Universidad Zaragoza
1542

Trabajo Fin de Máster

Estimación de línea del horizonte en imágenes de 360 grados con aprendizaje automático profundo

360 degree image horizon line estimation using deep machine learning

Autor

Eduardo Fraguas Bordonaba

Director

Jesús Bermudez Cameo

Escuela de Ingeniería y Arquitectura

2020

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe entregarse en la Secretaría de su Centro junto con el resto de documentación del TFG/TFM para su evaluación)

D./Dª. Eduardo Fraguas Bordonaba ,en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster (Título del Trabajo)

Estimación de línea del horizonte en imágenes de 360 grados con aprendizaje automático profundo.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 16/11/2020

Fdo:

RESUMEN

A lo largo de este trabajo, se ha propuesto un método general para la rectificación vertical de la orientación (considerando como referencia absoluta la dirección de la gravedad) en todo tipo de imágenes panorámicas, tanto de interior como de exterior, en ambientes naturales y urbanos y con cualquier tipo de iluminación.

El desarrollo de este método se centra principalmente en el entrenamiento de una red neuronal convolucional mediante deep learning, que es capaz de detectar tanto la línea del horizonte como los puntos de fuga verticales de las imágenes panorámicas. Estos elementos de las imágenes nos permiten obtener información geométrica muy valiosa para conocer la orientación de la cámara.

El trabajo aborda todas las fases necesarias para alcanzar los objetivos planteados, comienza con la creación desde cero de un dataset suficientemente grande y variado para que el entrenamiento sea fructífero, esta creación del dataset reúne las tareas de recolección de fotos, procesamiento de las imágenes y generación del ground truth. Posteriormente se entrena la red en la que va a basarse nuestro método de ajuste vertical con el dataset creado.

A partir de las imágenes de salida de nuestra red, se desarrolla un algoritmo basado en el procesamiento de imágenes con el que podremos corregir la orientación de las imágenes panorámicas a partir de los mapas de píxeles de los puntos de fuga que obtenemos.

Por último se comparan nuestros resultados con el estado del arte mediante la experimentación sobre un método geométrico de detección de puntos de fuga basado en el algoritmo RANSAC, el cual se ha implementado con el fin de comparar la efectividad de nuestro método respecto a otros trabajos existentes y las ventajas de introducir redes neuronales profundas en el desarrollo del proceso.

ABSTRACT

Throughout this project, a general method is proposed for upright rectification (considering the direction of gravity as absolute reference) in all types of panoramic images, both indoors and outdoors, in natural and urban environments and with any type of lighting.

The development of this method is mainly focused on the training of a deep learning convolutional neural network, which is capable of detecting both the horizon line and the vertical vanishing points of panoramic images. These elements of the images allow us to obtain very valuable geometric information to know the orientation of the camera.

The project addresses all the necessary phases to achieve the proposed objectives, it begins with the creation from scratch of a large and varied dataset enough for the training to be successful, this creation of the dataset brings together the tasks of collecting photos, processing images and generation of the ground truth. Subsequently, the network on which our vertical adjustment method will be based is trained with the created dataset.

From the output images of our network, an algorithm based on image processing is developed in order to correct the orientation of panoramic images with the information of the pixel maps of the vanishing points that we obtain.

Finally, our results are compared with the state of the art through experimentation on a vanishing point detection method based on the RANSAC geometric based algorithm, which has been implemented in order to evaluate the effectiveness of our method and the advantages of introducing deep neural networks in the process development, compared to other existing work.

ÍNDICE

CAPÍTULO 1

INTRODUCCIÓN

En los últimos años han aparecido numerosas cámaras omnidireccionales en el mercado, algunos de los ejemplos más conocidos que podemos encontrar son la cámara Surround 360 de Facebook, la Google Jump, Jaunt VR, Omni de GoPro, Gear 360 de Samsung, OZO de Nokia, Theta de Ricoh, Immerge de Lytro, 360 CAM de LG o KeyMision 360 de Nikonn entre otros.

Estas cámaras, permiten capturar imágenes omnidireccionales esféricas de alta calidad, es decir, imágenes en las que tenemos un campo de visión horizontal de 360º y un campo de visión vertical de 180º. Gracias a la aplicación para realidad virtual (VR) y la accesibilidad y simplicidad de estos dispositivos, este tipo de imágenes se están popularizando. Por otro lado, el uso de la realidad virtual se está generalizando en diversos contextos y aplicaciones, como el entretenimiento, industria, turismo virtual, patrimonio cultural o publicidad inmobiliaria por nombrar algunos ejemplos.

Las imágenes omnidireccionales esféricas pueden ser representadas en cualquier pantalla en el formato tradicional equirectangular (también llamado formato panorámico), o con un dispositivo específico para inmersión en realidad virtual como las gafas de realidad virtual, donde el usuario, como en la vida real, puede girar la cabeza para visualizar cualquier dirección de la fotografía.

Figura 1.1 Imagen en formato equirectangular (izquierda) y en formato para realidad virtual.

En la práctica, cuando la referencia del sistema de realidad virtual y de la cámara omnidireccional no están alineadas, la imagen se ve inclinada, lo que reduce notablemente la calidad de la experiencia de realidad virtual y conduce a molestias visuales. Para poder tener una referencia común, normalmente nos interesa que la referencia vertical de la cámara se proyecte en los polos de la esfera y denominaremos a esta orientación “orientación horizontal” (ver Figura 1.1). En esta configuración la línea del horizonte es una línea recta que va de izquierda a derecha mientras que la dirección vertical se proyecta en la primera y última fila de la imagen, ya que estas filas representan los polos de la esfera. Si la imagen se captura en otra orientación la línea del horizonte se presenta como una línea curva ondulada (ver Figura 1.2).

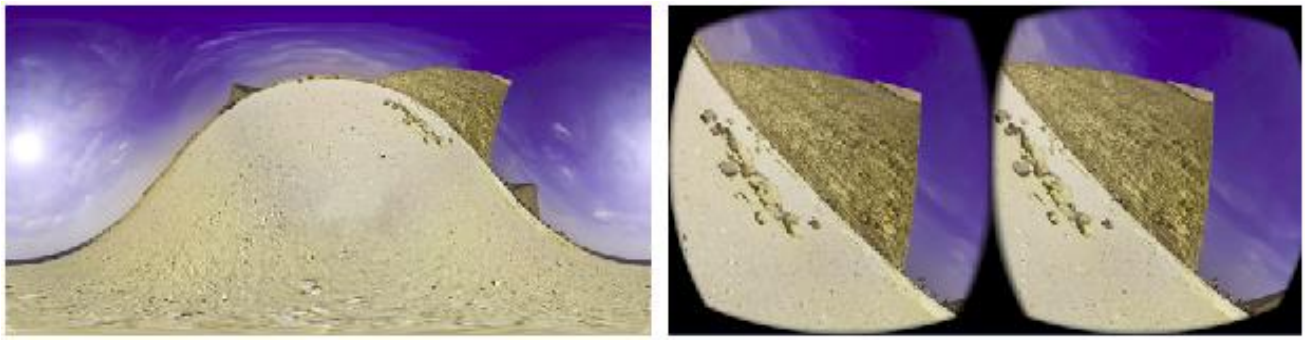


Figura 1.2 Imagen mal orientada en formato equirectangular y en formato para realidad virtual.

Sin embargo, al tener una proyección esférica, si se conoce la dirección vertical en la referencia de la cámara se puede rectificar esta sin problemas a una orientación horizontal que puede ser usada por el sistema de realidad virtual como imagen en la referencia base. Se hace necesario por tanto un método capaz de detectar la dirección vertical de la imagen, de manera que nos facilite la información necesaria para realizar esta rectificación. Si trabajamos con una secuencia de vídeo este algoritmo permitirá realizar el equivalente a una estabilización vertical mecánica.

Hoy en día podemos encontrar gran variedad de soluciones para el ajuste vertical de la imagen:

- **Mediante edición manual; re-alimentación visual de la horizontalidad de la línea del horizonte:** una solución simple consiste en utilizar este tipo de software como Adobe Premiere Pro CC con VR Plugin, que permite a los usuarios manualmente rectificar la orientación de la imagen. Este método, aunque pueda proporcionar unos resultados visualmente atractivos, tiene como principal inconveniente el tiempo que requiere para realizarse en la práctica ya que el proceso de ajuste debe repetirse manualmente para cada imagen.

- **Mediante hardware para detección de dirección de la gravedad:** Las unidades de medición inercial (IMUs) permiten estimar la dirección de la gravedad integrando la información acoplada de aceleración/gravedad proporcionada por el acelerómetro y la información de velocidad angular proporcionada por el giróscopo. En caso de que la adquisición se realice en reposo el acelerómetro proporciona directamente la dirección de la gravedad. Sin embargo, aumenta el coste de fabricación de la cámara al tener que añadir un sensor adicional, además del consumo de energía y la necesidad de la calibración de la cámara-giróscopo. Debido a esto, la mayoría de las principales cámaras omnidireccionales del mercado no incluyen IMU.

  Además, un enfoque basado en hardware no se puede aplicar como solución a imágenes ya capturadas por otras cámaras o para imágenes descargadas de internet sin ninguna información previa.

- **Mediante métodos de visión por ordenador:** Otra solución sería estimar la orientación de la imagen de entrada mediante métodos de visión por ordenador. Debido a la distorsión inherente y a los modelos de proyección específicos de las imágenes de realidad virtual, se han propuesto métodos dedicados a las imágenes de realidad virtual [1, 2]. Estos métodos utilizan la representación esférica de este tipo de imágenes y, por lo tanto, son aplicables a imágenes con un amplio campo de visión, como las obtenidas por cámaras VR.

Estos métodos pueden rectificar automáticamente la orientación de la imagen analizando las estructuras geométricas presentes en la imagen, especialmente líneas y puntos de fuga [1, 2] o línea del horizonte [3, 4]. Sin embargo, los métodos basados en líneas son aplicables solo a entornos urbanos artificiales en los que se puede encontrar una composición de líneas rectas. De manera similar, los métodos basados en el horizonte requieren de un horizonte fácil de reconocer, y por lo tanto no se puede aplicar en imágenes donde el horizonte no es claramente visible, como escenas interiores o imágenes tomadas en ambientes naturales como montañas o bosques.

1.1 ESTADO DEL ARTE

La estrategia que se usa de manera general en el ajuste vertical de las imágenes consiste en estimar la orientación (o rotación) de la cámara mediante el procesamiento de imágenes, generalmente mediante extracción de líneas, para posteriormente aplicar el inverso de esta rotación a la imagen reconstruyéndola con la orientación que se desea. Esta estrategia, originalmente se ha estudiado para la perspectiva estándar de las imágenes [5, 6].

Figura 1.3 Corrección vertical en una imagen de perspectiva clásica [5]

Por su parte, para la corrección de la orientación en imágenes de realidad virtual, se han propuesto otra serie de métodos debido a la distorsión inherente y a los modelos de proyección específicos de este tipo de imágenes [1, 2].

1.1.1 REPRESENTACIÓN ESFÉRICA DE LA IMAGEN

Los métodos existentes para la rectificación de la orientación en imágenes de realidad virtual suelen trabajar con una representación esférica de la imagen más habitualmente que con representaciones 2D.

La primera ventaja que nos encontramos en esta representación es que el punto de encuentro de las líneas paralelas, que conceptualmente se encuentra en el infinito, queda siempre representado en algún punto de la superficie limitada de la esfera. La segunda ventaja que presenta esta representación es que para modificar la orientación de la imagen tan solo es necesario aplicar una simple rotación esférica.

La representación esférica ha sido usada en imágenes tradicionales de modelo pinhole [7], y es directamente aplicable a imágenes equirectangulares gracias a su campo de visión esférico. Esta representación ha sido posteriormente extendida al concepto de "esfera equivalente" para poder manejar varios campos de visión que se extraen de otras cámaras como las de ojo de pez, las omnidireccionales o las catadióptricas [8, 9, 10, 11].

Figura 1.4 Imagen tomada con una cámara catadióptrica central a la derecha e imagen tomada con cámara de ojo de pez a la izquierda [11]

1.1.2 MÉTODOS BASADOS EN LA EXTRACCIÓN DE LÍNEAS

Los métodos más populares para estimar la orientación de una imagen equirectangular consisten en el uso de líneas y los puntos de fuga [12, 2]. Esto es debido a que el punto de fuga, la proyección en la imagen del punto donde se cortan dos rectas paralelas, contiene la información de la dirección de las rectas [13]. Obtener la dirección del punto de fuga nos da una indicación de la dirección de la vertical (coincidente con la gravedad) en la imagen.

Mientras algunos métodos obtienen cada punto de fuga independientemente [14, 15], se ha demostrado que imponer restricciones estructurales en las estimaciones de los puntos de fuga da lugar a unos resultados más robustos, por ejemplo utilizando las suposiciones basadas en "Manhattan" [12, 1, 16, 17] o "Atlanta world" [18, 2]. Sin embargo estos métodos funcionan solo en ambientes urbanos construidos por el hombre, que destacan por estar compuestos de un gran número de líneas rectas y paralelas.

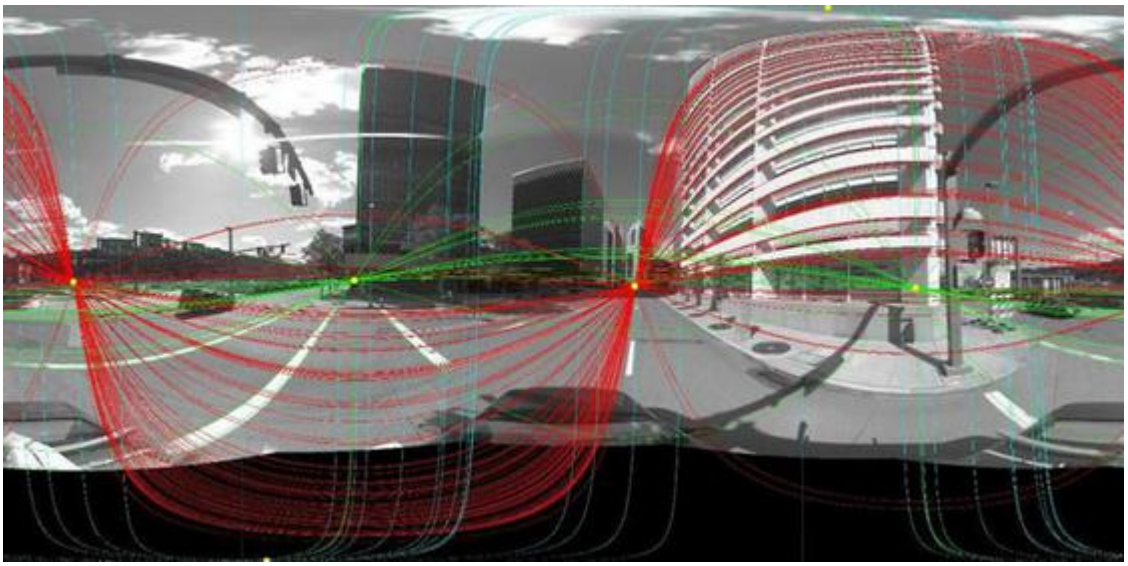


Figura 1.5 Ejemplo representativo de la extracción de líneas y punto de fuga por el método RANSAC [8]

1.1.3 MÉTODOS BASADOS EN EL HORIZONTE

La estimación del horizonte da lugar a otra popular categoría de trabajo [3, 4], donde la idea principal consiste en detectar la línea del horizonte. En la representación esférica, el horizonte es un plano que pasa por el centro de la esfera. Estos métodos aprovechan las diferencias de color que existen entre el cielo y la tierra para maximizar un criterio de separación, expresado por ejemplo como la distancia de Mahalanobis.

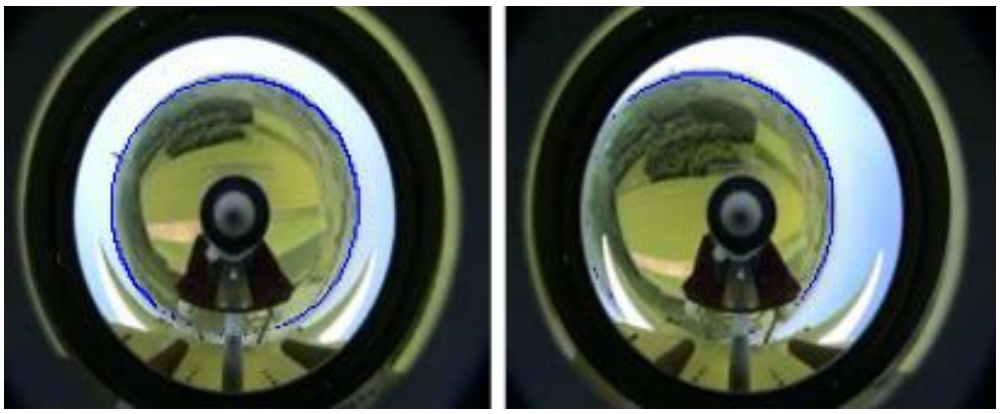


Figura 1.6 Estimación del horizonte en una imagen en perspectiva catadióptrica [4]

Este método sin embargo, no obtiene resultados satisfactorios cuando la línea del horizonte no se muestra de manera explícita o directamente no es observable como puede ocurrir en imágenes de interior.

1.1.4 ESTIMACIÓN DEL MOVIMIENTO DE LA CÁMARA

Existen un gran número de métodos que han sido propuestos para la estimación del movimiento de la cámara procedente de las imágenes omnidireccionales [19, 20, 21, 22, 23, 24, 25]. Estos métodos nos ofrecen unos resultados muy buenos sobre el movimiento y la reconstrucción 3D, pero las soluciones que nos ofrecen son relativas a una referencia elegida previamente, esta arbitrariedad es un problema para rectificar la orientación de este tipo de imágenes ya que lo que necesitamos es conocer la orientación respecto a una referencia absoluta.

De manera similar, métodos de estabilización de vídeos gran angular u omnidireccionales, pueden generar una versión refinada de un video tembloroso como entrada, pero no pueden obtener la orientación vertical absoluta que se necesita para la rectificación que estamos buscando.

1.1.5 TÉCNICAS DEEP LEARNING

Motivados por los grandes avances y logros obtenidos de la mano del deep learning, se han empezado a estudiar recientemente métodos que se basan en esta técnica para estimar la orientación de las imágenes y su rectificación. En el contexto de las imágenes con perspectiva tradicional, métodos recientes han entrenado redes neuronales con el fin de estimar la rotación en el plano de la foto [26, 27], o la rotación completa (orientación) [28, 29, 17].

Figura 1.7 Estimación de la línea del horizonte mediante Deep learning [28]

Sin embargo, estos métodos están dedicados a las perspectivas clásicas de las imágenes que todos conocemos. Haciendo falta una extensión a imágenes omnidireccionales.

1.2 OBJETIVOS

El objetivo principal de este trabajo consiste en lograr un método universal para estimar tanto la línea del horizonte como los puntos de fuga verticales característicos de cualquier foto en formato equirectangular, tanto para escenas de interior como para entornos de exterior independientemente si nos encontramos en una escena de carácter más urbano o con un paisaje más natural, donde hoy en día tenemos más limitaciones para extraer ambas características.

Para conseguirlo, se propone un método basado en el aprendizaje profundo de una red neuronal convolucional, la cual será entrenada con todo tipo de fotos equirectangulares para inferir la línea del horizonte y los puntos de fuga verticales a partir de elementos geométricos de la imagen pero también del contexto.

Como segundo objetivo del trabajo, vamos a desarrollar un método para poder estabilizar imágenes panorámicas estimando su dirección vertical y orientándolas de manera horizontal cuando sea necesario. Esta estabilización vertical se realiza a partir de la información que nos aporta la salida de la red que vamos a entrenar.

Por último, se va a desarrollar un método basado en geometría y visión por ordenador con el objetivo de comparar las prestaciones de nuestro método con los métodos actuales, buscando las limitaciones de ambos métodos.

CAPÍTULO 2

PANORÁMAS ESFÉRICOS

Como se ha comentado anteriormente, una manera popular y conveniente de representar y procesar imágenes para realidad virtual es la proyección esférica equirectangular.

Un panorama esférico es una fotografía que captura un campo de vista de 360º en horizontal y 180º o en vertical. Se trata de una imagen que almacena la proyección del mundo 3D en una esfera cuyo centro coincide con la posición del observador.

Esta representación se utiliza a menudo para que el usuario pueda sumergirse en un entorno. Quizá, el ejemplo más conocido en el que se utilizan este tipo de fotos sea el de la aplicación Google Street View que permite a un usuario conocer a pie de calle ciudades de todo el mundo.

Un panorama esférico, queda definido por una matriz de píxeles cuyas filas y columnas son proporcionales a dos de las tres coordenadas esféricas de los puntos 3Ds de la escena (los ángulos theta y phi) y representa la proyección de la escena en una esfera de radio unidad. En general un punto 3D puede describirse en coordenadas esféricas mediante tres parámetros:

$$(r, \phi, \theta)$$

Generalmente se definen como:

- La distancia r es la distancia de un punto al centro de la esfera (el radio).

- El ángulo ϕ es el acimut (o ángulo acimutal): el ángulo que nos da una idea de la posición del punto en el sentido horizontal. Se mide de -180º a 180º girando respecto del eje +Z.

- El ángulo θ es el ángulo polar (también llamado ángulo cenital o colatitud): el ángulo que nos da una idea de la posición en el sentido vertical. Se mide de -90º a 90º girando respecto del eje +X.

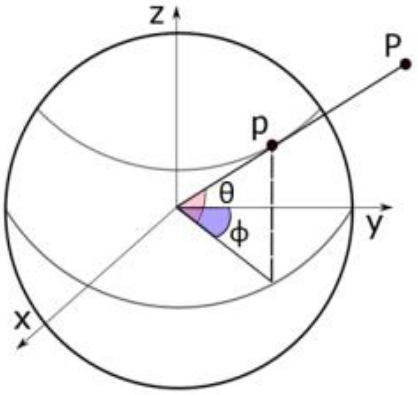


Figura 2.1 Representación de los parámetros de la representación esférica

Por ejemplo:

- (1, 0º, 0º), se corresponde con el eje +Y.
- (1, 0º, 90º), se corresponde con el eje +Z.
- (1, 90º, 0º), se corresponde con el eje +X.

Los panoramas esféricos cada vez son más habituales, sin embargo, la complejidad del tipo de proyección que emplean hace que no sea tan intuitivo o directo comprender las proporciones o la distribución de las escenas que estas imágenes muestran. Esto es debido a que, tratándose de una proyección esférica, lo que veríamos como una línea recta en la realidad o en una imagen convencional, aparece como una línea curva en la imagen panorámica. Además, el ser humano no es capaz de ver lo que hay detrás de sí mismo y sin embargo sí es posible con estas imágenes, lo cual muchas veces resulta extraño para nuestro cerebro.

Para ilustrarlo con un ejemplo, vamos a considerar una imagen panorámica en la que se pueda ver claramente el horizonte. Al pasar esta imagen a una proyección esférica, el horizonte se corresponde con un círculo que es el resultado de la intersección de la esfera con un plano que pasa por el centro de la esfera. El vector normal a este plano representa la orientación del plano del horizonte y a su vez la orientación de la cámara.

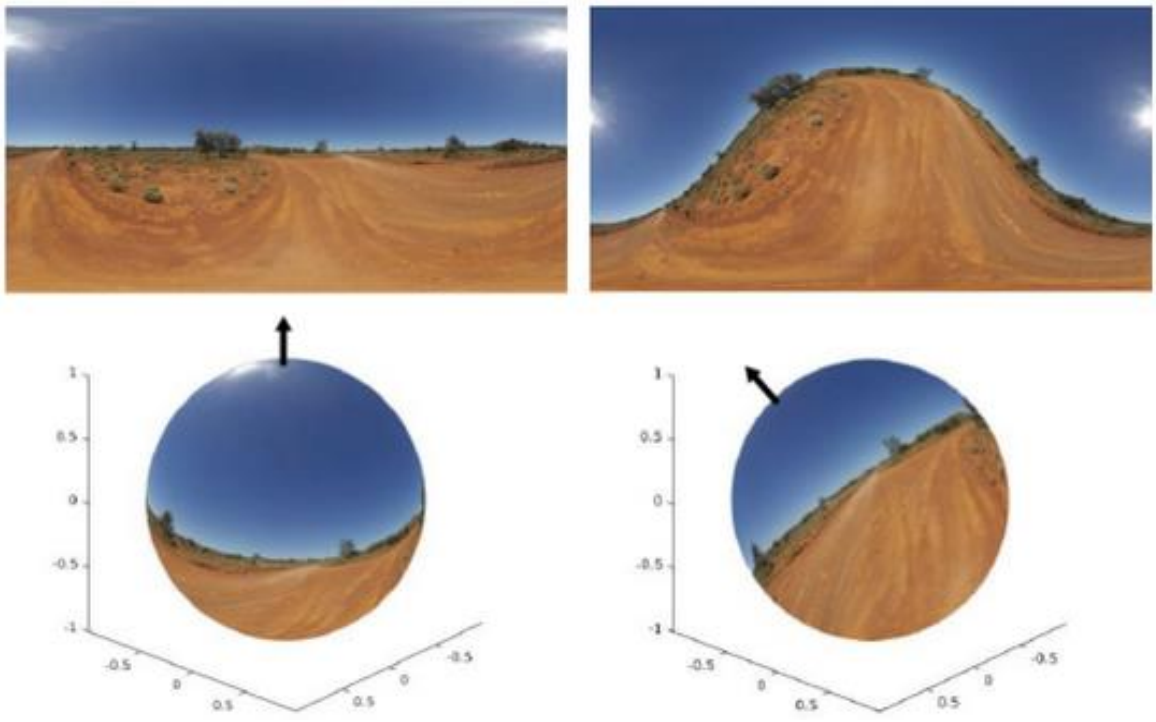


Figura 2.2 Representación equirectangular y esférica: orientación horizontal (izquierda) y orientación cualquiera (derecha).

De manera general, en la representación esférica de la imagen, la orientación de la cámara puede ser representada por un vector, al cual llamaremos el vector vertical y que coincide con el vector normal al plano del horizonte. Vamos a considerar que cuando este vector tenga la misma dirección que la gravedad pero en el sentido contrario a la fuerza que ejerce, la orientación de la cámara será horizontal y el vector vertical será v= (0, 0, 1). Por su parte, cuando la cámara no esté en orientación horizontal, el vector vertical será otro en función del giro respecto a la correcta orientación, v= (vx, vy, vz).

CAPÍTULO 3

TÉCNICAS DEEP LEARNING

Las redes neuronales convolucionales han sido aplicadas satisfactoriamente a una extensa variedad de tareas como pueden ser la detección de objetos, clasificación de escenas o la segmentación semántica. En este trabajo hemos entrenado una red neuronal profunda "end-to-end" adaptada a imágenes omnidireccionales con el objetivo de obtener una representación gráfica de la predicción de la línea del horizonte de la imagen por un lado y del punto de fuga vertical de la imagen por el otro.

Las FCNs (Fully Convolutional Networks) convierten cada capa totalmente conectada en una capa convolucional con un kernel cubriendo enteramente la región de inputs y luego la reajusta para la tarea de etiquetado a nivel pixel. Los modelos FCN son muy adecuados para tareas que requieren información contextual de la imagen completa.

3.1 ARQUITECTURA DE LA RED

La red que hemos entrenado está basada en la arquitectura de CFL [35], donde proponen una FCN para detectar esquinas y bordes estructurales en imágenes de interior. Esta FCN sigue la estructura del encoder-decoder cuyas primeras capas se basan en ResNet-50 [16]. En este caso la capa final totalmente conectada es reemplazada por un codificador-decodificador que predice conjuntamente las ubicaciones de la línea del horizonte y los puntos de fuga verticales.

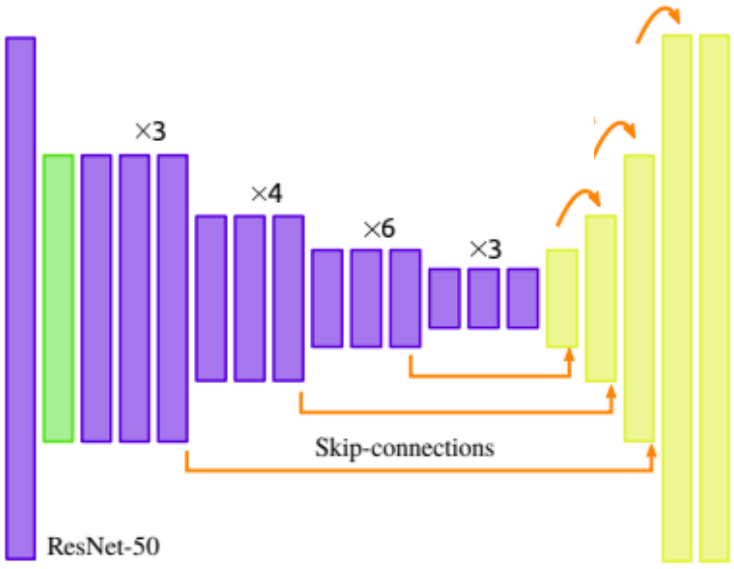


Figura 3.1 Imagen representativa de la arquitectura de la red.

- **Encoder**: Para diseñar el encoder se utiliza ResNet-50 [16], la cual ha sido previamente entrenada con el conjunto de datos ImageNet [26]. Este entrenamiento previo da lugar a una convergencia más rápida gracias a las características generales de bajo nivel aprendidas de ImageNet. Las redes residuales permiten aumentar la profundidad sin aumentar el número de parámetros. Esto lleva, en ResNet-50, a capturar un campo receptivo de 483 × 483 píxeles, suficiente para la resolución de entrada que vamos utilizar de 256 × 128 píxeles. El encoder incluye además una serie de capas convolucionales que permiten especializar la tarea.

- **Decoder:** Lo que se propone en la parte de la red que decodifica la información es una única rama con dos canales de salida que van a ser los mapas de píxeles que se estén buscando como solución, lo cual ayuda a reforzar la calidad de este tipos de mapas.

  En este decoder, se combinan dos ideas diferentes. En primer lugar skip-connections [25] del encoder al decoder. Más concretamente, se concatenan las características "up-convolved" con sus correspondientes características de la parte en decodificación. En segundo lugar, se hacen predicciones preliminares en una resolución menor, las cuales también se concatenan y retroalimentan la red, siguiendo la idea de [10], asegurando que las primeras etapas de las características internas se dirijan hacia la tarea que se busca. Se usa ReLU como función no lineal excepto para las capas de predicción, donde usamos Sigmoid.

Para la arquitectura de red propuesta, vamos a usar EquiConvs, tanto en el encoder como en el decoder. EquiConvs es una convolución que se define en el dominio esférico en lugar del dominio de la imagen y es implícitamente invariante a la distorsión que se da en la representación de imágenes equirectangulares.

Figura 3.2 Representación gráfica del ajuste del kernel según su ubicación en la escena.

La principal diferencia de la red que vamos a entrenar con CFL reside en que, al ser la tarea a realizar completamente distinta, la información de salida y el etiquetado ground truth debe ser distinto. El ground truth (GT) que vamos a tener asociado a cada imagen panorámica y por tanto la salida que se busca consiste en dos mapas de píxeles, el primero va a ser la línea del horizonte correspondiente y el segundo la localización de los puntos de fuga verticales de la foto. A partir de este GT, el error de las predicciones se reduce gradualmente a medida que la predicción se acerca al objetivo.

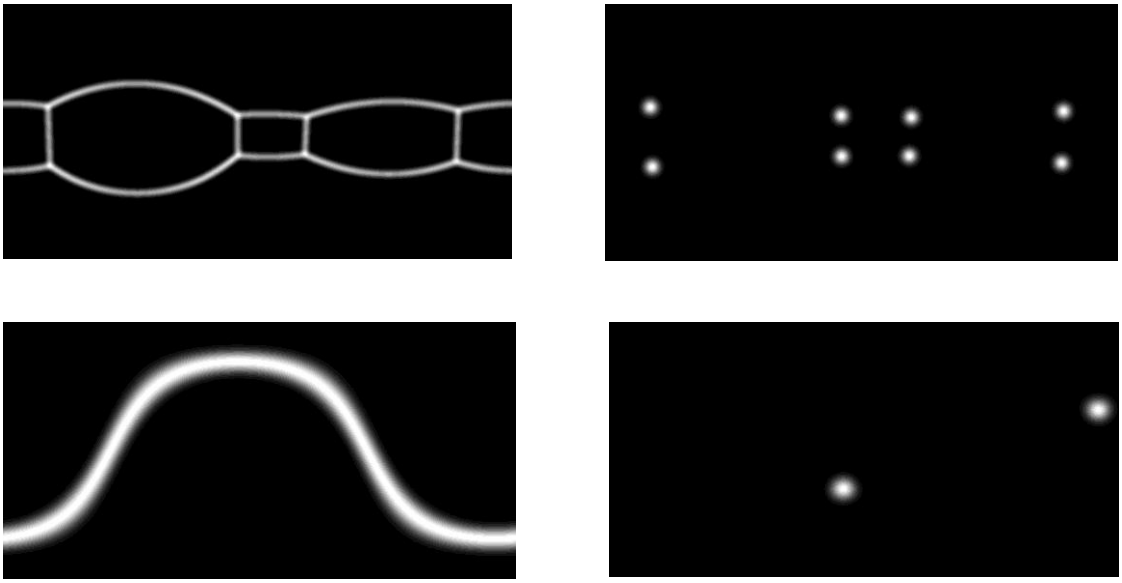


Figura 3.3 Ground truth y salida de la red CFL y ground truth y salida de nuestra red. Corner for Layouts (CFL) tiene una salida con dos canales (arriba): un canal representando contornos de elementos constructivos y otro representando esquinas. Nuestra propuesta (abajo) tiene una salida también con dos canales: uno representando la línea del horizonte (izquierda) y otro representando los puntos de fuga superior e inferior que describe la dirección vertical (derecha).

CAPÍTULO 4

GENERACIÓN DEL DATASET

Para poder entrenar la red neuronal con la que vamos a extraer la línea del horizonte y los puntos de fuga de panoramas esféricos necesitamos un numeroso grupo de imágenes equirectangulares con diferentes orientaciones, junto con su correspondiente ground truth tanto de la línea del horizonte como de los puntos de fuga, sin embargo, debido a la falta de una dataset con estas características entre los recursos con los que se cuenta, ha sido necesario crearlo de propio para este proyecto.

4.1 RECOLECCIÓN DE FOTOGRAFÍAS

Este dataset ha sido generado a partir de una gran cantidad de fotos equirectangulares obtenidas en internet, procedentes en parte de otros datasets existentes como SUN360 o F-360iSOD [30], aunque la mayoría de fotos han sido extraídas de la página web Flickr, ya que la existencia de un grupo especializado en imágenes equirectangulares [31] facilitó la búsqueda de este tipo de fotos.

Este conjunto de fotos recolectado contiene 1097 imágenes orientadas de manera vertical, con un campo de vista esférico completo, en un formato equirectangular y con una resolución de 1024x2048 píxeles en las imágenes de exterior y una resolución de 512x1024 píxeles en las imágenes de interior.

Figura 4.1 Ejemplo de fotos de exterior e interior del dataset

Las imágenes han sido capturadas por diferentes tipos de cámaras omnidireccionales, en varias localizaciones y escenas, tanto de interior como de exterior, como escenas urbanas/artificiales y naturales y con distinta iluminación (en distintos momentos del día) buscando que la red sea lo más universal posible siendo eficaz en el mayor número de imágenes panorámicas posibles.

4.2 GIRO DE LAS FOTOS OBTENIDAS

Una vez recolectadas las fotos, para generar una nueva imagen panorámica con una orientación específica lo que se hace es girarla de manera artificial. Este giro se realiza en tres pasos principales.

Primero se proyecta la foto equirectangular de entrada en una esfera, esta proyección se realiza haciendo el cambio de coordenadas de píxeles a coordenadas esféricas. Definimos la resolución de la imagen panorámica equirectangular como $W \times H$ píxeles, siendo W la anchura de la imagen y H, la altura de ésta. En un primer momento por tanto, la imagen viene definida como un mapa lineal de coordenadas (u, v), que representan la distancia en píxeles de la imagen.

De esta manera convertimos directamente u a su correspondiente ángulo azimutal de tal manera que $u \in (1, W)$, se transforma de manera lineal en $\vartheta \in (-180^o, 180^o)$ y de manera similar operamos con v convirtiéndolo a su correspondiente ángulo polar tal que $v \in (1, H)$ se transforma de manera lineal en $\varphi \in (-90^o, 90^o)$.

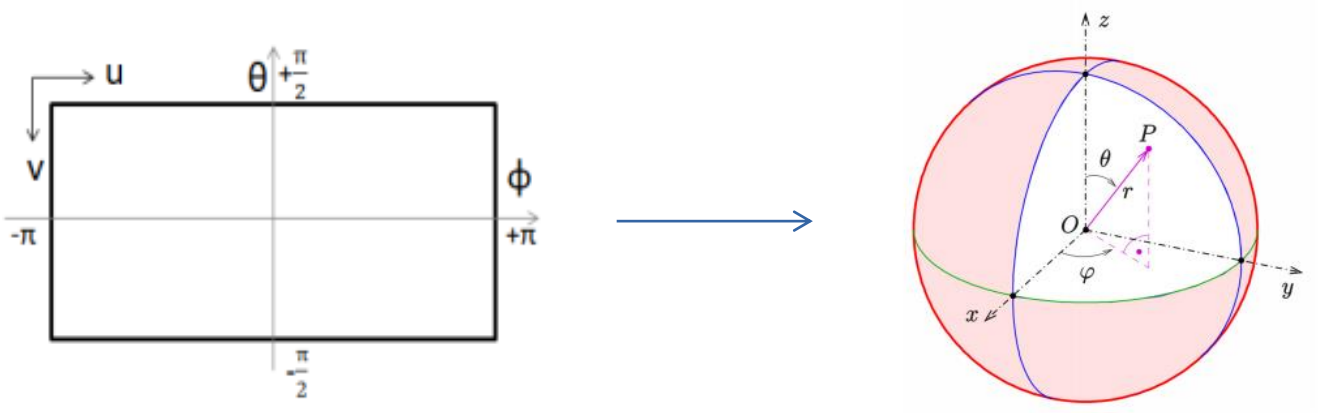


Figura 4.2 Representación esquemática del cambio de coordenadas

Una vez tenemos la imagen en coordenadas esféricas, el segundo paso es aplicarle una rotación determinada a la foto simulando una orientación no horizontal cualquiera de la cámara. Esta rotación se realiza cambiando la referencia absoluta de las coordenadas esféricas, respecto a la referencia correcta en la que el eje Z coincide con la gravedad. La nueva referencia absoluta se obtiene aplicando una rotación en el eje Z y una rotación el eje X.

RAbsCam = rotzRad (giroZ) * rotxRad (giroX)

VertVp = RAbsCam * [0; 0; 1];

vertVpDown = RAbsCam * [0; 0; -1];

Calculada la nueva referencia de la cámara se aplica el giro a la imagen en coordenadas esféricas.

Por último como tercer paso quedaría deshacer el cambio de coordenadas pasando, de manera inversa al primer paso, de coordenadas esféricas a coordenadas de distancia en píxeles.

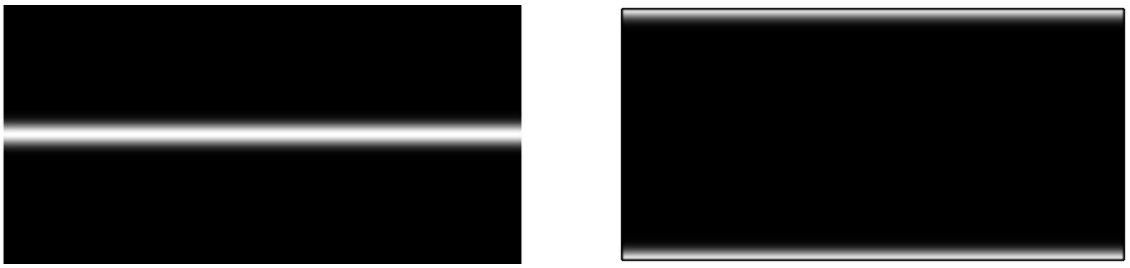
4.3 GENERACIÓN DEL GROUND TRUTH

Los puntos de fuga son aquellos puntos en el plano imagen donde convergen las proyecciones de las líneas paralelas del mundo. Son características invariantes a escala y rotación, por lo que pueden ser utilizadas para múltiples tareas como correspondencia entre imágenes, calibración de la cámara o reconocimiento de objetos. Si estas rectas paralelas siguen la dirección de la gravedad normalmente identificaremos los puntos de fuga asociados como puntos de fuga verticales.

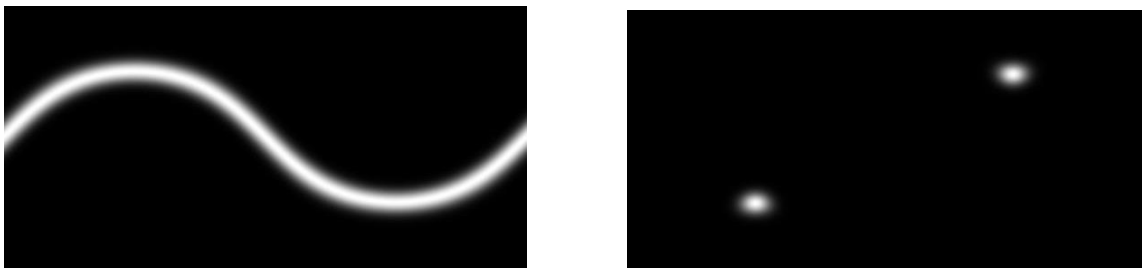
Por su parte el horizonte es la línea que aparentemente separa el cielo y la tierra. Esta línea es en realidad una circunferencia en la superficie de la Tierra centrada en el observador, en nuestro caso y para la representación esférica de la foto panorámica, la línea del horizonte se corresponde con un círculo que es resultado de la intersección de la esfera y un plano que pasa por el centro de la misma. Nótese que, excepto por irregularidades del relieve, el plano que contiene a la línea del horizonte es perpendicular a la dirección de la gravedad.

El ground truth del dataset se corresponde con un mapa de píxeles de la línea del horizonte y otro de los puntos de fuga verticales asociados a cada foto girada artificialmente tal y como hemos explicado en la sección anterior. Ambos mapas de píxeles son definidos de tal forma que en un principio sus valores van a ser 1 si el pixel pertenece a la línea del horizonte o al punto de fuga vertical y 0 en el caso contrario. Posteriormente se hace un engrosamiento de líneas y desenfoque gaussiano para facilitar la convergencia durante el entrenamiento, ya que hace que la evaluación del error sea continua en lugar de binaria.

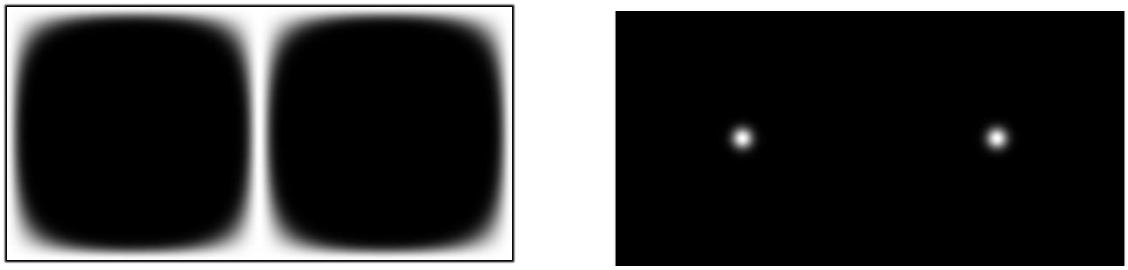
Estas dos fotos se generan al mismo tiempo que se realiza el giro de las fotos, ya que el resultado de la generación del ground truth depende únicamente de la rotación que se le aplica a cada foto y de la resolución de la foto de entrada, siendo independiente de la foto en sí.



a)



b)

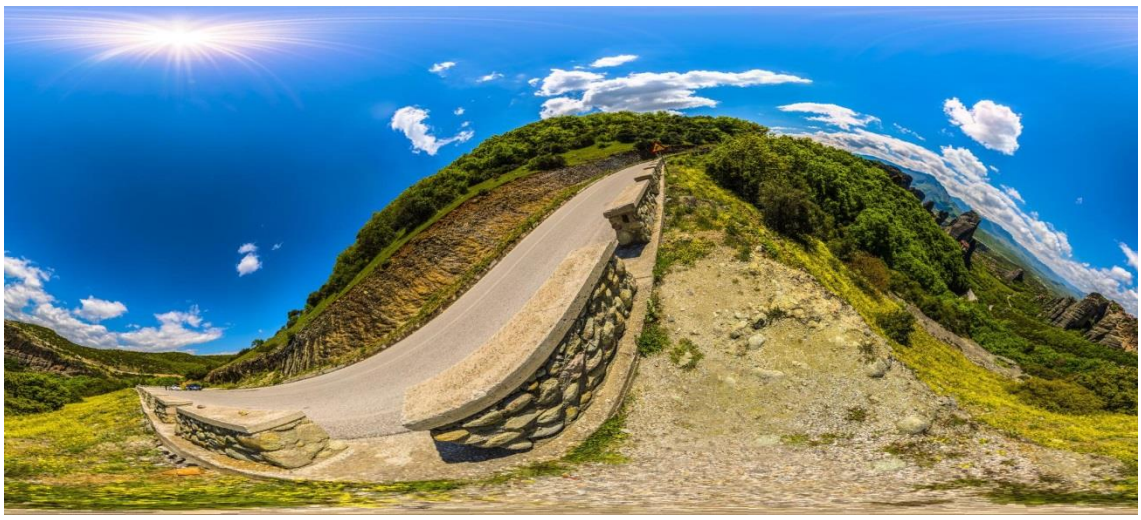


c)

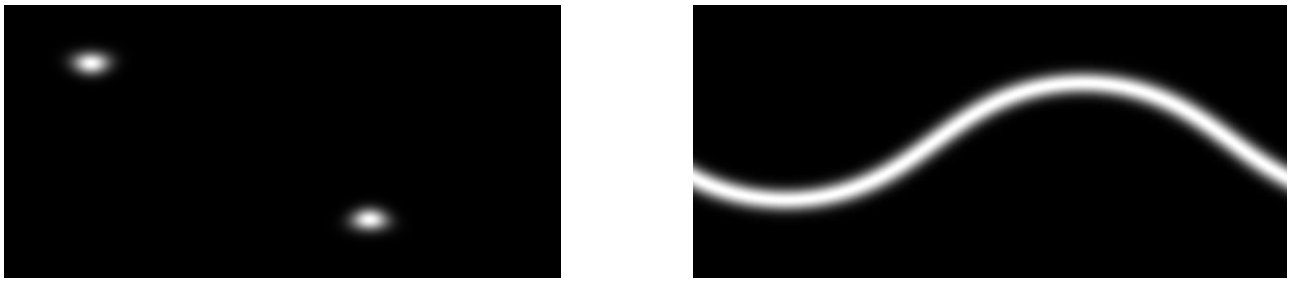
Figura 4.3 Ejemplos de línea del horizonte (derecha) y puntos de fuga (izquierda) generados. a) Ground truth asociado a una foto orientada de manera horizontal; b) Ground truth asociado a una foto girada 45º sobre el eje X; c) Ground truth asociado a una foto girada 90º sobre el eje X

4.4 EJEMPLOS VISUALES

Figura 4.4 Imagen original extraída de la página web Flickr [6]



a)



b)

Figura 4.5 a) Imagen girada artificialmente para la generación del dataset; b) Ground truth de la línea del horizonte y los puntos de fuga de la foto girada

a)

b)

Figura 4.6 a) Imagen girada artificialmente para la generación del dataset; b) Ground truth de la línea del horizonte y los puntos de fuga de la foto girada

Con el objetivo de aprovechar al máximo las fotos que hemos conseguido de internet y engrosar al máximo el tamaño del dataset para que el entrenamiento de la red resulte lo más fructífero posible dentro de un tiempo razonable, cada foto panorámica es girada cinco veces, las rotaciones aplicadas para realizar los giros se hacen de manera aleatoria, siendo el posible ángulo de rotación sobre el eje Z un ángulo comprendido entre –Pi y Pi, mientras que el posible ángulo de rotación sobre el eje X está comprendido entre –Pi/2 y Pi/2.

Finalmente concluimos el proceso con 5485 fotos giradas aleatoriamente con sus respectivas fotos de línea del horizonte y puntos de fuga verticales asociadas. Estas fotos se van a agrupar en tres datasets diferentes según su función en el entrenamiento:

- **Dataset de entrenamiento**: Compuesto por 4375 fotos destinadas a realizar el entrenamiento de la red.

- **Dataset de test**: Compuesto por 550 fotos destinadas a evaluar los resultados obtenidos de la red.

- **Dataset de validación**: Compuesto por 560 fotos destinadas para dar apoyo durante el entrenamiento de la red, como paso de comprobación intermedio durante el entrenamiento.

CAPÍTULO 5

DESHACER GIRO

El uso que se le da a la línea del horizonte y los puntos de fuga verticales calculados, como se ha comentado en los objetivos de este trabajo, es poder estimar la dirección vertical en la referencia de la cámara y rectificar las imágenes panorámicas para que queden en orientación horizontal. A través de estos elementos extraídos con la red neuronal podemos calcular los ángulos de giro que se ha desviado la fotografía respecto a la dirección vertical (coincidente con la gravedad) y deshacer ese giro para obtener la foto correctamente ajustada.

En nuestro caso, para este trabajo, vamos a trabajar con los mapas de píxeles de los puntos de fuga verticales buscando los ángulos de giro. Para ello, lo que vamos a hacer, es obtener en que coordenadas se encuentran los puntos de fuga en la imagen. Esta tarea de procesado de imagen la vamos a llevar a cabo con Matlab.

En primer lugar, se detectan los píxeles cuya iluminación es de 255, ya que estos píxeles representan el centro de ambos puntos de fuga. Lo segundo que se hace, es separar estos píxeles en grupos según si representan el punto de fuga vertical positivo o su punto de fuga antipodal (diametralmente opuesto). Una vez tenemos los dos grupos de píxeles, calculamos las coordenadas de su centro de gravedad por separado.

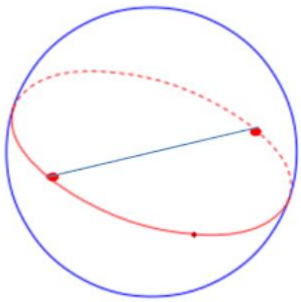


Figura 5.1 Puntos antipodales en una esfera

Las coordenadas en píxeles (u, v) en la que se encuentran los centros de gravedad calculados anteriormente la consideraremos la posición de los puntos de fuga verticales. Como hemos visto en el capítulo 4, si conocemos las coordenadas del punto de fuga en coordenadas en píxeles podemos conocer de manera directa sus coordenadas esféricas, a partir de las cuales podemos conocer cuál ha sido el giro realizado y corregirlo.

A partir de las coordenadas esféricas del punto de fuga vertical, también podemos calcular de manera directa cual es el vector vertical de la foto. A partir de este vector se puede corregir la orientación aplicando la rotación necesaria para llevar nuestro vector vertical calculado (vx, vy, vz) a la posición (0, 0, 1). El cálculo del vector vertical explicado en este apartado se usará en los siguientes apartados a la hora de evaluar las prestaciones de la red, ya que aparte de poder rectificar la foto a partir de él, nos sirve para calcular la desviación que presenta nuestro cálculo del punto de fuga vertical, respecto al punto de fuga vertical del ground truth conocido de antemano.

CAPÍTULO 6

EXPERIMENTACION

6.1 RESULTADOS DEL ENTRENAMIENTO

Para obtener los distintos resultados con los que evaluar las prestaciones de la red vamos utilizar la red con el dataset de test mencionado en el apartado 4. Más concretamente contamos en este dataset para evaluar la red con 260 fotos de interior (512x1024 píxeles) y 290 fotos de exterior (1024x2048 píxeles). Aunque la resolución de descarga de las fotos sea diferente, el tamaño de la entrada y salida de la red es común para todas las fotos en los experimentos realizados, tanto en la inferencia de los resultados de test, como en la comparativa de estos resultados con el ground truth. La inferencia para la evaluación de la red mediante el dataset de test se hace procesando los resultados con la CPU, procedimiento más lento para procesar imágenes que mediante GPU. El entrenamiento se realizó procesando la red mediante GPU con un tiempo aproximado de entrenamiento de 7 horas.

En esta sección evaluamos la salida de la red, calculando a nivel de píxeles, las distintas métricas que nos ayudan a saber cómo de buenos son los resultados obtenidos. Estas métricas se calculan teniendo en cuenta la coincidencia de píxeles entre el Ground Truth y su correspondiente imagen extraída.

Para obtener la coincidencia entre ambos mapas de píxeles se comparan las imágenes con el operador lógico &, que devuelve el valor booleano true si ambos operandos son true, es decir, devuelve el valor “1” en todos aquellos píxeles que estén en blanco en las dos imágenes que se comparan.

En las imágenes de salida, debido al desenfoque gaussiano empleado en la generación del ground truth, las figuras de la línea del horizonte y los puntos de fuga van perdiendo iluminación progresivamente. Considerando estas figuras como funciones de densidad probabilística vamos a imponer un nivel de confianza a partir del cual decidiremos si nuestro píxel pertenece o no a la figura según su nivel de iluminación.

Normalizando el nivel de iluminación entre 0 y 1, elegimos un umbral de 0,1, lo que significa que los píxeles que se van a considerar como 1 para hacer la comparativa booleana deben estar iluminados al menos un 10%. Este umbral en la función de densidad de probabilidad Gaussiana normalizada corresponde con un intervalo de confianza del 98.41% o lo que es lo mismo, 2.146 veces sigma.

De esta manera ya podemos comparar el ground truth con nuestros resultados de manera binaria, lo que nos permite dividir los píxeles según cuatro tipos de resultados posibles:

- True Positives (TP): Píxeles correctamente iluminados.
- True Negatives (TN): Píxeles correctamente no iluminados.
- False Positives (FP): Píxeles iluminados que no deberían estarlo.

- False Negatives (FN): Píxeles no iluminados que deberían estarlo.

A partir de esta catalogación de los píxeles calculamos las métricas que nos relacionan ambos conjuntos (ground truth y predicción)

- **IoU (Intersection over Union):** Representa el área de superposición entre la segmentación predicha y ground truth dividida por el área de unión entre la segmentación predicha y el ground truth.

$$IoU = \frac{TP}{TP + FP + FN}$$

- **Pixel Accuracy:** Es el porcentaje de píxeles de una imagen que están clasificados correctamente. Se calcula mediante la relación entre los resultados bien clasificados (comparando con el ground truth) y el total de píxeles.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Pixel Precision:** Describe la pureza de nuestras detecciones positivas en relación con el ground truth. Se calcula mediante la relación entre los resultados positivos que coinciden con el ground truth y el total de positivos que se ha calculado.

$$Precision = \frac{TP}{TP + FP}$$

- **Pixel Recall:** Describe la integridad de nuestras predicciones positivas en relación con el ground truth. Se calcula mediante la relación entre los resultados positivos bien clasificados y el total de resultados que deberían ser positivos.

$$Recall = \frac{TP}{TP + FN}$$

- **F1 score:** El valor F1 se utiliza para combinar las medidas de precision y recall en un sólo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la exhaustividad entre varias soluciones

$$F1 = 2 * \frac{Precision * Recall}{Precisión + Recall}$$

Operando con ambos tipos de fotos que tenemos (550 fotos), el tiempo de cálculo ha sido de 475 segundos y los resultados obtenidos han sido:

	IoU	Accuracy	Precision	Recall	F1
Línea del horizonte	0,689	0,925	0,981	0,696	0,814
Puntos de fuga	0,495	0,973	0,969	0,5	0,66

Tabla 6.1 Resultados del test para fotos de interior y exterior

Operando con las fotos de interior el tiempo de cálculo ha sido de 212 segundos y los resultados obtenidos han sido:

	IoU	Accuracy	Precision	Recall	F1
Línea del horizonte	0,702	0,927	0,999	0,703	0,825
Puntos de fuga	0,505	0,976	0,996	0,505	0,67

Tabla 6.2 Resultados del test para fotos de interior

Operando con las fotos de exterior el tiempo de cálculo ha sido de 275 segundos y los resultados obtenidos han sido:

	IoU	Accuracy	Precision	Recall	F1
Línea del horizonte	0,678	0,922	0,965	0,689	0,804
Puntos de fuga	0,487	0,971	0,945	0,459	0,619

Tabla 6.3 Resultados del test para fotos de exterior

Se puede observar como el entrenamiento de la red ha dado unos buenos resultados. Tanto la precisión como la exactitud (accuracy) están cercanas al 1, lo que significa que la salida de nuestra red es muy cercana al groun truth de las imágenes evaluadas, acertando a la hora de detectar tanto la línea del horizonte como los puntos de fuga verticales de la escena.

Por su parte, el IoU y la exhaustividad (recall), son muy parecidos en todos los casos, pero no están tan cercanos a la unidad. Esto se debe a que las imágenes de salida de la red tienen bastantes píxeles considerados como falsos negativos, pero casi no cuentan con falsos positivos. Esto no resulta un gran problema ya que con el escaso número de falsos positivos y junto con las otras métricas calculadas podemos asegurar que tanto la línea del horizonte como los puntos de fuga verticales se encuentran bien representados en nuestros resultados.

6.2 RESULTADOS DE LA CORRECCIÓN DEL GIRO

Para evaluar la eficacia del giro se han utilizado los resultados de los mapas de píxeles de los puntos de fuga verticales extraídos tras la evaluación del dataset de test mencionado en el capítulo 4. En los cuales es conocido el giro realizado previamente ya que es un dato que nos guardamos durante la generación del dataset.
En primer lugar vamos a comparar los ángulos de giro calculados a partir de los puntos de fuga para la rectificación de la foto, como hemos explicado en el capítulo 5, con los ángulos exactos de giros realizados, para comprobar en grados como de exacto es nuestro método.
Los resultados obtenidos son:

- Desviación media respecto al giro en X = 0,5º
- Desviación media respecto al giro en Z = 1,8º

Una vez obtenidos los datos numéricos, y con el objetivo de tener una referencia que nos pueda indicar cómo de buenos son nuestros resultados, vamos a fijarnos en un estudio realizado a 14 personas en el que se les mostraba una serie de imágenes rotadas artificialmente en un dispositivo Oculus Rift de realidad virtual con el objetivo de conocer la percepción que tienen los humanos respecto a las imágenes VR y su orientación [33].

Los participantes fueron preguntados por su nivel de conformidad con la afirmación "Estoy satisfecho con la orientación de la imagen mostrada", las respuestas estaban acotadas en una escala del 1 al 5 donde cada puntuación significaba: 1, fuertemente en desacuerdo; 2, en desacuerdo; 3, neutral; 4, de acuerdo; 5, fuertemente de acuerdo.

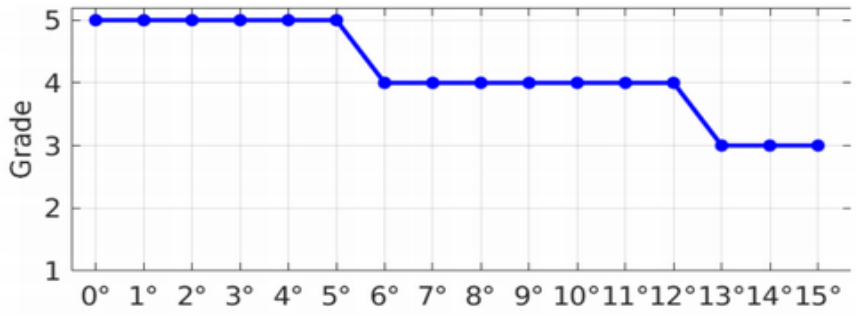


Figura 6.1 Percepción de los humanos respecto a la orientación en imágenes panorámicas [33]

La gráfica muestra como la desviación media para la cual los participantes consideraban que estaban fuertemente de acuerdo con la orientación de la imagen es de 5^o, y de acuerdo, con un error en la orientación de hasta 12^o.

Con esta nueva referencia para evaluar nuestros datos volvemos a procesar los resultados de nuestra red, esta vez diferenciando entre imágenes de interior e imágenes de exterior. Para que nuestros datos puedan compararse directamente con los resultados del estudio que acabamos de explicar, vamos a evaluar el error de nuestros resultados calculando la desviación entre el vector vertical calculado a partir de los puntos de fuga verticales de salida de la red y el vector vertical del ground truth que ya conocemos. Esta desviación se calcula con el arco coseno del producto escalar entre ambos vectores.

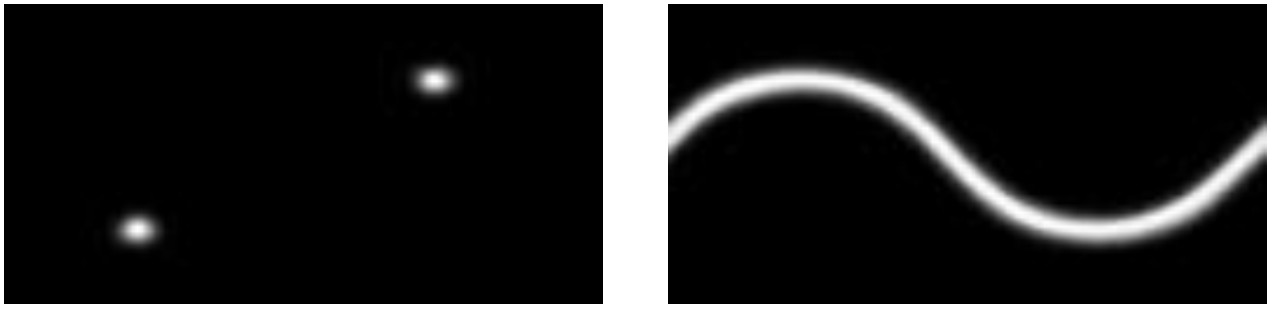
En las fotos de **interior**, considerando un error angular máximo de 5^o entre ambos vectores verticales, el algoritmo **acierta un 99,23%** de las veces, con un **error angular medio de 0,47º y una mediana del error de 0,15º.**

En las fotos de **exterior**, considerando un error angular máximo de 5^o, el algoritmo **aciertan un 93,5% de las veces,** con un **error angular medio de 1,78º y una mediana del error de 0,33º**.

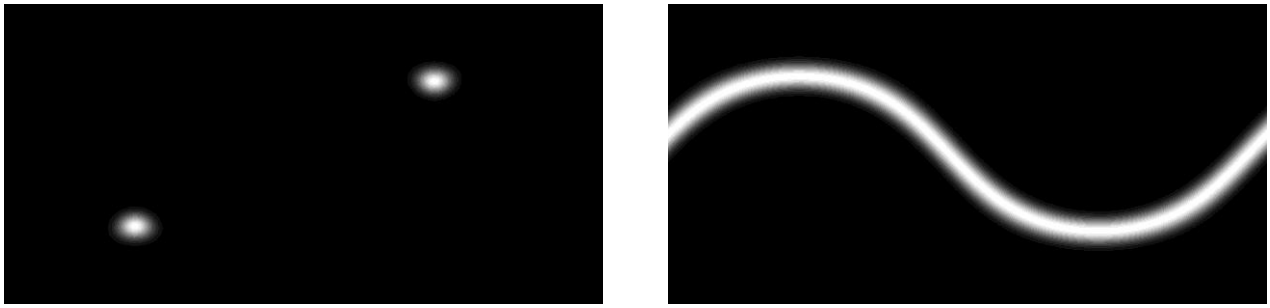
6.3 RESULTADOS VISUALES

6.3.1 ENTORNOS DE INTERIOR

a)



b)

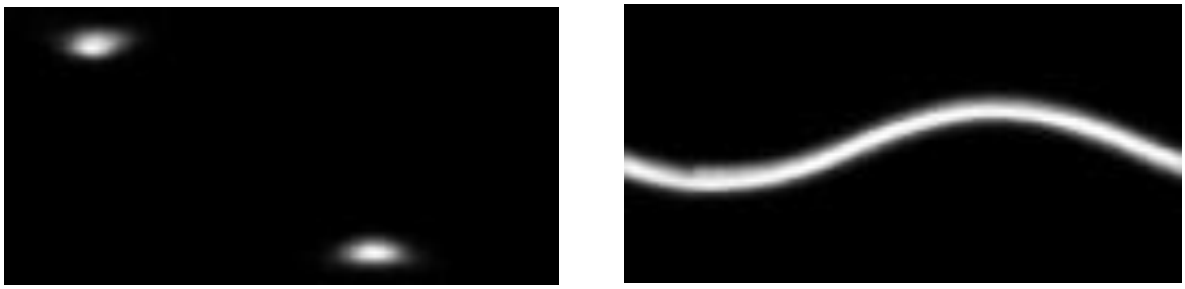


c)

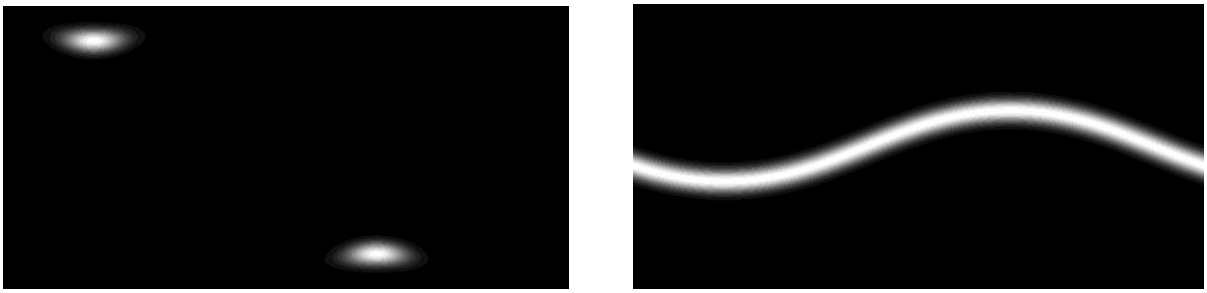
Figura 6.2 a) Imagen de interior girada artificialmente; b) Mapa de píxeles de los puntos de fuga y línea del horizonte calculados por la red; c)Ground truth de los puntos de fuga y la línea del horizonte de la foto girada

6.3.2 ENTORNOS DE EXTERIOR URBANOS

a)



b)

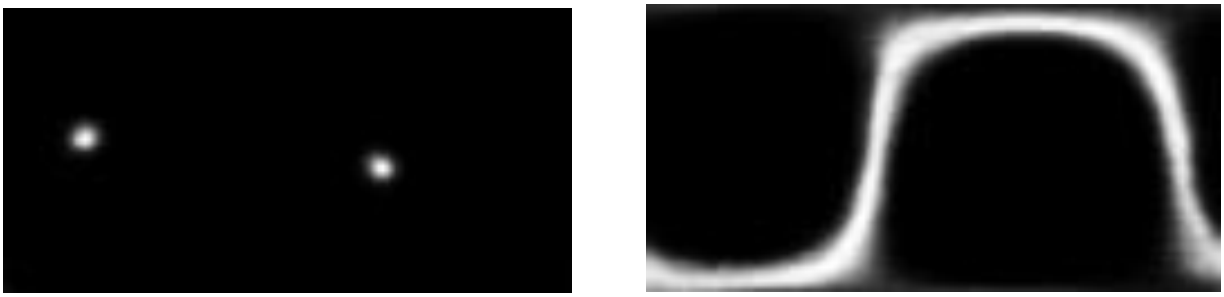


c)

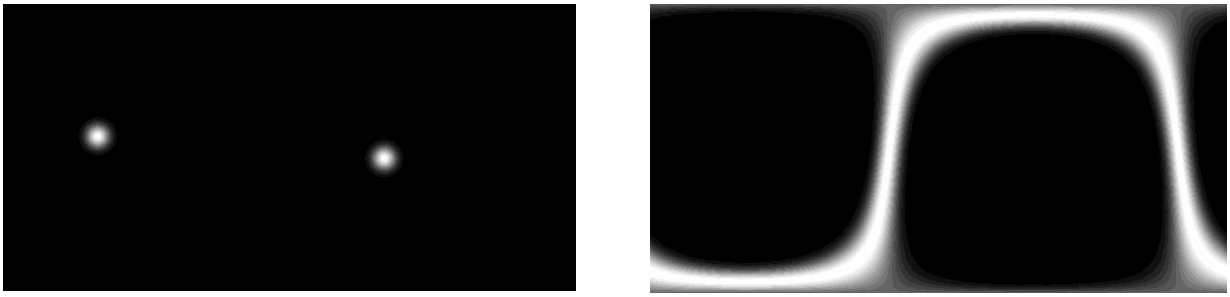
Figura 6.3 a) Imagen de exterior urbano girada artificialmente; b) Mapa de píxeles de los puntos de fuga y línea del horizonte calculados por la red; c)Ground truth de los puntos de fuga y la línea del horizonte de la foto girada

6.3.3 ENTORNOS DE EXTERIOR NATURAL

a)



b)



c)

Figura 6.4 a) Imagen de exterior natural girada artificialmente; b) Mapa de píxeles de los puntos de fuga y línea del horizonte calculados por la red; c)Ground truth de los puntos de fuga y la línea del horizonte de la foto girada

6.4 LIMITACIONES

De entre todos los tipos de fotos usados para evaluar la red, las únicas fotos que han dado problemas son las fotos de exterior en entornos naturales en las que la diferencia entre la superficie, el cielo y otros elementos de la escena no queda bien definida en la información que presenta la foto, como ocurre en cuevas o grutas.

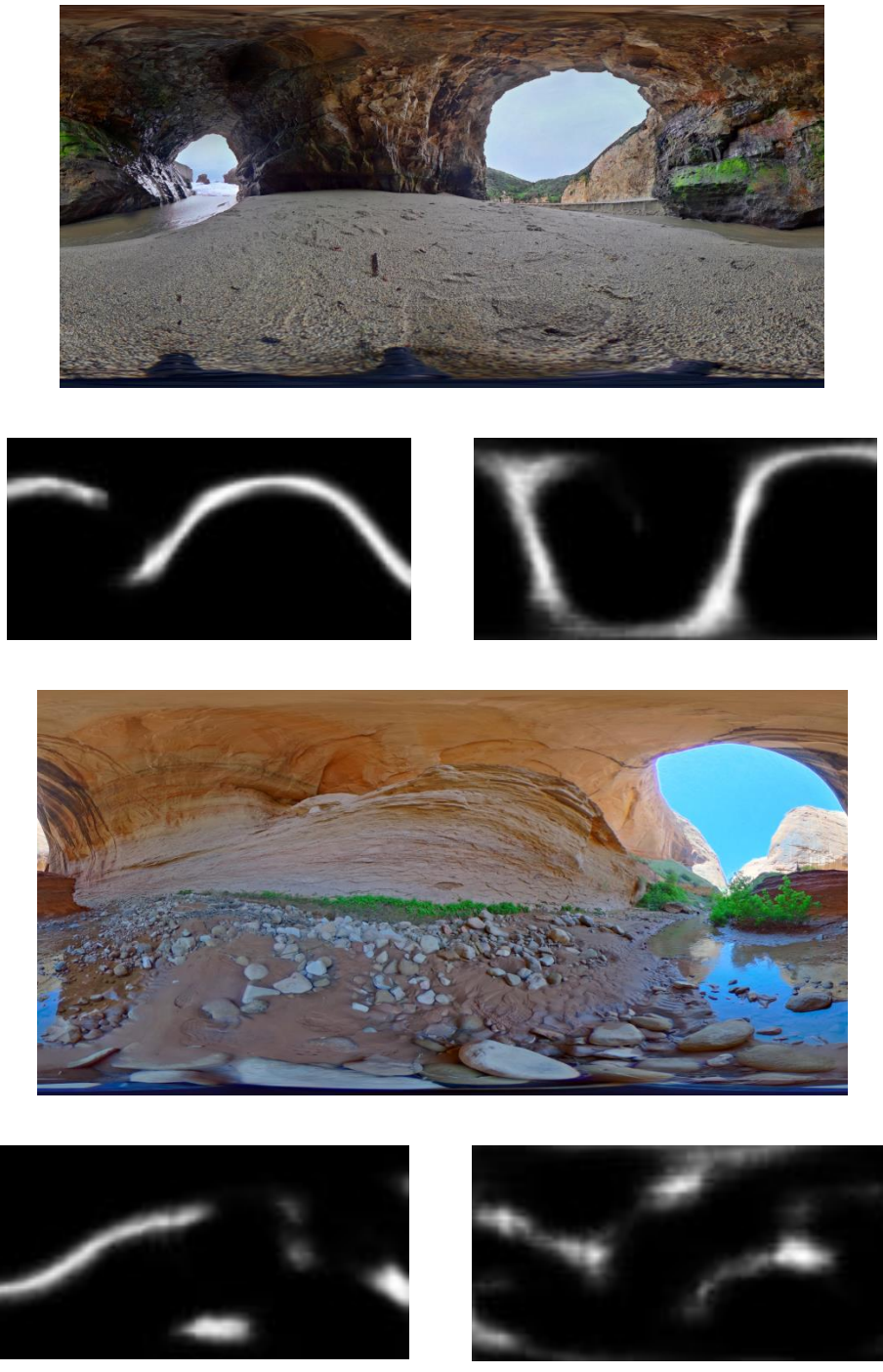


Figura 6.5 Ejemplos de fotos que dan problemas a la red y sus líneas del horizonte mal calculadas

CAPÍTULO 7

COMPARACIÓN CON MÉTODO GEOMÉTRICO

La estimación de los puntos de fuga en imágenes mediante métodos de visión artificial es un problema abordado desde hace más de una década debido a que su identificación nos permite comprender estructuras 3D a partir de características 2D.

7.1 FUNCIONAMIENTO DEL MÉTODO GEOMÉTRICO PROPUESTO

El método que vamos a emplear para poder hacer la comparativa y experimentar sobre él inicia con la extracción de líneas en la imagen panorámica mediante el procedimiento RANSAC (RANdom SAmple Consensous). Nuestro método comienza con la aplicación de un filtro Canny [32] a la imagen, el cual permite detectar los bordes existentes. De estos N bordes obtenidos eliminamos aquellos que están repetidos o cuya longitud es menor a un determinado umbral para evitar confusiones con pequeños objetos o ruido.

Figura 7.1 Contornos detectados por el filtro Canny

Al estar trabajando con imágenes panorámicas se debe tener en cuenta que una línea recta en la realidad, es proyectada como un segmento de arco sobre la esfera, y por tanto, aparece como un segmento de línea curva en la imagen. Por ello, cada segmento de arco es representado por el vector normal del plano proyectivo que incluye la propia línea y el centro de la cámara.

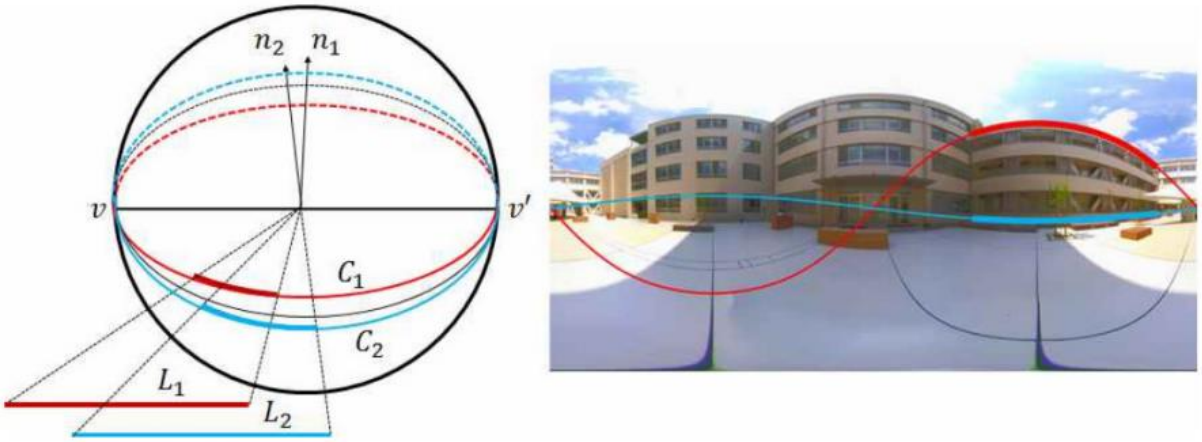


Figura 7.2 Líneas en la realidad proyectadas en la imagen esférica como arcos de un gran círculo. (Imagen de Seon Ho oh et al. [19])

Una vez obtenidos los bordes de la imagen, aplicamos nuestro algoritmo RANSAC para extracción de proyecciones de rectas. Este procedimiento elige inicialmente de manera aleatoria dos puntos de la imagen de uno de los borden extraídos para generar líneas candidatas de la imagen que son votadas por el resto de puntos del mismo borde. Para ello se computa el producto vectorial entre las direcciones de los rayos que corresponden a este par de puntos obteniendo así una posible dirección normal para este grupo de puntos. La normal obtenida se compara con el resto de rayos del grupo considerándose inliers del modelo aquellos que cumplen la condición de perpendicularidad con la normal calculada bajo un determinado treshold angular, por ejemplo 1^o, y considerándose outlayers aquellos que no la cumplen.

Este procedimiento se repite un número fijo de veces al tratarse de un algoritmo iterativo. Finalmente la iteración que haya dado lugar a un mayor número de inliers se considera el mejor modelo, obteniendo la dirección normal que mejor se ajusta a la línea. Si la línea tiene un número de inliers suficiente y su longitud es mayor a la longitud de segmento mínima establecida, conservamos dicha línea para el siguiente paso del algoritmo. En caso contrario, la línea es eliminada. Este procedimiento se aplica para cada uno de los bordes que se habían obtenido anteriormente de manera que obtenemos un conjunto de líneas candidatas de longitud considerable y con su dirección normal como información.

A partir de las líneas extraídas según el primer procedimiento RANSAC, obtendremos los puntos de fuga aplicando de nuevo un algoritmo tipo RANSAC. La información que tenemos de las líneas anteriores son las coordenadas de proyección de cada línea sobre la esfera y la dirección normal del círculo que forma la línea en dicha esfera.

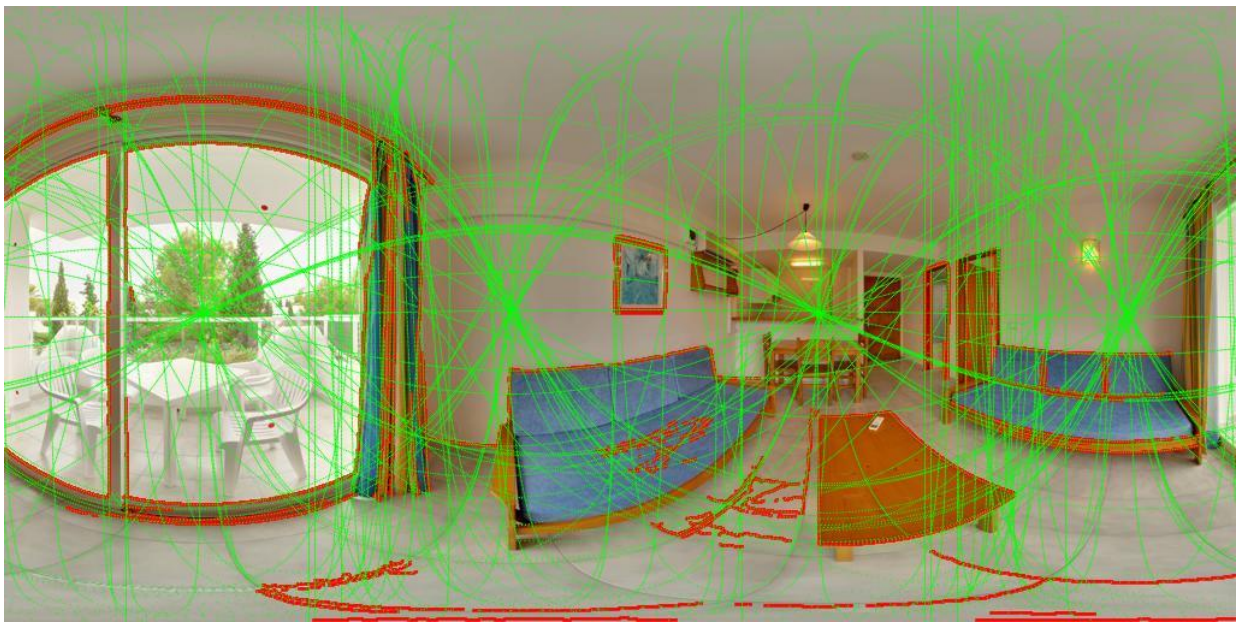


Figura 7.3 Resultado de la extracción de líneas por RANSAC

Para llevar a cabo la tarea de extracción de los puntos de fuga haremos la suposición de que existen tres puntos de fuga dominantes (Mundo Manhattan [34]), cuyas direcciones son ortogonales entre ellas y que están alineados con tres direcciones dominantes en el mundo, una por cada orientación posible de las aristas. También hay que tener en cuenta que en imágenes esféricas las proyecciones lineales dan lugar a curvas de modo que las líneas paralelas intersectan en dos puntos de fuga antipodales.

El segundo algoritmo tipo RANSAC se inicia con una selección aleatoria de pares de líneas de las extraídas con anterioridad. A partir de cada par de rectas, y considerando la hipótesis de que son paralelas se calcula un punto de fuga que es votado por el resto de las rectas. La dirección de este punto de fuga se calcula mediante el producto vectorial de las dos normales correspondientes a las dos rectas. El criterio para la votación es una distancia basada en la perpendicularidad entre las normales de cada recta y su dirección. La dirección más votada se considerará la dirección dominante de la escena. El algoritmo se repite entre las rectas restantes extrayéndose la segunda y tercera dirección más dominante.

Una vez que cada recta ha sido asignada a una dirección y tenemos una estimación suficientemente robusta de las tres direcciones dominantes, se refina su estimación mediante una optimización en la que se impone la ortogonalidad entre ellas.

7.2 EXPERIMENTACIÓN

Para comprobar la eficacia del método comentado, vamos a hacer una serie de experimentos de manera que puedan darnos una idea de sus prestaciones.

7.2.1 EXPERIMENTACIÓN EN FOTOS DE EXTERIOR

Comenzamos evaluando el algoritmo con las fotos de exterior, donde están presentes entornos en los que, según el estado del arte [12], este tipo de métodos no presenta un buen funcionamiento.

En primer lugar se evaluará un set de 60 fotos orientadas horizontalmente, es decir, su vector vertical coincide con la dirección de la gravedad (0, 0, 1). Dado que el algoritmo usado da como resultado tres direcciones ortogonales, y a priori, no conocemos cuál de ellas es la dirección vertical, vamos a hacer dos experimentos distintos.

En el primero, a falta de más información sobre las direcciones extraídas, vamos a considerar como dirección vertical a la dirección más dominante extraída de RANSAC, es decir el primer vector de la base. En el segundo vamos a darle apoyo al algoritmo seleccionando de las tres direcciones que nos da la más cercana al vector vertical (0, 0, 1).

Una vez obtenidos los resultados vamos a considerar como fotos bien calculadas aquellas en las que el error entre el vector seleccionado como vertical y el vector vertical real no sea superior a un treshold angular determinado. Este error se calcula a partir del arco coseno del producto escalar entre ambas direcciones.

Dirección seleccionada como vertical	% de acierto para theshold de 5º	% de acierto para theshold de 10º	Media del error (grados)	Mediana del error (grados)
Dirección más dominante	9,6%	9,6%	53,85	57,74
Dirección más cercana a la vertical	21,15%	28,85%	21,45	20,74

Tabla 7.1 Resultados en imágenes de exterior horizontales

Comprobamos, que a pesar de que las fotos están orientadas de manera horizontal, el algoritmo no es eficaz a la hora de calcular correctamente la dirección vertical de las imágenes.

Pasamos ahora a aplicar el algoritmo a las fotos giradas de exterior del dataset de test, que son las mismas fotos con las que hemos evaluado las prestaciones de nuestra red. Se incluye también el resultado obtenido mediante nuestro método, al cual vamos a llamar *HorizonLineNet-360*, para facilitar la comparativa de los resultados.

Como en el caso anterior, vamos a hacer distintos experimentos según la dirección que vamos a elegir como vertical, pero ahora, a parte de los métodos de selección anterior, vamos a añadir uno nuevo en el que la dirección seleccionada como vertical va a ser la dirección más cercana a la dirección vertical del ground truth, la cual conocemos de antemano.

Método	% de acierto para theshold de 5º	% de acierto para theshold de 10º	Media del error (grados)	Mediana del error (grados)
Geométrico: dirección más dominante	3%	6,47%	43,33	43,36
Geométrico: dirección más cercana a la vertical	7,96%	14,43%	34,28	30,72
Geométrico: dirección más cercana al ground truth	15,92%	33,33%	15,93	15,08
HorizonLineNet-360	**93,5%**	-	**1,78**	**0,33**

Tabla 7.4 Resultados en imágenes de exterior rotadas

Como se podía esperar después de ver los resultados del algoritmo en las imágenes horizontales, se observa que, se confirma la hipótesis planteada en la introducción del trabajo, de manera que en entornos de exterior donde predominan los elementos naturales carentes de líneas rectas y composiciones geométricas, este método está lejos de ser eficaz, es decir, no puede ser usado como solución al problema de ajuste de la dirección vertical.

7.2.1 EXPERIMENTACIÓN EN FOTOS DE INTERIOR

Como hemos hecho con las fotos de exterior, comenzamos aplicando el algoritmo a 60 fotos, en este caso de interior, orientadas horizontalmente. Repetimos los mismos experimentos del apartado anterior manteniendo los criterios que se han seguido.

Dirección seleccionada como vertical	% de acierto para theshold de 5º	% de acierto para theshold de 10º	Media del error (grados)	Mediana del error (grados)
Dirección más dominante	51,92%	57,69%	33,66	6,02
Dirección más cercana a la vertical	75%	86,54%	5,4	2,12

Tabla 7.3 Resultados en imágenes de interior horizontales

Se observa que, al tener tres direcciones principales como solución, si no tenemos un buen criterio de selección a la hora de considerar una dirección como la dirección vertical, el método no resulta eficaz tampoco en entornos de interior. En este caso, en el que las fotos están orientadas horizontalmente, el criterio de selección de la dirección vertical como la dirección más cercana a la

vertical real da unos resultados más decentes, ya que, aunque el porcentaje de acierto en las predicciones no llega al 90%, tanto el error medio, como la mediana del error, representan una desviación a partir de la cual los humanos siguen estando fuertemente de acuerdo con la orientación de la foto, tal y como se ha explicado en el capítulo 6.

En los casos en los que, la dirección más cercana a la vertical se aleja en más de 10º de la vertical real, encontramos que la dirección más dominante coincide con el eje X (1, 0, 0), o con el eje y (0, 1, 0); y calculadas las otras dos direcciones en base a los criterios de ortogonalidad descritos al principio del capítulo, en un considerable porcentaje, no se llega a estimar correctamente la dirección vertical como una de las tres direcciones dominantes en la escena.

Pasamos ahora a aplicar el algoritmo en las fotos giradas de interior del dataset de test con el que se han evaluado las prestaciones de la red. Se incluye también el resultado obtenido mediante nuestro método, *HorizonLineNet-360*, para facilitar la comparativa de los resultados. Y al igual que con las fotos orientadas horizontalmente, repetimos los mismos criterios de experimentación empleados en el apartado anterior

Método	% de acierto para theshold de 5º	% de acierto para theshold de 10º	Media del error (grados)	Mediana del error (grados)
Geométrico: dirección más dominante	42,28%	53,73%	22,12	7,82
Geométrico: dirección más cercana a la vertical	21,4%	28,36%	32,78	32,25
Geométrico: dirección más cercana al ground truth	79,6%	87,06%	4,08	0,97
HorizonLineNet-360	**99,23%**	-	**0,47**	**0,15**

Tabla 7.4 Resultados en imágenes de interior rotadas

Como era de esperar y al igual que sucedía con las fotos horizontales, el algoritmo necesita de un apoyo para conocer cuál de las direcciones extraídas es la que estamos buscando. Eligiendo como dirección vertical, la dirección más cercana a la vertical del ground truth, vemos que los resultados vuelven a ser bastante aceptables, llegando a mejorar el porcentaje de acierto y el error respecto al grupo de fotos horizontales.

El principal problema de este método, es que para fotos en las que sepamos de antemano que van a estar orientadas horizontalmente o con desviaciones cercanas, podemos usar como referencia la dirección vertical de la referencia absoluta. Pero para otras orientaciones con desviaciones mayores de 45º no nos valdría esta aproximación, ya que se asignaría como dirección vertical una de las direcciones horizontales de la escena.

Este problema a la hora de conocer cuál es la dirección vertical según el método geométrico expuesto, resulta crítico a la hora de elegir este algoritmo como solución para la rectificación de la orientación en imágenes panorámicas.

CAPÍTULO 8

CONCLUSIONES

En este trabajo se ha investigado el uso de redes de aprendizaje profundo para la estimación de la línea del horizonte y los puntos de fuga verticales a partir de imágenes panorámicas. Como resultado del trabajo, hemos logrado alcanzar los objetivos planteados al desarrollar un método eficaz a la hora de rectificar la orientación vertical de imágenes panorámicas independientemente del tipo de escena que representen, solucionando de este modo la problemática presente en las fotos de exterior natural como bosques o playas, donde al no haber una composición de líneas rectas artificial como si ocurre en entornos de interior o entornos urbanos, los métodos geométricos clásicos referenciados en el estado del arte fallaban en su cometido.

Una de las principales contribuciones de este trabajo ha sido la explotación de técnicas de aprendizaje profundo (deep learning), más concretamente a partir de las redes neuronales convolucionales, las cuales son muy apropiadas para tareas que requieren información contextual de la imagen completa. La configuración autoencoder de la red, representando la salida de la misma como dos canales de imágenes que representan la línea del horizonte y los puntos de fuga se ha visto eficaz de cara al entrenamiento, siendo una representación de salida lo suficientemente detallada para una eficiente estimación numérica de la dirección vertical.

Los resultados experimentales demuestran que el método propuesto tiene un buen desempeño en la detección de la línea del horizonte y los puntos de fuga verticales superando a los métodos clásicos basados en rectas, no solo en escenarios desfavorables para ellos como entornos naturales exteriores, sino también en escenas de interior donde se esperaba un comportamiento similar.

Destacar que el método propuesto es capaz de funcionar en escenas y situaciones con las que los métodos basados en rectas directamente fallan, como escenas de exterior donde la dirección de la gravedad está implícita en el contexto de la imagen o en escenas de interior donde la identificación de techo y suelo es capaz de resolver la ambigüedad entre direcciones dominantes.

BIBLIOGRAFÍA

[1] J.-C. Bazin and M. Pollefeys. 3-line RANSAC for orthogonal vanishing point detection. In IROS, 2012.

[2] J. Jung, B. Kim, J. Lee, B. Kim, and S. Lee. Robust upright adjustment of 360 spherical panoramas. The Visual Computer, 2017.

[3] C. Demonceaux, P. Vasseur, and C. Pegard. Omnidirectional vision on UAV for attitude computation. In ICRA, 2006.

[4] C. Demonceaux, P. Vasseur, and C. Pegard. Robust attitude estimation´ with catadioptric vision. In IROS, 2006.

[5] A. C. Gallagher. Using vanishing points to correct camera rotation in images. In Canadian Conference on Computer and Robot Vision, 2005.

[6] H. Lee, E. Shechtman, J. Wang, and S. Lee. Automatic upright adjustment of photographs with robust camera calibration. TPAMI, 2014.

[7] S. T. Barnard. Interpreting perspective image. Artificial Intelligence Journal, 1983.

[8] J. P. Barreto. A unifying geometric representation for central projection systems. CVIU, 2006.

[9] C. Geyer and K. Daniilidis. Catadioptric projective geometry. IJCV, 2001.

[10] C. Mei. Laser-augmented omnidirectional vision for 3D localisation and mapping. In PhD Thesis, 2007.

[11] X. Ying and Z. Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model? In ECCV, 2004.

[12] J.-C. Bazin, C. Demonceaux, P. Vasseur, and I. S. Kweon. Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. IJRR, 2012.

[13] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2004.

[14] J.-C. Bazin, C. Demonceaux, P. Vasseur, and I. S. Kweon. Motion estimation by decoupling rotation and translation in catadioptric visión CVIU, 2009.

[15] J.-C. Bazin, I. S. Kweon, C. Demonceaux, and P. Vasseur. UAV attitude estimation by vanishing points in catadioptric images. In ICRA, 2008.

[16] J.-C. Bazin, Y. Seo, and M. Pollefeys. Globally optimal consensus set maximization through rotation search. In ACCV, 2012.

[17] L. Zhang, H. Lu, X. Hu, and R. Koch. Vanishing point estimation and line classification in a Manhattan world with a unifying camera model. IJCV, 2016.

[18] K. Joo, T.-H. Oh, I. S. Kweon, and J.-C. Bazin. Globally optimal inlier set maximization for Atlanta frame estimation. In CVPR, 2018.

[19] M. Antone and S. Teller. Scalable extrinsic calibration of omnidirectional image networks. IJCV, 2002.

[20] H. Guan and W. A. P. Smith. Structure-from-motion in spherical video using the von Mises Fisher distribution. TIP, 2017.

[21] M. Kamali, A. Banno, J.-C. Bazin, I. S. Kweon, and K. Ikeuchi. Stabilizing omnidirectional videos using 3D structure and spherical image warping. In MVA, 2011.

[22] J. Kopf. 360° video stabilization. TOG, 2016.

[23] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand. SKYLINE2GPS: localization in urban canyons using omni-skylines. In IROS, 2010.

[24] D. Scaramuzza and R. Siegwart. Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. In IEEE Transactions on Robotics, 2008.

[25] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In IROS, 2008.

[26] P. Fischer, A. Dosovitskiy, and T. Brox. Image orientation estimation with convolutional networks. In German Conference on Pattern Recognition (GCPR), 2015.

[27] U. Joshi and M. Guerzhoy. Automatic photo orientation detection with convolutional neural networks. In Conference on Computer and Robot Vision (CRV), 2017.

[28] Y. Hold-Geoffroy, K. Sunkavalli, J. Eisenmann, M. Fisher, E. Gambaretto, S. Hadap, and J.-F. Lalonde. A perceptual measure for Deep single image camera calibration. In CVPR, 2017.

[29] G. Olmschenk, H. Tang, and Z. Zhu. Pitch and roll camera orientation from a single 2D image using convolutional neural networks. In Conference on Computer and Robot Vision (CRV), 2017.

[30] Zhang Yi. A fixation-based 360-degree image dataset (Enero, 2020). Recuperado de: https://github.com/PanoAsh/F-360iSOD.

[31] Equirectangular (Marzo 2020). Recuperado de: https://www.flickr.com/groups/equirectangular/.

[32] J. Canny. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6):679–698, 1986.

[33] R. Jung, A. S. J. Lee, A. Ashtari and J. Bazin, "Deep360Up: A Deep Learning-Based Approach for Automatic VR Image Upright Adjustment," 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 2019.

[34] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In IEEE International Conference on Computer Vision, volume 2, pages 941–947, 1999.

[35] C. Fernandez-Labrador, J. M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera and J. J. Guerrero, "Corners for Layout: End-to-End Layout Recovery From 360 Images," in IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1255-1262, April 2020