos como el balance de blancos por lo que propuso solucionar estos problemas al aplicar una corrección de balance de blancos y al descomponer la imagen en dos componentes: la luz reflejada en la imagen y la luz ambiental [3].

Recientemente se han reportado ampliamente trabajos relacionados con el uso del Dark Channel Prior. En 2016 Zhang & Zao proponen un método de dehazing en tiempo real, basado en el canal oscuro. Se emplea un método aproximado para estimar la luz atmosférica y el mapa de transmisión. Para aplicaciones de sistemas embebidos, propusieron una arquitectura de hardware en una FPGA Stratix con una velocidad de pro-

cesamiento de 116 MHz. Los resultados de la simulación indicaron que el hardware es altamente eficiente, ya que se obtienen buenos resultados de recuperación de imágenes y se satisfacen los requisitos en tiempo real inclusive para imágenes de gran tamaño [4]. En 2018 Salazar propone un algoritmo de dehazing basado en el DCP, donde su principal contribución se relaciona con el bajo tiempo de procesamiento obtenido mediante la combinación de morfología matemática, filtrado gaussiano y el DCP que son algoritmos de baja complejidad informática [5].

Como se puede observar el algoritmo de DCP es de interés para la comunidad científica, habiéndose desarrollado variantes del mismo, sin embargo, se habla poco de su implementación en sistemas embebidos para procesamiento en tiempo real. Este trabajo muestra una arquitectura en un FPGA para calcular el canal oscuro que es esencial para el algoritmo de DCP.

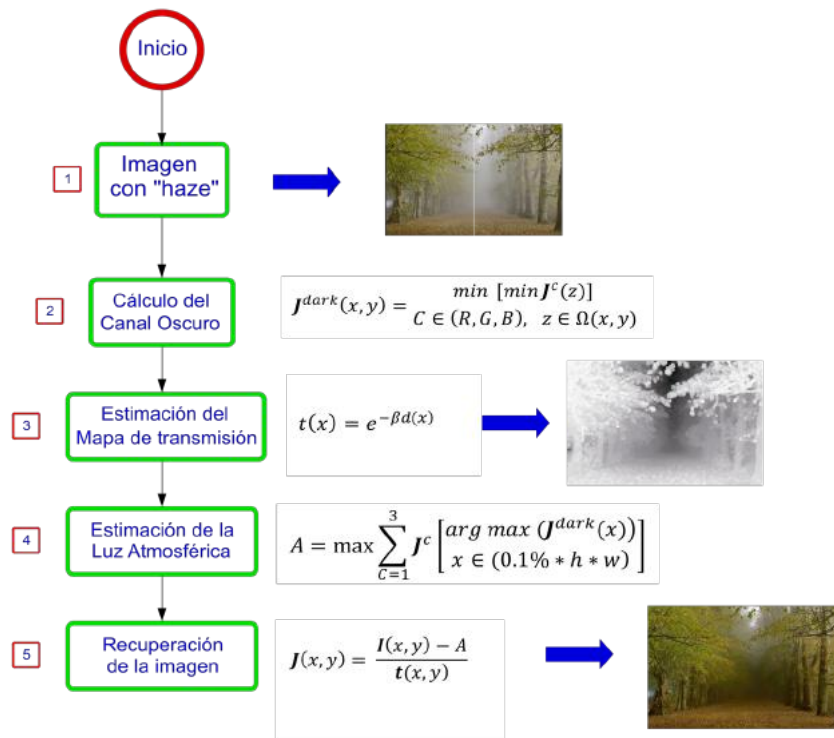


Figura 1. Algoritmo de dehazing con el DCP [2].

METODOLOGÍA

La metodología consiste en realizar una arquitectura estructural en VHDL. Los principales componentes de esta arquitectura son la adquisición de las imágenes entre el PC y el FPGA vía serial, el almacenamiento de las imágenes en memoria RAM y memoria FLASH, y el controlador de video VGA para mostrar las imágenes directamente del FPGA.

Arquitectura en el FPGA

El procesar imágenes en un FPGA requiere de un modelo generalizado que interactúe entre la PC, el FPGA y un monitor VGA para visualizar las imágenes, por lo cual se diseña el modelo de la Figura 2.

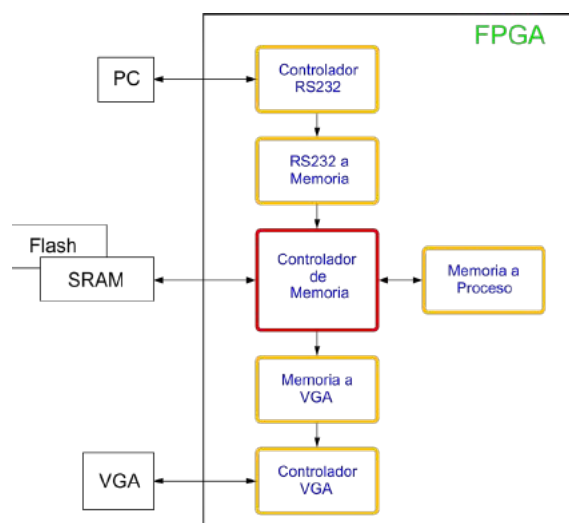


Figura 2. Diagrama a bloques del modelo generalizado para procesamiento de las imágenes en FPGA.

Este modelo se implementa en la Nexys 4 y se describe de la siguiente manera: sistema modular que adquiere las imágenes RGB mediante el puerto RS232, donde un controlador de memoria se encarga del almacenamiento de las imágenes en memoria Flash que, a su vez, se comunica con la memoria RAM para almacenar valores

temporales generados durante los procesos y, finalmente, un controlador de VGA permite mostrar las imágenes en un monitor.

Almacenamiento en el FPGA

El almacenamiento de las imágenes se lleva a cabo tanto en la memoria Flash como en bloques de memoria RAM, debido a que se almacenan las imágenes recibidas del PC en la memoria Flash. De esta forma no es necesario transmitir las imágenes en cada nuevo experimento. Mientras que los bloques de memoria RAM se utilizan para almacenar los cálculos y operaciones que se realizaron durante los procesos que se aplican a las imágenes. La arquitectura para el almacenamiento se muestra a continuación en la Figura 3.

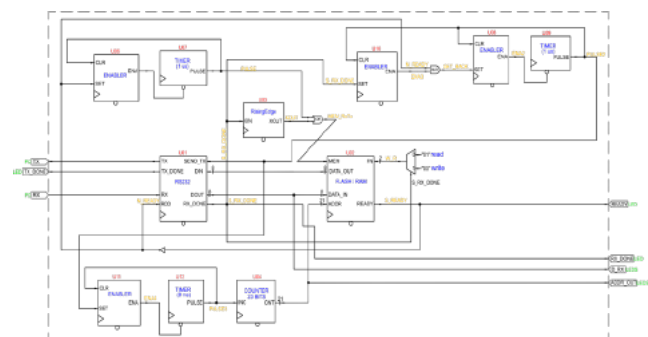


Figura 3. Diagrama de bloques de la arquitectura en hardware para el almacenamiento en memoria RAM y Flash.

Los datos se transmiten vía serial mediante una interfaz creada en Visual Studio 2017, que permite transferir los datos de los canales RGB de cada uno de los pixeles a una velocidad de 115,200 baudios. El puerto VGA de la Nexys 4 DDR es de 12 bit de color por lo que cada pixel de la imagen se envía como dos arreglos de la forma: [RRRRGGGG] y [BB-BBXXXX], donde en un arreglo se envían los canales rojo y verde y en el otro arreglo el canal azul.



Conexión VGA entre monitor y FPGA

La tarjeta Nexys 4 tiene un conector VGA de 12 bits de color, sin embargo, se desea tener una mayor gama de colores para mostrar las imágenes con una mayor fidelidad, para lo cual se diseña una tarjeta con un conector VGA de 24 bits. El diagrama electrónico se muestra en la Figura 4. Los diagramas de conexión entre el puerto VGA y el FPGA se pueden consultar en el documento Nexys4 DDR™ FPGA Board Reference Manual en la sección VGA Port [7].

La cantidad de colores que se puede tener con n cantidad de bits se describe en la Ecuación 1.

$$n^{\circ} \text{ de colores} = 2^n \text{ bits} \quad (1)$$

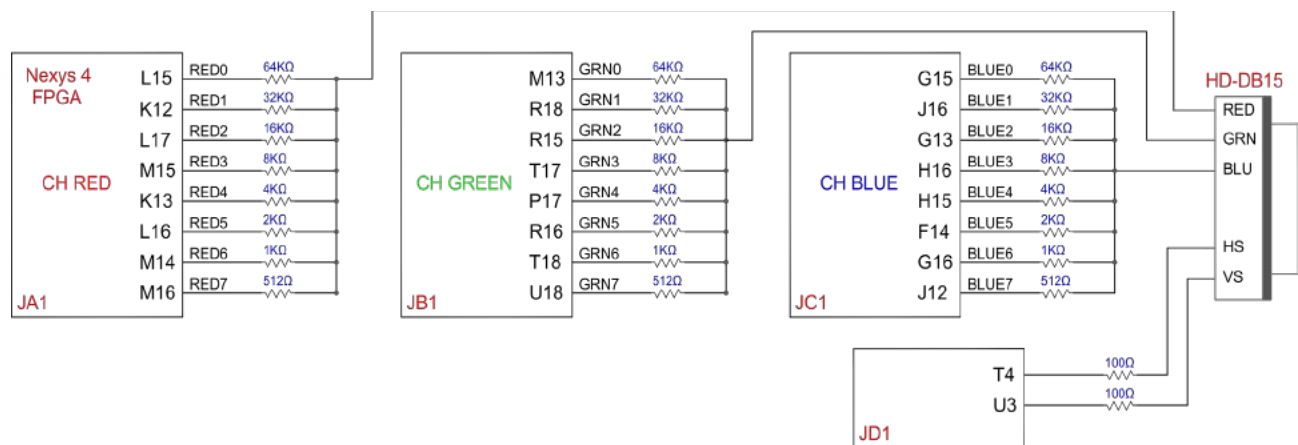


Figura 4. Diagrama de conexiones del puerto VGA externo para conexión con el FPGA.

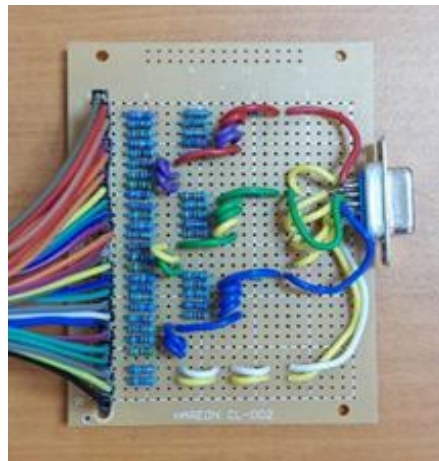


Figura 5. Tarjeta VGA 24 bits de color para conexión entre FPGA y monitor.

Con 24 bits se tiene una gama de 16.7 millones de colores, lo que permite mostrar las imágenes con una mayor calidad. El prototipo de esta tarjeta puede observarse en la Figura 5.

Generación de imágenes con haz

Las imágenes utilizadas pertenecen a la base de datos Kittí, la cual es una base de datos que contiene imágenes de escenas exteriores relacionadas con vehículos en carretera y pertenece a un benchmark para realizar tareas de visión por computadora para conducción autónoma [6].

Modelo de una imagen con haze

En visión por computadora el modelo que describe a una imagen con haze es el siguiente:

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (2)$$

Donde:

I = la intensidad observada (imagen con haze).

x = la posición del píxel (x, y).

$J(x)$ = la intensidad real de la escena (imagen sin haze).

A = la luz atmosférica global del aire.

$t(x)$ = llamado mapa de transmisión y describe la porción de luz que llega al sensor de la cámara sin dispersarse o ser absorbida.

Idealmente, es decir, si la atmósfera es homogénea el mapa de transmisión se describe como sigue:

$$t(x) = e^{-\beta d(x)} \quad (3)$$

Donde:

β = el coeficiente de dispersión atmosférica
 $d(x)$ = el mapa de profundidad.

A partir de este modelo se crean las imágenes con haze al tomar como imágenes de entrada la base de datos Kitti. En la Figura 6 se ilustra el proceso para generar las imágenes con haze artificialmente a partir de una imagen y su mapa de profundidad (lidar).

Canal Oscuro Prioritario (DCP)

El canal oscuro prioritario, propuesto por He en 2011, es una característica de imágenes adquiridas en escenas exteriores libres del efecto de hazing. Lo que se observa en estas imágenes es que, en la mayoría de los píxeles, al menos uno de sus canales RGB tiene una intensidad muy baja, cercana al 0. Al considerar lo anterior, para obtener el DCP, se aplica un parche de tamaño n con centro en un píxel, se evalúan los vecinos cercanos a este píxel que se encuentran en un parche de tamaño n con centro en el píxel en cuestión y presentan la misma propiedad, por lo cual,

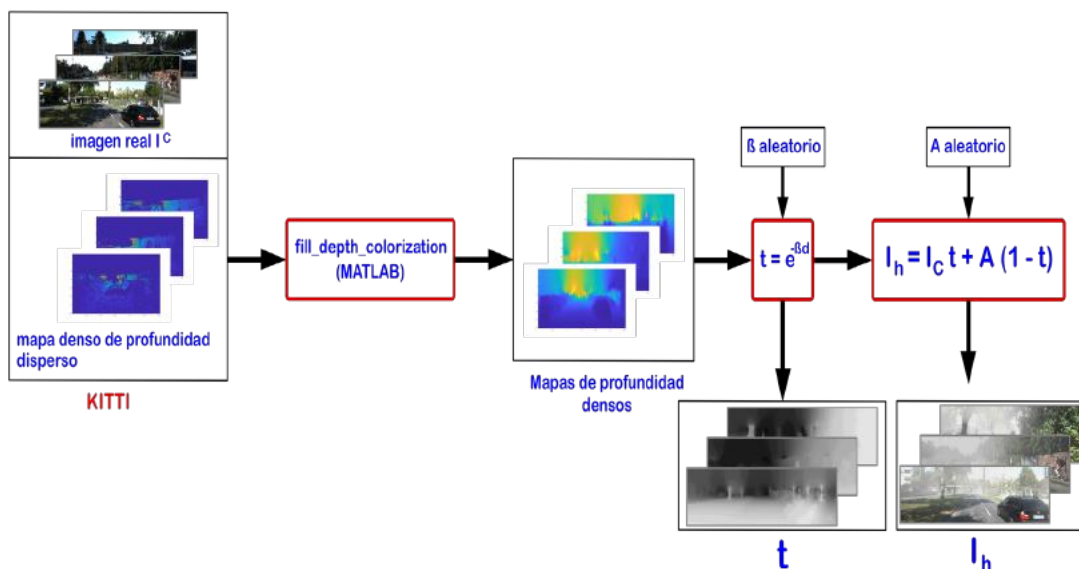


Figura 6. Diagrama para la generación de imágenes con haze usando la base de datos Kitti.



el canal oscuro prioritario se define como: el conjunto más pequeño de valores RGB dentro de un parche de tamaño n , lo que se expresa formalmente en la Ecuación 4 [2].

$$J^{dark}(x, y) = \min_{C \in (R, G, B)} [\min_{z \in \Omega(x, y)} J^c(z)] \quad (4)$$

Donde:

- J^c =un canal de color de J (R, G o B)
- $\Omega(x, y)$ =un parche local de tamaño n centrado en (x, y)
- z =el índice del pixel dentro del parche

Canal Oscuro (DC)

Si se aplica un parche de tamaño $n=1$, la Ecuación 4 se reduce a encontrar solamente el valor mínimo sobre los 3 canales RGB, con lo cual se obtiene el canal oscuro. El canal oscuro es la representación más básica del DCP. En este trabajo se obtiene el canal oscuro de

una imagen con haze y se pretende aplicar parches de diferentes tamaños como $n=3, 5, 7, 9$ en futuros trabajos.

Implementación del DC en el FPGA

La arquitectura de hardware para obtener el canal oscuro es una arquitectura estructural con un controlador de VGA, un bloque de RAM que se creó con el catalogo IP de vivado, un módulo que separa los 3 canales RGB y el módulo que se encarga de procesar los 3 canales RGB para obtener el canal oscuro (véase Figura 5), mientras que un generador de direcciones se encarga de sincronizar el controlador de VGA y el acceso al bloque de RAM.

RESULTADOS Y DISCUSIÓN

Por la arquitectura que está diseñada en general para procesar las imágenes, primeramente, se realizaron 2 pruebas a las imágenes de entrada: un filtro de binarización y un filtrado de escala de grises. En la Figura 8 se

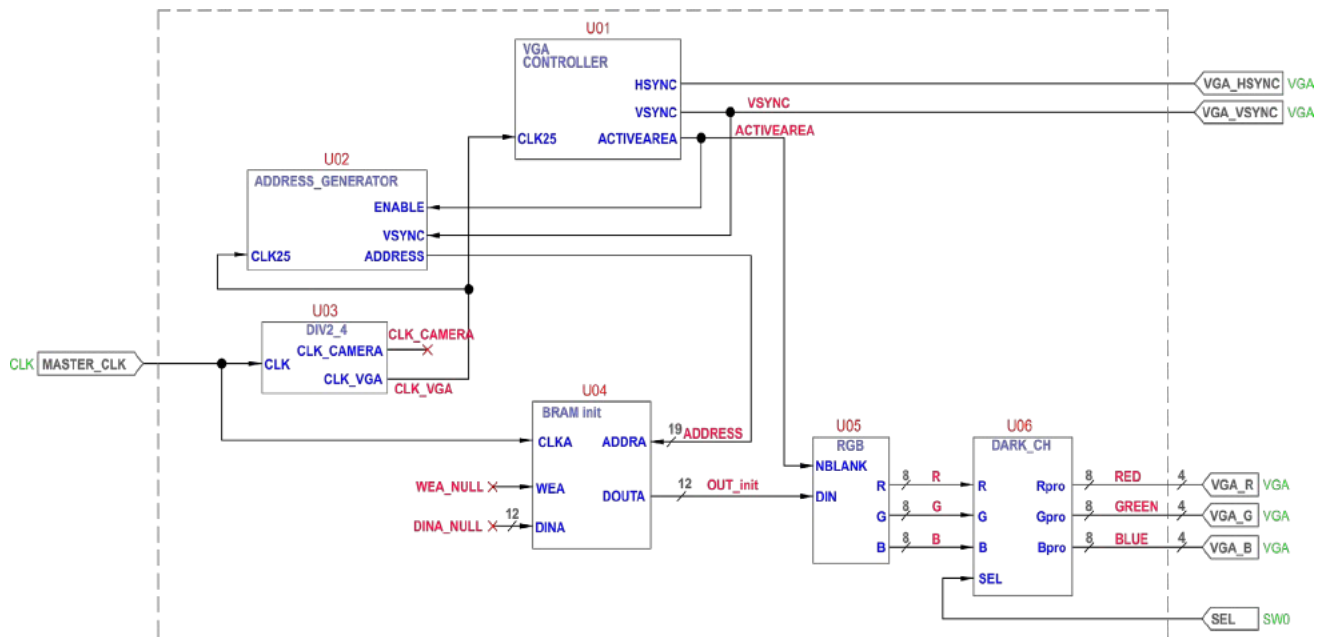


Figura 7. Diagrama de bloques de la arquitectura en hardware para obtener el canal oscuro.

observa en el monitor VGA la imagen de entrada almacenada en el FPGA. En la Figura 9 se observa la imagen de entrada binarizada, mientras que en la Figura 10 se puede observar la misma imagen en escala de grises. Finalmente, en la Figura 11 se presenta el resultado de la obtención del canal oscuro.



Figura 8. Imagen original con haze.

Binarización

La binarización se realizó mediante segmentación por umbral, en la cual los píxeles toman solo uno de dos valores posibles: 0 si el valor del píxel es menor al umbral y 1 si el valor del píxel es mayor o igual al umbral.



Figura 9. Binarización en Nexys 4.

Escala de grises



Figura 10. Escala de grises en Nexys 4.

Canal oscuro

Finalmente se obtuvo el canal oscuro con una sola imagen.



Figura 11. Obtención del DC en Nexys 4.

CONCLUSIONES

Previamente a realizar la arquitectura en el FPGA, se implementó la metodología descrita en el presente trabajo para llevar a cabo dehazing mediante el algoritmo DCP en Matlab, lo cual nos ha servido para analizar las ecuaciones que están implicadas en el algo-



ritmo, y hemos podido analizar cómo interactúan el mapa de transmisión, el Airlight y el Canal Oscuro.

Con los experimentos realizados en Matlab se ha observado que la clave para recuperar una escena sin haze en una imagen es estimar un mapa de transmisión preciso. La implementación del algoritmo en el FPGA ha requerido una arquitectura bien estructurada para interactuar entre la PC, el FPGA y el almacenamiento en las memorias, y se logró diseñar una estructura robusta para realizar el procesamiento de dehazing en futuros trabajos.

Este trabajo pretende crear una base para implementar algoritmos de DCP y sus variantes en sistemas embebidos FPGA, puesto que su aplicación en TR es de gran interés en el área de la visión por computadora en aplicaciones de vehículos autónomos.

AGRADECIMIENTOS

Los autores agradecen al Consejo Nacional de Ciencia y Tecnología y a la Universidad Autónoma de Querétaro por su apoyo durante toda esta investigación.

REFERENCIAS

[1] Tan, R. T. (2008, June). Visibility in bad weather from a single image. In 2008 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

[2] He, K., Sun, J., & Tang, X. (2010). Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12), 2341-2353.

[3] He, R., Wang, Z., Xiong, H., & Feng, D. D. (2012, December). Single image dehazing with white balance correction and image decomposition. In 2012 International Conference on Digital Image Computing

[4] Zhang, B., & Zhao, J. (2016). Hardware implementation for real-time haze removal.

IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25(3), 1188-1192.

[5] Salazar-Colores, S., Ramos-Arreguin, J. M., Echeverri, C. J. O., Cabal-Yepez, E., Pedraza-Ortega, J. C., & Rodriguez-Resendiz, J. (2018). Image dehazing using morphological opening, dilation and Gaussian filtering. *Signal, Image and Video Processing*, 12(7), 1329-1335.

[6] Karlsruhe Institute of Technology (2019). The Kitti Vision Benchmark Suite. <http://www.cvlibs.net/datasets/kitti/>

[7] Digilent, A National Instruments Company (2019). Nexys 4. Digilent Documentation. <https://reference.digilentinc.com/reference/programmable-logic/nexys-4/start>.