Revista ELECTRO, Vol. 46, 2024, pp. 387 - 393
https://itchihuahua.mx/revista_electro
ISSN Electrónico: 3061-774X

DETECCIÓN DE PROXIMIDAD DE VEHÍCULOS CON VISIÓN ESTÉREO

Isidro Robledo Vega, Lorenzo Araiza Moreno, Carmen L. García Mata, Rogelio E. Baray Arana

Tecnológico Nacional de México / Instituto Tecnológico de Chihuahua
División de Estudios de Posgrado e Investigación
Ave. Tecnológico #2909, Col. 10 de mayo, Chihuahua, México
Tel. +52(614)201-2000 Ext. 2114
[isidro.rv, M20061534, carmen.gm, rogelio.ba]@chihuahua.tecnm.mx

RESUMEN.

Los sistemas de asistencia de conducción avanzados son tecnologías diseñadas para aumentar la seguridad en los vehículos al incrementar la habilidad del conductor a reaccionar a situaciones de peligro. En este artículo se presenta el desarrollo de un sistema de visión embebido para alertar sobre vehículos próximos en la parte posterior de un auto. Se utilizó una tarjeta embebida y dos cámaras para capturar pares de imágenes en estéreo desde la parte trasera de un vehículo. Se usa un modelo pre-entrenado de una red neuronal de aprendizaje profundo para detectar vehículos en los pares de imágenes en estéreo y se probaron diferentes métodos para el emparejamiento de las cajas delimitadoras de los vehículos detectados. Se determinó que el método de Intersección sobre la Unión (IoU) dio los mejores resultados de emparejamiento. Se calcula la disparidad o diferencia en posición de los vehículos detectados en los pares de imágenes en estéreo para determinar su proximidad, esto permite alertar al conductor si un vehículo rebasa un umbral de proximidad permitido para una conducción segura.

Palabras Clave: Visión en estéreo, detección de vehículos, redes neuronales de aprendizaje profundo, sistemas embebidos, sistemas de asistencia de conducción.

ABSTRACT.

Advanced driver assistance systems are technologies designed to increase vehicle safety by increasing the driver's ability to react to dangerous situations. This paper presents the development of an embedded vision system to warn about approaching vehicles in the rear of a car. An embedded card and two cameras were used to capture pairs of stereo images from the back of a vehicle. A pre-trained deep neural network model is used to detect vehicles in the pairs of stereo images and different methods were tested for bounding box matching of the detected vehicles. It was determined that the Intersection Over Union (IoU) method gave the best matching results. The disparity or difference in position of the vehicles detected in the pairs of stereo images is calculated to determine their proximity, this allows the driver to be alerted if a vehicle exceeds a proximity threshold allowed for safe driving.

Keywords: Stereo vision, vehicle detection, deep neural network, embedded systems, advanced driver assistance systems.

1. INTRODUCCIÓN

Los vehículos autónomos son la tendencia del futuro que van a desempeñar un papel esencial en nuestra vida cotidiana en varios escenarios. Por ejemplo, la tecnología de los vehículos autónomos tendrá un rol crucial para asistir a los discapacitados para trasladarse de un lugar a otro. Adicionalmente, eliminar el manejo bajo la influencia de sustancias será otra contribución de la tecnología autónoma [1]. De acuerdo a cifras del INEGI, en 2023 se registraron en México un total de 4,803 muertes relacionadas con accidentes de tránsito, de estas fatalidades, 4,419 muertes se le atribuyen al conductor del vehículo, lo que equivale a más del 90% del total [2]. Los vehículos autónomos tienen el potencial de reducir dramáticamente el número de accidentes causados por errores y negligencia del conductor.

En el campo de la visión por computadora, la navegación en exteriores de vehículos móviles es una de las aplicaciones que se les dedica más recursos de investigación. En general, un sistema guiado por visión por computadora aplicado a la navegación en exteriores se puede dividir principalmente en tres tareas de percepción:1) determinar la geometría del camino; 2) detección de obstáculos; y 3) reconocimiento de señales de tráfico.

Hoy en día, muchos fabricantes de autos ofrecen ayudas de manejo avanzadas en algunos de sus vehículos que permiten a éstos un cierto grado de autonomía. Estas ayudas de conducción pueden consistir en operar el acelerador y los frenos, estacionar el vehículo de manera automática, hacer cambios de dirección sin la intervención del conductor. También se ofrecen sistemas más avanzados, como el Autopilot de Tesla ó el Super Cruise de General Motors. A estos sistemas se les conocen como Sistemas de Conducción Automatizada (ADS - *Automated Driving Systems*) y Sistemas de Asistencia de Conducción Avanzada (ADAS - *Advanced Driver Assistance Systems*). Si bien, los vehículos autónomos, los ADS y los ADAS prometen una segura, confortable y eficiente experiencia de manejo, las fatalidades que involucran vehículos equipados con este tipo de sistemas van en ascenso [3]. El potencial de los ADS no podrá ser alcanzado mientras no se mejore la robustez en los sistemas mediante la investigación y desarrollo de nuevas tecnologías.

Si bien mucho se habla de la capacidad del vehículo para poder observar lo que sucede al frente, tener una buena percepción de lo que sucede en la parte trasera del vehículo es igual de importante. Los sensores montados en un vehículo deben ser capaces, no solamente de saber si el vehículo conducido se está acercando a un obstáculo, sino también deben de detectar los vehículos que se aproximen en la parte trasera, incluyendo los

puntos ciegos, lo cual es una ayuda importante para realizar maniobras de cambio de carril y rebase de forma segura.

Se desarrolló un sistema de visión embebido para ser instalado en un vehículo de manera que las cámaras puedan capturar imágenes de la parte trasera. El sistema desarrollado es capaz de detectar vehículos y alertar al conductor en caso de que alguno de estos exceda un umbral de proximidad pre-establecido. Primero se desarrolló el sistema de adquisición de imágenes utilizando dos cámaras de video en estéreo conectadas a una tarjeta embebida Nvidia Jetson Xavier NX. Después se desarrolló el software de detección de vehículos basado en el modelo pre-entrenado YOLOv4, luego se realizó el emparejamiento de las cajas contenedoras de los vehículos detectados en los pares de imágenes en estéreo y, por último, se estima la disparidad que existe entre la posición de los vehículos detectados en las imágenes. Esto permite al sistema determinar si algún vehículo se aproxima demasiado y alertar al conductor sobre una posible colisión.

El área de investigación sobre emparejamiento en estéreo (*Stereo Matching*) es de las más activas dentro del área de visión por computadora. El emparejamiento en estéreo es un proceso que genera correspondencias densas a partir de pares de imágenes en estéreo para crear mapas de disparidad para la estimación de profundidad. En [4] se describen dos tipos de algoritmos de emparejamiento en estéreo basados en programación explícita y en aprendizaje profundo, se menciona que con el avance del aprendizaje de máquina se ha mejorado la exactitud pero que la velocidad de procesamiento sigue siendo un reto. Se han desarrollado modelos de redes neuronales de aprendizaje profundo (DNN - *Deep Neural Network*) que realizan la detección de rasgos y miden la disparidad a partir de pares de imágenes en estéreo [5-6] y otras que además realizan detección de objetos generando cajas delimitadoras en 3D [7-10]. Los métodos de emparejamiento en estéreo analizan imágenes completas y en algunos casos utilizan modelos de DNN en 3D, por lo que requieren demasiados recursos computacionales y son difíciles de implementar en tiempo real. En [11] se propone un modelo de DNN que usa como entrada un par de imágenes en estéreo y las cajas delimitadoras de los objetos de interés para calcular la disparidad de todos los puntos dentro de las cajas delimitadoras obtenidas previamente con otro modelo de DNN.

En esta investigación se utiliza un modelo de DNN pre-entrenado para detectar vehículos en los pares de imágenes en 2D, se realiza el emparejamiento de las cajas delimitadoras de los vehículos detectados y solo se calcula la disparidad sobre algunos puntos de interés dentro de las cajas delimitadoras emparejadas. La contribución de esta investigación se encuentra en la comparación de diferentes métodos para el emparejamiento de las cajas delimitadoras de los vehículos detectados por el modelo DNN. Se probaron diferentes métodos basados en rasgos como ORB, AKAZE y SIFT, además del método de Intersección sobre la Unión (IoU – *Intersection over Union*) basado en el área de traslape de las cajas contenedoras.

2. SISTEMA DE VISIÓN EMBEBIDO

Se construyó el sistema de visión embebido utilizando como plataforma base la tarjeta Nvidia Jetson Xavier NX. Esta tarjeta embebida cuenta con un CPU Nvidia Carmel ARM v8.2 de 6 núcleos, un GPU Nvidia Volta de 384 núcleos con 48 núcleos tensores y 8GB de memoria RAM de 128 bits LPDDR4x con una velocidad de 59.7GB/s. Estas características técnicas le permiten ejecutar modelos de DNN de forma eficiente al implementarlos mediante la interfaz de programación de aplicaciones (API por sus siglas en inglés) denominada TensorRT [12] desarrollada por Nvidia utilizando las funciones de la librería CUDA.

Se utilizaron dos cámaras Arducam IMX477 Mini las cuales fueron diseñadas para usarse con la tarjeta Jetson Xavier NX por medio de los puertos de interfaz MIPI CSI-2. Se caracterizan por su tamaño compacto y buena calidad de imagen. Tienen una resolución máxima de 4032x3040 pixeles a 30 cuadros por segundo (12.3 Megapixeles) y una lente de enfoque manual con distancia focal de 3.9mm. Se diseño una base para montar las cámaras de forma que los planos de imagen queden alineados horizontalmente y no se muevan, la base fue montada en una carcasa donde se coloca la tarjeta Jetson Xavier NX. La base de las cámaras y la carcasa fueron impresas en 3D. La Figura 1 muestra el sistema de visión en estéreo que fue construido para esta investigación.

Figura 1.- Sistema de visión embebido.

La disposición de las cámaras colocadas en la base permite adquirir pares de imágenes en estéreo que capturan la misma escena con suficiente traslape. Esto permite realizar la calibración del sistema con pares de imágenes alineadas horizontalmente.

La tarjeta Nvidia Jetson Xavier NX ejecuta una versión modificada del sistema operativo Ubuntu, el software para la adquisición y procesamiento de los pares de imágenes en estéreo se desarrolló en lenguaje Python con el uso de funciones de las librerías OpenCV y Numpy. También se utilizaron las librerías PyCUDA y TensorRT para la ejecución del modelo de DNN pre-entrenado.

Se realizó el proceso de calibración del sistema de visión estéreo para verificar la alineación de los planos de las cámaras. Se usaron las funciones de la librería OpenCV para llevar a cabo este proceso [13]. Se utilizó un patrón de calibración con la forma de un tablero de ajedrez de 9x6 cuadros de 2.5 cms por lado, se adquirieron 20 pares de imágenes en estéreo donde aparece el patrón de calibración en diferentes posiciones. Se aplicó la fución para detección de las esquinas del tablero de ajedrez. La Figura 2 muestra un par de imágenes en estéreo donde se observan las esquinas del tablero de ajedrez detectadas.

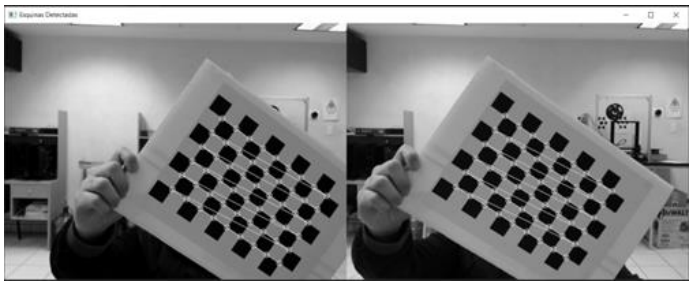


Figura 2. Par de imágenes en estéreo adquiridas con el sistema de visión embebido para el proceso de calibración.

Las coordenadas de los puntos detectados permiten estimar la posición geométrica del plano del tablero en coordenadas en 3D del mundo real y su proyección en coordenadas 2D del plano de la imagen. Este proceso se lleva a cabo por medio de funciones para obtener las matrices de parámetros extrínsecos, intrínsecos y los coeficientes de distorsión geométrica de cada cámara, luego se utilizan estos parámetros para la calibración del sistema estéreo que calcula una matriz de transformación con parámetros de translación y rotación que relacionan la posición de los ejes coordenados de las cámaras. Finalmente se utiliza la función de rectificación estéreo que calcula la rotación de las matrices de cada cámara para alinear sus planos, como resultado se obtiene la matriz de reproyección Q que tiene la forma:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -cx_1 \\ 0 & 1 & 0 & -cy \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T_x} & \frac{cx_1 - cx_2}{T_x} \end{bmatrix} \quad (1)$$

Donde f es la longitud focal de las cámaras, T_x es la distancia entre las cámaras o línea base, cx_1 y cx_2 son los centros de las imágenes de las cámaras. Para esta invetigación las cámaras deben estár alineadas en el eje *x* por lo que $cx_1 = cx_2$ y Q[3,3] debe ser igual a cero. La finalidad del proceso de calibración en estéreo es verificar que se cumpla esta última condición y, si es necesario, realizar ajustes al sistema de cámaras en estéreo.

3. DETECCIÓN DE VEHÍCULOS

La detección de vehículos se lleva a cabo de forma separada en cada una de las imágenes capturadas por el sistema de visión embebido. Se utilizó el modelo pre-entrenado de red neuronal de aprendizaje profundo denominado YOLOv4 [14], este es un algoritmo de detección de objetos en tiempo real que identifica objetos específicos en videos, transmisiones en vivo o imágenes fijas, utiliza los rasgos aprendidos por una red neuronal convolucional profunda. Es una versión mejorada de las versiones anteriores de YOLO. La arquitectura original está formada por 24 capas convolucionales, seguida de 2 capas completamente conectadas, YOLOv4 está formado por 53 capas convolucionales y fue entrenado utilizando la base de datos COCO. El algoritmo es capaz de reconocer 80 objetos diferentes y funciona de la siguiente forma: primero se redimensiona la imagen de entrada a 448 x 448 pixeles; luego se realiza una sola pasada por la red convolucional sobre la imagen y finalmente se aplica un umbral sobre el valor de confianza para obtener las detecciones resultantes.

En un principio, se utilizó la versión de YOLOv4 desarrollada en Darknet, que es un framework de código abierto para el desarrollo de DNNs escrito en Lenguaje C, utiliza la librería de CUDA para acelerar el procesamiento por medio del uso de GPUs. Las primeras versiones de YOLO se desarrollaron sobre Darknet, sin embargo, no es tan eficiente dentro del ambiente de desarrollo de NVIDIA, ya que no aprovecha al máximo el hardware. Entoces se utilizaron los programas del kit de desarrollo de software (SDK por sus siglas en inglés) de la tarjeta Jetson Xavier NX que permiten convertir un modelo de DNN primero al formato de intercambio ONNX y luego a TensorRT. De esta forma se obtuvo la versión de YOLOv4 sobre TensorRT que proporciona baja latencia en operaciones en tiempo real y que optimiza el modelo para proporcionar un mejor desempeño [12]. En esta invetigación se requiere detectar vehículos, por lo que se limitaron las detecciones de YOLOv4 solamente a objetos que cumplen con esta característica, es decir, el sistema solamente detecta las siguientes clases de vehículos:

- Bicycle (Bicicleta)
- Car (Automóvil)
- Motorcycle (Motocicleta)
- Airplane (Avión)
- Bus (Autobús)
- Train (Tren)
- Truck (Camión de carga)
- Boat (Bote)

La Figura 3 muestra un par de imágenes en estéreo que fueron procesadas por YOLOv4 en modo de inferencia. Se obtiene como resultado las coordenadas de las cajas delimitadoras que contienen los vehículos detectados en la escena.

Figura 3.- Vehículos detectados por YOLOv4 en un par de imágenes en estéreo.

4. EMPAREJAMIENTO DE CAJAS DELIMITADORAS

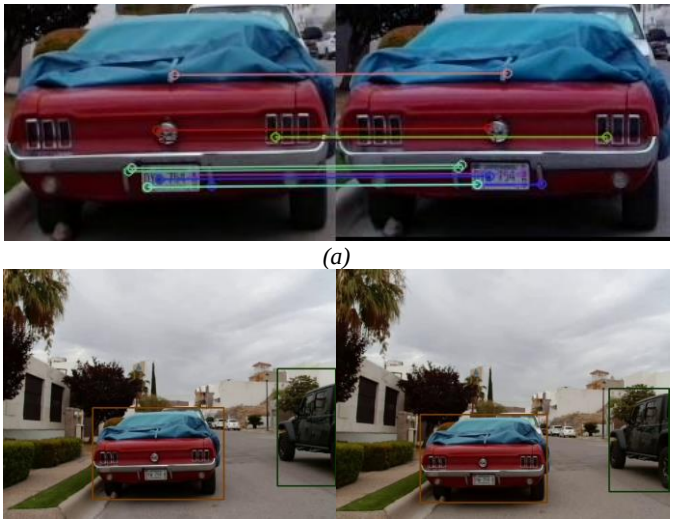
Para estimar la profundidad de los objetos de una escena capturados en un par de imágenes en estéreo es necesario establecer las correspondencias entre los objetos de las dos imágenes. En el caso de esta investigación se debe determinar correspondencia de cada vehículo detectado en la imagen izquierda con algún vehículo detectado en la imagen derecha. Para establecer las correspondencias se lleva cabo el proceso de emparejamiento de cajas delimitadoras, es decir, se utiliza la información contenida dentro de las cajas delimitadoras para determinar si se trata del mismo vehículo.

El proceso de emparejamiento de cajas delimitadoras consiste en tomar cada caja delimitadora de un vehículo detectado en la imagen izquierda y compararla con todas las cajas delimitadoras de los vehículos detectados en la imagen derecha. Debido a la geometría del sistema estéreo y las características del detector de vehículos, se pueden establecer algunas restricciones para limitar el emparejamiento, estas restricciones son: 1) la clase o etiqueta del vehículo detectado (auto, camión, bicicleta, etc.) debe ser la misma en las cajas delimitadoras de las imágenes izquierda y derecha; 2) no deberá haber diferencia en la alineación horizontal de las cajas delimitadoras por arriba de un umbral pre-establecido; 3) no deberá haber diferencia en el área total (ancho x alto) de las cajas delimitadoras por arriba o abajo un porcentaje pre-establecido.

Se probaron cuatro diferentes métodos para el emparejamiento de las cajas delimitadoras, tres de ellos están basados en la detección, descripción y emparejamiento de rasgos y el cuarto método está basado en el traslape de las áreas que ocupan las cajas delimitadoras.

El primer método basado en la detección, descripción y emparejamiento de rasgos, puntos clave o *keypoints* es ORB (Oriented FAST and Rotated BRIEF) [15], este es un detector y descriptor de rasgos eficiente que combina el detector FAST y el descriptor BRIEF con algunas modificaciones para mejorar el rendimiento. Se detectan los *keypoints* y se calculan sus descriptores dentro de cada caja delimitadora y luego se lleva a cabo el emparejamiento de los *keypoints* detectados para determinar si se emparejan estas dos cajas. Ya que los descriptores ORB son de tipo binario, se aplica el método de emparejamiento de *keypoints* por Fuerza Bruta utilizando la distancia de Hamming y un umbral de 4 *keypoints* emparejados para considerar que los vehículos dentro de las cajas son los mismos. Se realizaron pruebas de emparejamiento de cajas delimitadoras con ORB. La Figura 4(a) muestra los keypoints detectados y emparejados en un par de cajas delimitadoras y la Figura 4(b) muestra las cajas delimitadoras emparejadas en un par de imágenes en estéreo.

El segundo método basado en la detección, descripción y emparejamiento de rasgos, puntos clave o *keypoints* es AKAZE [16], este método utiliza un enfoque de detección y descripción rápida para rasgos en múltiples escalas que explota los beneficios de los espacios de escala no lineales, siendo más robusto a variaciones de luz y ruido. Es una variante del método original denominado KAZE que acelera su ejecución.

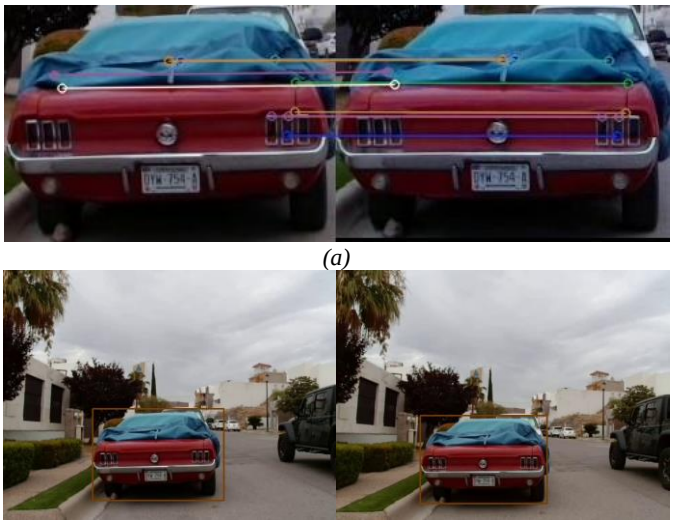


(a)

(b)

Figura 4.- Emparejamiento de (a) keypoints y (b) cajas delimitadoras con ORB.

Se lleva a cabo la detección de *keypoints* y se calculan sus descriptores en cada caja delimitadora y luego se lleva a cabo el emparejamiento de los *keypoints* detectados, los descriptores de AKAZE también son binarios, por lo que se utiliza el método de Fuerza Bruta con distancia de Hamming y un umbral de 4 *keypoints* emparejados para considerar el emparejamiento de las cajas. La Figura 5(a) muestra los *keypoints* detectados y emparejados con AKAZE en un par de cajas delimitadoras y la Figura 5(b) muestra las cajas delimitadoras emparejadas en un par de imágenes en estéreo.



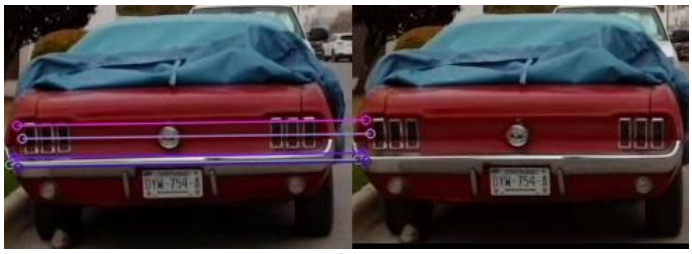
(a)

(b)

Figura 5.- Emparejamiento de (a) keypoints y (b) cajas delimitadoras con AKAZE.

Revista ELECTRO, Vol. 46, 2024, pp. 387 - 393
https://itchihuahua.mx/revista_electro
ISSN Electrónico: 3061-774X

El tercer método basado en la detección, descripción y emparejamiento de rasgos, puntos clave o *keypoints* es SIFT [17], este método es robusto y ampliamente utilizado, los rasgos son detectados mediante el cálculo de un conjunto de filtros de sub-octavas de diferencias de gaussianos (DoG), buscando máximos en la estructura en 3D resultante y luego calcula la ubicación en espacio+escala a nivel de sub-pixeles utilizando ajuste cuadrático. Se asigna una orientación a cada punto clave para lograr invariancia a la rotación. SIFT pueden detectar puntos clave con la misma ubicación y escala, pero en diferentes direcciones. Se lleva a cabo la detección de *keypoints* y se calculan sus descriptores en cada caja delimitadora y luego se lleva a cabo el emparejamiento de los *keypoints* detectados. SIFT produce descriptores con datos de punto flotante, por lo que se utiliza el método de Fuerza Bruta con distancia Euclidiana (Norm-L2). Se utiliza el método knnMatch() con k=2 para encontrar los k *keypoints* más cercanos al punto de prueba, los métodos ORB y AKAZE utilizan el método match() que solo encuentra el *keypoints* más cercano. Se establece un umbral de 4 *keypoints* emparejados para considerar el emparejamiento de las cajas. La Figura 6(a) muestra los *keypoints* detectados y emparejados con SIFT en un par de cajas delimitadoras y la Figura 6(b) muestra las cajas delimitadoras emparejadas en un par de imágenes en estéreo.



(a)

(b)

Figura 6.- Emparejamiento de (a) keypoints y (b) cajas delimitadoras con SIFT.

El cuarto método es el de Intersección sobre la Unión (IoU por sus siglas en inglés) [17], este método es una evaluación métrica usada para medir la similitud de objetos basado en el traslape entre las cajas delimitadoras que los contienen y puede ser usada para comparar objetos potencialmente correspondientes en ambas imágenes de un sistema de visión estéreo. IoU es una métrica crucial para evaluar algoritmos de segmentación, también es conocido como índice de Jaccard, ya que cuantifica qué tan bien el algoritmo puede distinguir los objetos del fondo en una imagen. IoU es el rango de la intersección de las áreas de dos cajas delimitadoras y de sus áreas combinadas. Matemáticamente se escribe como:

$$IoU = \frac{area\_caja1 \cup area\_caja2}{area\_caja1 \cap area\_caja2} \quad (2)$$

Se calcula el numerador de la Ecuación 2 mediante la intersección de las áreas de las cajas delimitadoras A y B, y el denominador mediante la unión de las áreas de las cajas delimitadoras A y B.

Para llevar a cabo el emparejamiento de las cajas delimitadoras, primero se calcula la métrica de IoU entre cada par de cajas delimitadoras y se guardan estos valores en una matriz, en la cual, los índices de los reglones corresponden a las cajas delimitadoras de la imagen izquierda y los índices de las columnas corresponden a las cajas delimitadoras de la imagen derecha. Se determina el valor mayor de esta matriz, los índices del renglón y columna del valor mayor se agregan a la lista de cajas delimitadoras emparejadas para después hacer cero los valores de este reglón y columna. Se repite este proceso hasta que ya no haya un valor mayor, es decir, todos los elementos de la matriz son cero, lo que indica que se realizaron todos los emparejamientos posibles. La Figura 7 muestra las cajas delimitadoras emparejadas con el método de IoU en un par de imágenes en estéreo.

Figura 7.- Emparejamiento de cajas delimitadoras con IoU.

Se realizaron pruebas con el sistema de visión embebido capturando 4 pares de videos desde la parte posterior de un vehículo, de los cuales 1 fue tomado con el vehículo en movimiento y 3 con el vehículo estacionado. El número de cuadros de imagen por video varía entre 240 y 1300 para un total de 3230 cuadros de imagen. En cada cuadro aparecen varios vehículos que son detectados con muy pocos errores por el modelo YOLOv4. Se realizó la detección, descripción y emparejamiento de rasgos dentro de las cajas delimitadoras de los vehículos detectados en cada par de imágenes en estéreo por medio de los métodos ORB, AKAZE y SIFT y el emparejamiento directo de las cajas delimitadoras con el método de IoU. En la Tabla 1 se muestra el total de cajas delimitadoras de vehículos detectados presentes los pares de imágenes en estéreo y el porcentaje de detecciones correctas.

Tabla 1.- Resultados del emparejamiento de cajas delimitadoras con cuatro diferentes métodos.

Método	Captura de Video con Auto:	Total de cajas delimitadoras a emparejar	Porcentaje de cajas delimitadoras emparejadas
ORB	Estacionado	4713	46.93%
	En movimiento	9711	6.26%
AKAZE	Estacionado	4713	41.22%
	En movimiento	9711	3.87%
SIFT	Estacionado	4713	54.75%
	En movimiento	9711	30.81%
IoU	Estacionado	4713	**99.89%**
	En movimiento	9711	**73.98%**

El método de IoU fue casi perfecto en los emparejamientos cuando el sistema de visión embebido se encuentra en un auto estacionado con vehículos cercanos estacionados y en movimiento. Cuando el auto está en movimiento el porcentaje de cajas delimitadoras emparejadas con el método de IoU disminuye, pero sigue siento mejor a los métodos basados en rasgos.

5. ESTIMACIÓN DE PROXIMIDAD

La matriz de reproyección Q de la Ecuación 1 permite determinar la posición en 3D de un punto dado en la imagen izquierda y su punto correspondiente en la imagen derecha. Dado un punto de la imagen con coordenadas homogéneas en dos dimensiones (x,y) y su disparidad asociada d, según el algoritmo de Bouguet [18] se puede proyectar este punto hacia un espacio en 3D por medio de la siguiente ecuación:

$$Q\begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \tag{3}$$

Si se toma como referencia un punto específico de la imagen izquierda (x_l, y_l) y su punto correspondiente en la imagen derecha (x_r, y_r) dentro de las cajas emparejadas con vehículos detectados de un par de imágenes en estéreo, la disparidad estaría dada por $d = x_l - y_l$, ya que las imágenes están alineadas horizontalmente. Entonces la profundidad P en ese punto pude estimarse por:

$$P = \frac{Q[2,3]}{d*Q[3,2]+Q[3,3]} \tag{4}$$

En la Ecuación 4 se observa que la disparidad d es inversamente proporcional a la profundidad P, por lo que se puede estimar la proximidad de un vehículo basado en los valores de disparidad. Entre mayor sea la disparidad, más próximo estará el vehículo detectado, entonces se establece un umbral de disparidad para determinar si un vehículo se acerca demasiado y poder generar una alerta al conductor.

En el caso de los métodos de emparejamiento de cajas delimitadoras basados en rasgos (ORB, AKAZE y SIFT) se calculó la disparidad para cada rasgo emparejado dentro de las cajas delimitadoras y se seleccionó el valor mayor, que corresponde al punto más cercano. En el caso del método de IoU, se tomó como referencia el centro de las cajas emparejadas para el cálculo de la disparidad. Para el sistema de visión embebido desarrollado en esta investigación se estableció un umbral de disparidad $d_{max}=10$ que representa una profundidad de 5.2 metros. La Figura 8 muestra un par de imágenes en estéreo donde los vehículos detectados que superan el umbral de disparidad se encierran en un recuadro de color rojo a modo de alerta.

Figura 8.- Vehículos detectados que superan el umbral de disparidad.

En la Figura 8 el emparejamiento de cajas delimitadoras se llevó a cabo con SIFT, se calculó la disparidad a partir de los rasgos emparejados entre las cajas de las dos imágenes. La estimación de proximidad fue mejor con los métodos basados en rasgos, siendo SIFT el método que proporcionó mejores resultados debido a que supera a ORB y AKAZE en el emparejamiento de cajas delimitadoras.

6. CONCLUSIONES

Se construyó un sistema de visión embebido que fue colocado en la parte trasera de un auto para detectar vehículos y alertar si alguno se aproxima demasiado. La proximadad de los vehículos detectados se estima utilizando pares de imágenes en estéreo adquiridas por las cámaras del sistema. Se calcula la disparidad sobre puntos de interés dentro de las cajas delimitadoras de los vehículos detectados que fueron emparejadas en cada par de imágenes en estéreo capturada por el sistema.

La principal contribución de esta investigación radica en probar cuatro diferentes métodos de emparejamiento de cajas delimitadoras, tres basados en la detección, descripción y emparejamiento de rasgos: ORB, AKAZE y SIFT; y otro método basado en el traslape de las áreas de las cajas delimitadoras denominado Intersección sobre la Unión (IoU). Las pruebas realizadas permitieron observar que el método de Intersección sobre la Unión obtuvo los mejores resultados en el emparejamiento de cajas delimitadoras, aunque la mayoría de fallos de emparejamiento en los métodos basados en rasgos fueron en vehículos alejados cuando las cajas delimitadoras son muy pequeñas y no se alcanzan a detectar suficientes rasgos.

En cuanto al cálculo de disparidades para determinar la proximidad de los vehículos detectados, SIFT proporcionó los mejores resultados. En este caso, el método de Intersección sobre la Unión no fue tan confiable, ya que no es tan preciso usar el centro de las cajas como referencias para el cálculo de la disparidad.

Se observó que SIFT proporciona los mejores resultados para alertar al conductor sobre vehículos muy próximos, aunque tiene fallos en el emparejamiento de cajas delimitadoras, estos suceden cuando los vehículos están alejados. Se ha considerado mejorar el método de Intersección sobre la Unión (IoU) detectando y emparejando puntos de interés dentro de las cajas emparejadas para mejorar el cálculo de disparidad.

El sistema de visión embebido desarrollado procesa un promedio de 3 pares de imágenes en estéreo por segundo con los diferentes métodos de emparejamiento de cajas delimitadoras, lo que es suficiente como sistema de alerta de conducción.

Agradecemos al Tecnológico Nacional de México por el apoyo para la realización de esta investigación por medio del proyecto 21048.24-P.

7. BIBLIOGRAFÍA

[1] Zhou, C., Li, F., Cao, W., "Architecture, design and implementation of image based autonomous car: THUNDER-1", *Multimedia Tools and Applications*, vol. 78, no. 20, pp. 28557-28573, 2019.

[2] Accidentes de tránsito terrestre en zonas urbanas y suburbanas. INEGI. [En línea]. Disponible en: https://www.inegi.org.mx/sistemas/olap/proyectos/bd/continuas/transporte/accidentes.asp? [Último acceso: 31-mayo-2024].

[3] Hawkins, A. J., The Verge, 15 Junio 2022. [En línea]. Disponible en: https://www.theverge.com/2022/6/15/23168088/nhtsa-adas-self-driving-crash-data-tesla. [Último acceso: 31-mayo-2024]

[4] Liu, CW., Wang, H., Guo, S., Bocus, M.J., Chen, Q., Fan, R., "Stereo Matching: Fundamentals, State-of-the-Art, and Existing Challenges", *In: Fan, R., Guo, S., Bocus, M.J. (eds) Autonomous Driving Perception. Advances in Computer Vision and Pattern Recognition*. Springer, Singapore. https://doi.org/10.1007/978-981-99-4287-9_3

[5] Chen, S., Xiang, Z., Qiao, Ch., Chen, Y., Bai, T., "PGNet: Panoptic parsing guided deep stereo matching", *Neurocomputing*, vol. 463, pp. 609-622, 2021, doi: 10.1016/j.neucom.2021.08.041

[6] H. Wu, H. Su, Y. Liu and H. Gao, "Object Detection and Localization Using Stereo Cameras", *5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, Shenzhen, China, 2020, pp. 628-633, doi: 10.1109/ICARM49381.2020.9195365

[7] J. Choe, K. Joo, F. Rameau and I. So Kweon, "Stereo Object Matching Network", *IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, pp. 12918-12924, doi: 10.1109/ICRA48506.2021.9562027

[8] H. Königshof, N. O. Salscheider and C. Stiller, "Realtime 3D Object Detection for Automated Driving Using Stereo Vision and Semantic Information", *IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 1405-1410, doi: 10.1109/ITSC.2019.8917330

[9] P. Li, X. Chen and S. Shen, "Stereo R-CNN Based 3D Object Detection for Autonomous Driving", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019 pp. 7636-7644.
doi: 10.1109/CVPR.2019.00783

[10] W. Peng, H. Pan, H. Liu and Y. Sun, "IDA-3D: Instance-Depth-Aware 3D Object Detection from Stereo Vision for Autonomous Driving", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 13012-13021, doi: 10.1109/CVPR42600.2020.01303

[11] A. D. Pon, J. Ku, C. Li and S. L. Waslander, "Object-Centric Stereo Matching for 3D Object Detection", *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 8383-8389, doi: 10.1109/ICRA40945.2020.9196660.

[12] TensorRT, NVIDIA, [En línea]. Disponible en: https://developer.nvidia.com/tensorrt. [Último acceso: 31 mayo 2024].

[13] Camera Calibration and 3D Reconstruction. OpenCV. [En línea]. Disponible en: https://docs.opencv.org/4.10.0/d9/d0c/group__calib3d.html. [Última visita: 31-mayo-2024].

[14] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., "YOLOv4: Optimal speed and accuracy of object detection", arXiv preprint, arXiv:2004.10934, 2020.

[15] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: An efficient alternative to SIFT or SURF", *2011 International Conference on Computer Vision*, Barcelona, Spain, 2011, pp. 2564-2571, doi: 10.1109/ICCV.2011.6126544.

[16] P. F. Alcantarilla, J. Nuevo, A. Bartoli, "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces", *British Machine Vision Conference*, Bristol, U.K., 2013.

[17] Lowe, D.G., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004, doi.org/10.1023/B:VISI.0000029664.99615.94

[18] Kaehler, A., Bradski, G., Learning OpenCV 3: computer vision in C++ with the OpenCV library. O'Reilly Media, Inc., 2016.