Anteproyecto de Trabajo Fin de Máster

Fabio Sánchez García

27 de octubre de 2023

Autor: Fabio Sánchez García

Tutores: Luis Miguel Bergasa Pascual y Santiago Montiel Marín

Titulación: Máster Universitario en Ingeniería Industrial

Departamento: Departamento de Electrónica

Título: Detección de objetos 3D mediante fusión multimodal de imágenes de rango y RGB.

Título en inglés: Multimodal-based 3D Object Detection with Range Images and RGB Fusion.

1 Introducción

La visión por computador y sus tecnologías asociadas han experimentado un avance significativo en la última década, propiciando el progreso de aplicaciones como la robótica o la conducción autónoma entre otros. En este escenario, la detección y el reconocimiento de objetos en tres dimensiones (3D) supone un desafío crucial para permitir a los sistemas navegar e interactuar con su entorno. Este Trabajo de Fin de Máster (TFM) se enfoca en explorar y desarrollar una metodología eficaz para la detección de objetos 3D mediante la fusión de datos provenientes de dos sensores masivamente utilizados como cámaras y LiDAR.

La tecnología LiDAR está especializada en proporcionar información de profundidad, con lo que se complementa de manera efectiva con las cámaras, que son capaces de capturar información visual detallada (semántica) en el espectro RGB. Típicamente, las nubes de puntos o imágenes de rango obtenidas de LiDAR se suelen emplear en modelos de detección de objetos 3D como PointPillars [1]. Por otro lado, las imágenes RGB (*Red, Green, Blue*), que son esencialmente tensores tridimensionales que contienen información, se utilizan en detección de objetos 2D como YOLO ("*You Only Look Once*") [2]. En la Fig. 2 se puede observar un ejemplo de cada uno de estos datos de entrada. Existe una minoría de algoritmos que también utilizan imágenes RGB y que son capaces de estimar profundidad a partir únicamente de imágenes realizando detección de objetos 3D como SMOKE [3], pero sus métricas suelen ser considerablemente peores a los sistemas que sólo usan LiDAR.

Por tanto, esta combinación multimodal busca mejorar sustancialmente la precisión y robustez de los sistemas de detección de objetos 3D. Sin embargo, llevar a cabo la fusión de estos datos puede suponer un reto significativo para la interpretación conjunta de la información.

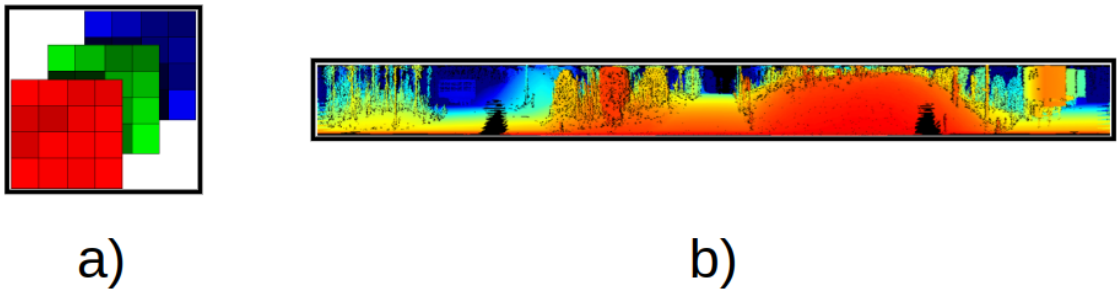

Figura 2: a) Imagen RGB con sus 3 canales separados. b) Imagen de rango.

Con todo esto, el TFM pone el foco en el *Deep Learning* (DL), una tecnología en auge en la actualidad, que se centra en la construcción y entrenamiento de redes neuronales profundas. Otro campo relevante para el trabajo es la fusión sensorial, ya que la información que contienen los datos utilizados debe mezclarse de forma que mejore el rendimiento de estas redes.

Como se ha mencionado anteriormente, una de las principales áreas de aplicación del DL es la conducción autónoma. Este trabajo se realiza en el seno del RobeSafe Reseach Group de la UAH, estableciendo así una fuerte conexión con esta temática. Del mismo modo, la cámara y el LiDAR componen la mayor parte de las soluciones de percepción en los vehículos autónomos, por lo que el trabajo se sitúa fuertemente en el estado del arte.

En cuanto a la procedencia de los datos, en el campo del *Deep Learning*, se utilizan los denominados *datasets* o conjuntos de datos, en los que se agrupan enormes cantidades de datos para entrenar y validar redes neuronales. Como ya se ha comentado, estos datos pueden ser de diferente naturaleza, dependiendo del propósito para el que se vayan a utilizar. En este caso, se usan datos de LiDAR y cámara. En conducción autónoma, existen tres principales *datasets*: KITTI [4], nuScenes [5] y Waymo [6]. Estos representarán la fuente principal de datos además de ser la base sobre la que se sustentará la evaluación del modelo desarrollado.

2 Objetivos

El principal objetivo de este trabajo es el estudio, implementación y evaluación de un modelo multimodal Deep Learning basado en la arquitectura YOLO para la detección de objetos 3D sobre *datasets* enfocados en conducción autónoma a partir de imágenes RGB de una o varias cámaras monoculares e imágenes de rango provenientes de LiDAR. Este enfoque se alinea con el estado del arte en el que se busca maximizar la eficacia de modelos que utilicen datos de estos sensores, a la par que maximizar la velocidad de inferencia.

YOLO es un arquitectura extremadamente rápida, ya que permite identificar objetos en imágenes con un solo paso a través de la red neuronal, en contraposición a otros métodos que requieren varias pasadas. Sin embargo, su restricción es que no tiene la capacidad de inferir profundidad, ya que es un modelo de detección 2D. Así, combinar imágenes RGB e imágenes de rango supone una potenciación y complementación de estos métodos de representación. Si

a esto se le suma una modificación de la arquitectura de YOLO para que pueda tomar como entrada estas imágenes mejoradas, y para que sea capaz de realizar inferencia de objetos en 3D, se pueden lograr resultados sorprendentemente precisos y rápidos.

Se busca que la red tenga un rendimiento similar al de otros métodos basados en nubes de puntos de LiDAR como *PointPillars* (contra el que se realizará una comparación) pero mejorando los tiempos de inferencia, como se observa en [7], donde se consiguen mejorar los tiempos de inferencia en gran medida manteniendo unas métricas de precisión al nivel de otros detectores del estado del arte. Además, se espera que el modelo sea capaz de mejorar notablemente modelos como SMOKE, que se centran en la obtención de profundidad sólo con cámara monocular.

Otro aspecto importante a explorar en el trabajo es la viabilidad de incorporar un sistema de múltiples cámaras. Esto es factible dado que el LiDAR captura datos en un espectro de 360º, por lo que las imágenes de rango encapsulan información dentro de dicho espectro. Al emplear un sistema multicámara que abarque el mismo rango, se posibilita una percepción global al entorno del sistema.

El objetivo final, alineado con el grupo de investigación RobeSafe es implementar la red diseñada para su funcionamiento en un arquitectura completa de vehículo autónomo, comprobando su eficacia y eficiencia en escenarios simulados o reales, pasando así de *datasets* externos y ajenos al grupo por verdaderos casos de uso.

3 Metodología y plan de trabajo

El proyecto a realizar tiene una duración prevista de unos 8 meses, entre los meses de noviembre de 2023 y junio de 2024. Para lograrlo, se planea una distribución temporal basada en objetivos:

1. Formación teórica previa. (1'5 meses)
    - Estudio de diferentes versiones de la arquitectura YOLO.
    - Estudio de *framework* de aprendizaje automático *PyTorch*.
    - Estudio de arquitecturas de modelos del estado del arte de detección de objetos 3D.
2. Estudio de bases de datos descritas anteriormente y generación de datos adaptados al formato requerido. (1 mes)
3. Modificación y/o desarrollo de la arquitectura YOLO para detección de objetos 3D. (2 meses)
    - Es la tarea medular del trabajo y la más compleja.
    - Consiste en modificar la entrada del modelo (*backbone*), así como la salida (*head* o cabeza), los bucles de entrenamiento y la función de pérdida. Será importante el paso previo de estudio teórico para tomar las decisiones adecuadas.
4. Evaluación del modelo, utilizando los datos generados a partir de LiDAR y cámara, con las métricas pertinentes para la tarea de detección de objetos en 3D sobre los datasets KITTI y Waymo o nuScenes. (1 mes)

5. Aplicación del modelo en un entorno simulado en el simulador hiperrealista CARLA. (15 días)

6. Documentación del proceso y escritura de la memoria. (2 meses)

4 Medios

Para poder desarrollar el trabajo mencionado es importante contar con ciertos medios tangibles e intangibles. Estos se pueden dividir en recursos *hardware* y *software*:

Hardware:

- Ordenador personal: memoria RAM 16 GB y tarjeta gráfica NVIDIA RTX 2070 con 8 GB de VRAM.
- Cámara monocular ZED, si es posible.
- LiDAR VLP-16 o VLS-128, si es posible.

Sotfware:

- Sistema operativo Ubuntu 22.04 LTS.
- Editor código fuente Visual Studio Code.
- Software de control de versiones Git.
- Lenguaje de programación Python.
- *Framework* de aprendizaje automático PyTorch.

Como ya se ha comentado, es importante el acceso a *datasets* de conducción autónoma como son KITTI, nuScenes y Waymo.

Referencias

[1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang y O. Beijbom, *PointPillars: Fast Encoders for Object Detection from Point Clouds*, 2019. arXiv: 1812.05784 [cs.LG].

[2] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, 2016. arXiv: 1506.02640 [cs.CV].

[3] Z. Liu, Z. Wu y R. Toth, "SMOKE: Single-Stage Monocular 3D Object Detection via Keypoint Estimation", en *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, jun. de 2020.

[4] A. Geiger, P. Lenz, C. Stiller y R. Urtasun, "Vision meets Robotics: The KITTI Dataset", *International Journal of Robotics Research (IJRR)*, 2013.

[5] H. Caesar, V. Bankiti, A. H. Lang y col., *nuScenes: A multimodal dataset for autonomous driving*, 2020. arXiv: 1903.11027 [cs.LG].

[6] P. Sun, H. Kretzschmar, X. Dotiwalla y col., *Scalability in Perception for Autonomous Driving: Waymo Open Dataset*, 2020. arXiv: 1912.04838 [cs.CV].

[7] M. Simon, S. Milz, K. Amende y H.-M. Gross, "Complex-YOLO: Real-time 3D Object Detection on Point Clouds", *arXiv*, 2018.