PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

# DEEPSOCNAV: SOCIAL NAVIGATION BY IMITATING HUMAN BEHAVIORS

**JUAN PABLO DE VICENTE**

Tesis para optar al grado de
Magíster en Ciencias de la Ingeniería

Profesor Supervisor:
ÁLVARO SOTO ARRIAZA

Santiago de Chile, Enero 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

# DEEPSOCNAV: SOCIAL NAVIGATION BY IMITATING HUMAN BEHAVIORS

**JUAN PABLO DE VICENTE**

Miembros del Comité:
ÁLVARO SOTO ARRIAZA
DAVID ACUÑA URETA
IVÁN LILLO VALLES
CRISTIÁN TEJOS NÚÑEZ

Tesis para optar al grado de
Magíster en Ciencias de la Ingeniería

Santiago de Chile, Enero 2021

*Agradecidamente a mi familia y a*
*mis amigos*

## AGRADECIMIENTOS

Este trabajo fue parcialmente financiado por el proyecto FONDECYT número 1181739.

Un agradecimiento especial a mi profesor guía Álvaro Soto, quien me guió en mi aprendizaje durante esta investigación, haciendo este trabajo posible.

Agradezco a mi padre Rafael (Q.E.P.D), quien siempre me apoyó y me motivó a seguir mis sueños. Agradezco también a mi madre Carolina y a mis hermanos Tomas y Magdalena, por su indispensable apoyo y cariño.

Agradezco a mis amigos y compañeros de IALab, especialmente a los miembros del grupo Robótica Cognitiva, por su ayuda y consejos.

# ÍNDICE GENERAL

# ÍNDICE DE FIGURAS

# ÍNDICE DE CUADROS

## ABSTRACT

Current 2D representations and coordinate based interactions for social navigation leave aside precious relationships and visual cues that could only be captured through a first-person view of the scene. This limits the performance of data-driven social navigation models. In this work, we propose to exploit the power of current game engines, such as Unity, to transform pre-existing bird's-eye view datasets into a first-person view, in particular, a depth view. Furthermore, we show the benefits of using synthetic data from a game engine to pre-train a navigational model. To test our ideas, we present DeepSocNav, a deep learning model that takes advantage of the transformed and synthetic datasets, as well as, a self-supervised strategy that is included as an auxiliary task. This task consists of anticipating the next depth frame that the agent will face. Our experiments show the benefits of the proposed model that is able to outperform relevant baselines in terms of social navigation scores.



**Keywords: Social Navigation, Deep Learning Model, Robotic's Dataset**.

## RESUMEN

Las actuales representaciones 2D e interacciones basadas en coordenadas para la navegación social dejan de lado importantes relaciones y pistas visuales que pueden ser capturadas solamente a través de una vista en primera persona de la escena. Esto limita el rendimiento de modelos de navegación social del tipo *data-driven*. En este trabajo, proponemos aprovechar las herramientas de los motores de videojuegos actuales, -como Unity-, para transformar datasets preexistentes con vista de pájaro a datasets con vista en primera persona, y en particular una vista de profundidad. Además, demostramos los beneficios de usar data sintética generada por el motor de videojuegos para pre-entrenar un modelo de navegación. Para probar nuestras ideas, presentamos DeepSocNav, un modelo de aprendizaje profundo el cual toma ventaja de los datasets transformados y sintéticos, además de una estrategia auto-supervisada incluida en la forma de una tarea auxiliar. Esta tarea consiste en anticipar la siguiente imagen de profundidad que el agente verá. Nuestros experimentos muestran el beneficio del modelo propuesto el cual es capaz de superar *baselines* relevantes en términos de métricas sociales de navegación.

**Palabras Claves: Navegación Social, Modelo de Aprendizaje Profundo, *Dataset* de Robótica**.

# CAPÍTULO 1. INTRODUCCIÓN

## 1.1. Motivación

En el último tiempo se ha visto una creciente diversificación en las tareas desarrolladas por robots. En particular, existen nuevas aplicaciones en ambientes naturales donde robots deben interactuar con humanos, o al menos compartir el espacio con ellos. Ejemplo de ello son los robots asistenciales o de servicio, los cuales pueden ejercer trabajos como ofrecer alimentos a los invitados en un servicio de cáterin (Lai y Tsai, 2018), entregar servicio a la habitación en un hotel (Zhong y Verma, 2019) o ser guía en un aeropuerto (Triebel y cols., 2016). Un ejemplo de este último puede verse en la figura 1.1.

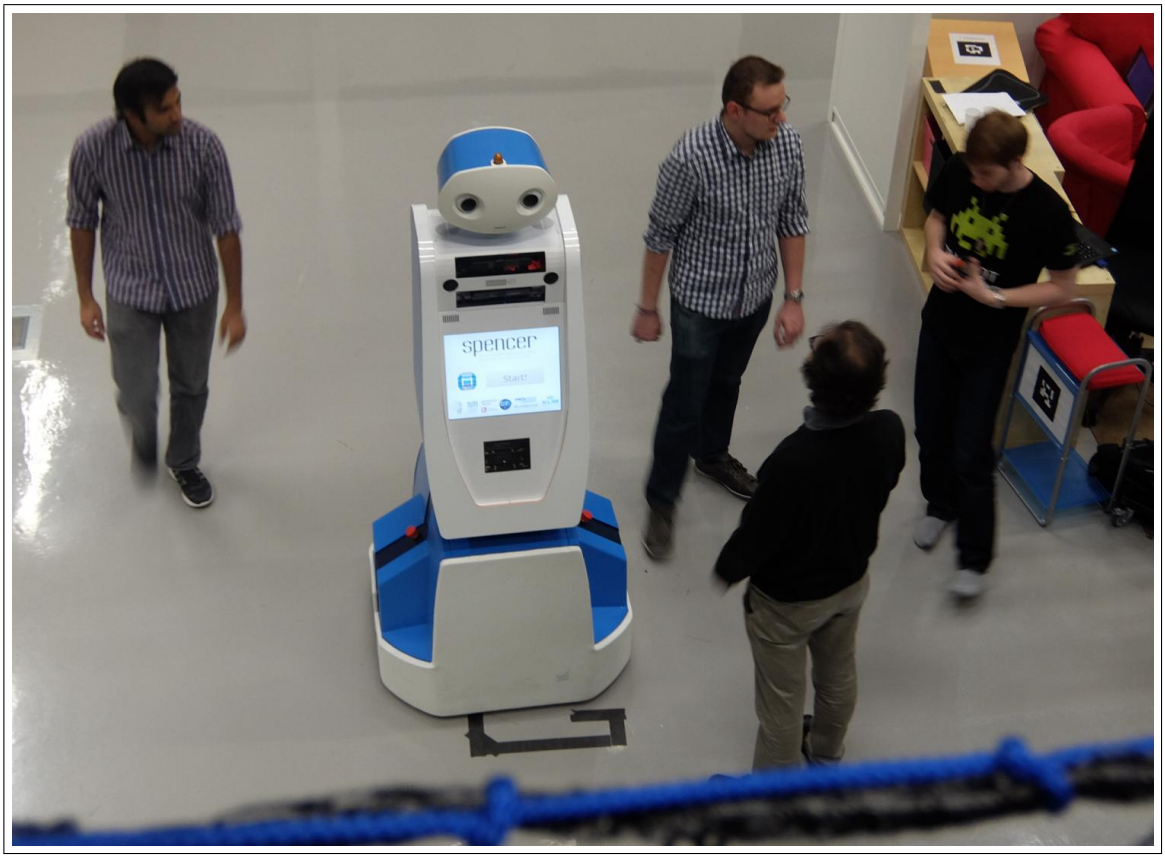



Figura 1.1. SPENCER: Robot asistencial para aeropuertos. Fuente de la imagen: LASS Laboratoire.

Acorde a esta transición, los algoritmos de control y comportamiento de este tipo de robots han tenido que ser modificados para ajustarse a sus nuevas condiciones de trabajo. Estos nuevos algoritmos no solo deben asegurar la seguridad de las personas con las que el robot pueda tener contacto, sino que también deben evitar incomodarlas. Con esto en mente, muchos de los nuevos comportamientos buscan imitar la conducta humana. Siguiendo esta linea de investigación, en esta tesis se busca desarrollar un nuevo sistema de navegación social robótica en base a la imitación de la navegación social humana.

Si bien cada robot debe contar con habilidades especificas dependiendo de su labor, todo robot móvil que se emplee en un ambiente natural compartido con personas debe ser capaz de desplazarse en forma que no entorpezca o incomode a las personas presentes, es decir, deben ser capaces de ejercer una navegación social. Debido a esto, técnicas clásicas de navegación basadas en algoritmos como A* o simple evasión de obstáculos estáticos o dinámicos no son aplicables (Siegwart y cols., 2004) porque no toman en consideración los comportamientos intrínsecos de las personas. Algunos ejemplos de ello son transitar por un lado específico de la vereda o mantener una distancia prudente del resto de los transeúntes.

## 1.2. Tipos de soluciones y trabajo previo

Expertos han tratado de resolver este problema mediante distintos tipos de trabajos. En particular, estos pueden clasificarse en 2 grupos: *model-driven* y *data-driven*.

### 1.2.1. *Model-driven*

Las soluciones *model-driven* se caracterizan por utilizar un profundo entendimiento de las relaciones y reglas que componen un sistema para tomar decisiones sobre él. Esto quiere decir que si se tiene una representación completa de cómo se comporta un sistema, podemos saber *a priori* cual será la salida del sistema en función de la entrada. Los problemas aparecen cuando el sistema es demasiado complejo para modelar, momento donde

se deben tomar simplificaciones o aproximaciones que pueden ocasionar comportamientos inesperados. Ejemplo de soluciones *model-driven* para el problema de la navegación social son (Alahi y cols., 2014; Leal-Taixé y cols., 2011; Lerner y cols., 2007; Luber y cols., 2010; Pellegrini y cols., 2009; Pellegrini y cols., 2010; Trautman y Krause, 2010; Treuille y cols., 2006; Yamaguchi y cols., 2011), las cuales utilizan el concepto de social force introducido por Helbing y Molnar (1998) para describir el comportamiento social de navegación de los transeúntes.

### 1.2.2. *Data-driven*

Por otro lado, las soluciones *data-driven* buscan predecir el comportamiento correcto mediante un gran número de ejemplos, los cuales usa para encontrar relaciones y probabilidades que rigen al sistema sin la necesidad de explicitarlas en la programación del algoritmo. Aquello le otorga la ventaja de poder inferir relaciones complejas presentes en los datos de forma autónoma. Debido a que el comportamiento humano es altamente estocástico (Chen y cols., 2017; Kretzschmar y cols., 2016) y varía dependiendo de la cultura (*The wisdom of crowds*, 2011) las técnicas *data-driven* han obtenido mejores resultados que su contraparte *model-driven* (Chen y cols., 2016; Chen y cols., 2017; Fahad y cols., 2018; Joo y cols., 2019; Hamandi y cols., 2019). La desventaja de este paradigma es que no solo se necesita de un gran volumen de datos, sino que también el tipo de solución es dependiente del tipo de datos disponible.

### 1.2.3. Trabajo previo

Un trabajo que permitió el desarrollo de múltiples métodos de navegación social fue el de social force presentado por Helbing y Molnar (1998). En él se presentaron los comportamientos sociales como una fuerza de atracción o repulsión de acuerdo a campos potenciales generados por los peatones presentes en la escena. Leal-Taixé y cols. (2011) utilizaron la social force para mejorar el rendimiento de seguidores de personas en ambientes concurridos. Luber y cols. (2010) modelaron las dinámicas de las personas a través de

un modelo de social force para predecir sus movimientos. Pellegrini y cols. (2009) obtiene buenos rendimientos para seguir una persona al combinar la social force con conocimiento de la escena incluso con oclusión. Treuille y cols. (2006) hacen uso de campos potenciales basados en social force para evitar diseñar un evasor de obstáculos explícito. Es decir, que se le deban indica que comportamientos tomar frente a un obstáculo para evadirlo. Estos trabajos utilizan técnicas *model-driven*, las cuales buscan diseñar un sistema de acuerdo a reglas y representaciones explícitas las cuales deben son definidas de forma manual. El problema con este tipo de diseño es que al aumentar la complejidad de la tarea, comienzan a aparecer dinámicas demasiado complejas para modelar manualmente, generando problemas como el *Freezing Robot Problem*, el cual Trautman y Krause (2010) buscan resolver mediante modelos estadísticos basados en procesos Gaussianos.

Como consecuencia, últimamente han tomado mayor relevancia los algoritmos *data-driven*, los cuales buscan construir un sistema que identifique la respuesta correcta en base a los datos observados. Uno de los primeros ejemplos fue presentado por Kuderer y cols. (2012), donde hacen uso de *inverse reinforcement learning* (IRL) en base al principio de máxima entropía para aprender *features* que capturen aspectos relevantes de la navegación humana. Paralelamente, Kitani y cols. (2012) combinaron el uso de IRL con un conocimiento semántico de la escena, lo que les permitió modelar el efecto del ambiente en las decisiones de navegación de las personas. Posteriormente, Fahad y cols. (2018) entrenaron un modelo de IRL capaz de generar trayectorias con comportamientos sociales similares a las de humanos utilizando un gran volumen de datos y *Social Affinity Maps* (SAM). Algo que caracteriza a IRL es que primeramente el modelo debe aprender las recompensas implícitas en el dataset, y posteriormente se debe desarrollar un algoritmo que haga uso de estos recompensas para obtener las acciones.

Otra línea del tipo *data-driven* es aquella basada en *deep reinforcement learning* (DRL), la cual necesita de recompensas y penalizadores diseñados manualmente para aprender una política. Un ejemplo de esto se encuentra en (Chen y cols., 2017), donde diseñaron

penalizadores en torno a lo que el robot no debe hacer, en contraste a recompensar comportamientos deseados, para obtener una conducta de navegación social.

Finalmente, existen trabajos basados en el aprendizaje supervisado. Estos no requieren de *features* o recompensas diseñadas a mano. Alahi y cols. (2016) hacen uso del aprendizaje supervisado y redes recurrentes del tipo Long Short Term Memory (LSTM) (Hochreiter y Schmidhuber, 1997), para captar los comportamientos sociales presentes en secuencias de datos para poder predecir su trayectoria futura. Estas LSTM reciben información de las LSTM cercanas mediante *pooling*, lo que les permite conocer una representación del estado de otros transeúntes. Vemula y cols. (2017), por otro lado, usan una LSTM con atención sobre una representación de grafo de los transeúntes para que el modelo aprenda que no todos los transeúntes son igual de importantes a la hora de predecir trayectorias sociales. En (Alahi y cols., 2016) y (Vemula y cols., 2017) los agentes son representados como coordenadas en un mapa 2D, Sadeghian y cols. (2017), por su parte, proponen usar imágenes en vista de pájaro para predecir trayectorias futuras utilizando un modelo de CNN con atención y una LSTM. Este trabajo busca aprovechar las características de la escena presente en las imágenes para predecir mejores trayectorias. En una línea paralela al de la navegación social, trabajos como los de Gupta y cols. (2018), Kosaraju y cols. (2019), Sadeghian y cols. (2018) y Tai y cols. (2017) buscan predecir trayectorias sociales dado un fragmento inicial de éstas utilizando *generative adversarial networks* (GAN). Finalmente, Hamandi y cols. (2019) simula escaneados LiDAR a partir de datos etiquetados en vista de pájaro para entrenar la navegación de un robot móvil, aprendiendo los comportamientos sociales presentes en el dataset. En esta investigación daremos un paso mas allá y simularemos la vista en primera persona para aprovechar indicios visuales que no pueden apreciarse en una vista 2D aérea, ya que los consideramos necesarios para aprender una navegación social más avanzada.

## 1.3. Descripción del problema

Al día de hoy, los datos disponibles públicamente en la web para el entrenamiento de modelos de navegación social consisten únicamente en grabaciones en tercera persona, generalmente en vista de pájaro, de transeúntes navegando en ambientes con distintos niveles de densidad de personas. Debido a esto, los modelos que utilizan directamente aquellos datos, como es el caso de los modelos *data-driven* con aprendizaje supervisado presentados en la sección 1.2.3, están limitados a trabajar con una representación 2D para la navegación. Esto deja de lado relaciones que solo pueden capturarse mediante observaciones 3D en primera persona, relaciones que pueden ser de gran utilidad para mejorar el rendimiento de los sistemas de navegación social. Algunos ejemplo de este tipo de relaciones son el movimiento de los brazos con la velocidad del agente y la orientación del cuerpo con la dirección de movimiento.

Para poder desarrollar un modelo *data-driven* basado en observaciones en primera persona, una posible solución sería recolectar y etiquetar un set de datos de navegación en primera persona. Lamentablemente esta solución tiene dos problemas asociados. El primero es la mayor complejidad para medir la posición y velocidad del agente que lleva la cámara. El segundo -y mas relevante- es el costo en tiempo necesario para recolectar un volumen suficiente de datos en primera persona. En 5 minutos a través de una cámara en vista de pájaro se pueden recolectar cientos de trayectorias pertenecientes a todos los transeúntes que transitaron por el área durante la grabación. En cambio, al recolectar datos en primera persona, en esos 5 minutos de grabación solo se recolectara una sola trayectoria: la del agente que llevaba la cámara. Tomando todo esto en consideración, no es sorpresa que a la fecha no exista un *dataset* de gran volumen en primera persona que sea de utilidad para la navegación social.

## 1.4. Hipótesis

Dada una trayectoria 2D en vista de pájaro, es posible recrear ésta en un ambiente virtual 3D donde se tenga acceso a información visual similar a la que tuvieron los agentes presentes en la escena cuando se captura la trayectoria.

Respecto a las trayectorias recreadas, por simplicidad se tomaron los siguientes supuestos respecto a la naturaleza de los datos:

1. Los peatones solo utilizan su vista y no se comunican verbalmente entre ellos.
2. Cada agente tiene como objetivo llegar a un punto específico en el mapa antes de darse como terminada su navegación.

Otra hipótesis para nuestro trabajo consiste en que dado un set de ejemplos de trayectorias de transeúntes en ambientes concurridos y acceso a la vista en primera persona de los agentes, es posible desarrollar mediante técnicas de aprendizaje profundo, un modelo capaz de aprender a replicar los comportamientos sociales implícitos presentes en las trayectorias de los transeúntes a partir de su visión y conocimiento de la escena, en especifico, su ubicación y la del objetivo.

Además, se hipotetiza que para una mejor navegación social, el modelo debe contar con los siguientes componentes y habilidades:

- Inteligencia espacial: la capacidad de conocer dónde uno se encuentra y la dirección que debe tomar para llegar a una meta.
- Reconocimiento de agentes y obstáculos: si uno no es capaz de percibir y reconocer la presencia de otros individuos o de obstáculos que obstruyen nuestra trayectoria, entonces resulta imposible saber cuándo cambiar nuestro comportamiento para evitar colisionar con ellos.
- Memoria de corto plazo: es necesario tener la capacidad de recordar, al menos en el corto plazo, las posiciones previas tanto de uno mismo como la de los otros agentes para poder inferir la dirección y movimiento de estos en el ambiente.

## 1.5. Objetivos

Los objetivos generales de esta investigación son:

- Obtener acceso a imágenes en primera persona a partir de datos etiquetados de trayectorias en vista de pájaro.
- Desarrollar un sistema de navegación social robótica basado en utilizar imágenes en primera persona como *input*.

Dados estos objetivos generales, los objetivos específicos para cada uno de estos se describen a continuación:

- Desarrollar un ambiente virtual realista que mediante la replicación de datos pre-existentes en vista de pájaro, nos de acceso a la vista en primera persona de cada uno de los agentes.
- Diseñar, implementar y evaluar un sistema de navegación social robótica basado en el aprendizaje e imitación de trayectorias humanas presentes en un set de datos. Este sistema de navegación debe tener la característica de utilizar una vista en primera persona, con el fin de aprovechar patrones visuales ricos en información que solo están presentes desde esta perspectiva.

## 1.6. Contribuciones

Este trabajo aporta en primera instancia a superar la limitante de obtener observaciones de trayectorias de peatones en primera persona mediante el desarrollo de un ambiente virtual, el cual tiene la capacidad de replicar la abundante cantidad de datos en vista de pájaro dando acceso a la vista en primera persona de todos los agentes presentes en la escena. Este ambiente simulado cuenta además con modelos y animaciones realistas de humanos, aumentando así la información disponible en los datos de trayectorias 2D con movimientos humanos completos en un ambiente 3D. De esta forma damos acceso al desarrollo de soluciones *data-driven* que sean capaces de aprovechar imágenes ricas en

información visual de la escena y de los otros transeúntes acorde a lo que vería un robot presente en ambientes concurridos. Cabe mencionar que estas escenas son replicadas de manera automatizada y solo requieren el diseño manual de un mapa si se desean considerar obstáculos inanimados. A partir de las herramientas ofrecidas en este ambiente virtual, es también posible generar escenas artificiales de múltiples agentes navegando a puntos aleatorios o predefinidos en el mapa. Estas escenas artificiales -al ser generadas por el algoritmo de navegación del ambiente virtual- no aseguran un comportamiento social, pero de todas formas son útiles ya que nos ofrecen un número infinito de ejemplos de navegación y evasión de obstáculos, los cuales pueden utilizarse para pre-entrenar un modelo de navegación social con el fin de que este inicie su entrenamiento social con una base robusta de navegación.

En conjunto al ambiente simulado, presentamos también DeepSocNav, un modelo de aprendizaje supervisado que busca imitar los comportamientos sociales de humanos reales en ambientes concurridos, mediante su vista de profundidad en primera persona. Este modelo es capaz de aprender a replicar los patrones de velocidad que tuvieron los agentes en los datos de acuerdo a lo que ellos vieron en ese instante, junto con conocer su posición y la ubicación del objetivo final de la trayectoria. Gracias a que los datos de entrada son más similares a lo que utilizan las personas para transitar, el modelo se ve alentado a desarrollar una política social más cercana a la que desarrollan las personas. Para guiar el aprendizaje en esta línea, se le agregó al modelo la tarea auxiliar de predecir como se verá la escena un instante de tiempo en el futuro, la cual guía al modelo a desarrollar una mejor noción del espacio y de los agentes en su campo visual, mejorando así su rendimiento social y su capacidad para evadir colisiones.

En resumen, las principales contribuciones de esta tesis son las siguientes:

- Presentamos a través de un ambiente virtual un método que permite aprovechar la abundante cantidad de datos en vista de pájaro para generar datos en primera persona, los cuales pueden ser usados para entrenar modelos de navegación social acorde al tipo de observación que vería un robot con una cámara frontal al

navegar en ambientes concurridos -como los presentes en los datos originales-. Además, este ambiente virtual permite generar un número ilimitado de ejemplos de navegación y evasión de obstáculos, aunque estos no presentan necesariamente una componente social.

- Presentamos también un modelo basado en aprendizaje profundo supervisado que predice una salida de velocidad basada en imágenes de profundidad en primera persona y las coordenadas actuales y de destino. Este modelo busca replicar los comportamientos sociales durante la navegación de personas reales en ambientes concurridos presentes en el dataset utilizado. Para guiarlo en su aprendizaje, se le exigirá a través de otra cabecera la tarea de predecir como se verá la escena en primera persona un instante de tiempo en el futuro.

Realizamos además experimentos para evaluar los distintos componentes de nuestro modelo en trayectorias de manera online, es decir, en tiempo real haciendo uso de nuestro mismo ambiente simulado.

## 1.7. Organización del documento

El cuerpo de esta tesis se organiza de la siguiente manera. El capítulo 2 presenta la metodología con la cual se busca aportar a la solución del problema de la navegación social. Dentro de este, en la sección 2.1 se presenta el método utilizado para la obtención de datos en primera persona mediante la replicación autónoma de datos en vista de pájaro y la generación de datos sintéticos. Posteriormente, en la sección 2.2, se describe y detalla el modelo de navegación social propuesto (DeepSocNav). La implementación y descripción de los métodos y métricas para la evaluación del modelo se presentan en el capítulo 3, mientras que los resultados de la evaluación y su posterior análisis se encuentran en el capítulo 4. Finalmente las conclusiones y trabajo futuro se encuentran en el capítulo 5.

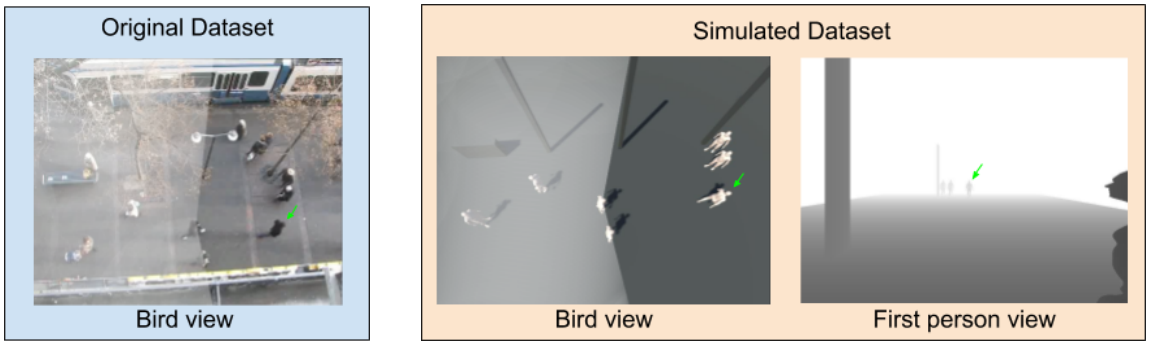
# CAPÍTULO 2. MÉTODO PROPUESTO

## 2.1. Ambiente simulado y generación de datos

La orientación en la que miran los otros transeúntes y la forma en la que estos mueven su cuerpo nos dan pistas sobre cuáles son sus intenciones, velocidades y direcciones, lo que permite planear una ruta acorde para no colisionar ni entorpecerlos. Creemos que esta información se ve reflejada en las trayectorias y velocidades de cada agente en la escena. La mayoría de las soluciones actuales para la navegación social utilizan datos 2D desprovistos de este tipo de información, lo que limita a sus modelos a generar relaciones solo entre las posiciones y velocidades de los agentes en el tiempo. Esto motiva nuestro trabajo a aprovechar imágenes de profundidad en primera persona ricas en relaciones implícitas presentes en los datos, lo que permite que modelos desarrollen una navegación social más fiel al comportamiento presente en los datos.

### 2.1.1. Generación de datos en primera persona

Para la recreación de datasets se desarrolló un ambiente simulado utilizando el motor para el desarrollo de videojuegos Unity (Technologies, 2018). Este ambiente carga mapas previamente diseñados y, a partir de la secuencia de datos de la posición de los agentes en el tiempo, automáticamente los ubica y hace navegar en su escenario respectivo utilizando la librería de *pathfinding* y navegación NavMesh de Unity, la cual hace uso de *reciprocal velocity obstacles* (RVO) para la evasión de obstáculos (van den Berg y cols., 2008). Esta librería permite desplazar a cada agente $i$ desde su posición actual $(x_t^i, y_t^i)$ hasta su posición futura $(x_{t+1}^i, y_{t+1}^i)$ en cada instante de tiempo $t$ hasta que el agente alcanza su objetivo final $(G_x^i, G_y^i)$. Estos desplazamientos siguen una animación realista de caminata, la cual es correspondiente a la dirección y velocidad que tiene el agente durante su trayectoria. Esto se logra mediante el paquete de control de animación de Unity, el que asocia los movimientos del agente a una respectiva animación de acuerdo a su orientación y rapidez.




(a) BIWI dataset: captura en vista de pájaro de hotel.

(b) Simulación en vista de pájaro y primera persona.

Figura 2.1. (A) Captura representativa de la escena real en el dataset original. (B) Vista de profundidad en primera persona del cuadro en a) recreado por el simulador.

Por simplicidad, los mapas se reconstruyeron utilizando Sketchup, software de modelado 3D el cual tiene completa compatibilidad con Unity.

Para cada agente se almacena su vista de profundidad en primera persona, en conjunto a su coordenada actual $(x_t^i, y_t^i)$, la coordenada de destino $(G_x^i, G_y^i)$, y la velocidad $(vx_t^i, vy_t^i)$ en cada instante de tiempo. A partir de estos datos, se espera que el modelo logre extraer los comportamientos sociales implícitos en las trayectorias del dataset original. Un ejemplo de un frame recreado en el ambiente simulado puede verse en la figura 2.1, donde (A) corresponde al frame en el video original y (B) a su recreación en el ambiente simulado en conjunto a la vista en primera persona del agente.

### 2.1.2. *Data Augmentation*

En el contexto de nuestro trabajo, una problemática que actualmente tienen los datasets de trayectorias de peatones son la cantidad de peatones totales en la escena. El dataset más extenso a la actualidad corresponde al Stanford Drone Dataset (Robicquet y cols., 2016) con un total de 11.2K peatones repartidos en 6 escenas distintas. El problema, es que en este dataset también hay presentes bicicletas, skateboards, automóviles, buses y

carros de golf, lo cual puede ser de gran utilidad para estudiar las relaciones entre estos y los peatones, pero que para nuestro caso pueden ocasionar ruido, ya que deseamos solo modelar las interacciones entre personas. Datasets puramente de peatones, por otro lado, tienen volúmenes más reducidos, en el orden de 1k personas en total. Número bajo para un correcto entrenamiento supervisado.

Un segundo problema es la naturaleza de dichos datos: éstos fueron grabados en un ambiente donde se puede ignorar a los otros peatones, y caminar línea recta hacia el objetivo, sin generar colisiones en una proporción considerable de cada trayectoria completa tal como puede verse en en la figura 2.2. Esto puede ocasionar que el modelo pase por alto la necesidad de aprender habilidades básicas de navegación, tales como doblar y desviarse de una línea recta con el objetivo de evitar colisiones.

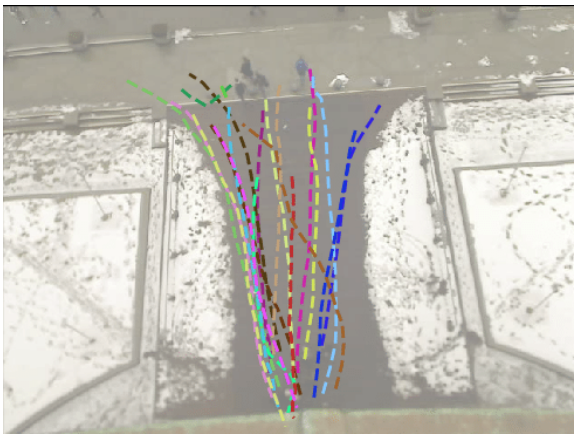


Figura 2.2. Ejemplo de trayectorias en dataset BIWI de peatones. Fuente: Stefan Becker

Como solución a ambos problemas, se aprovechó la misma herramienta de navegación de Unity que se utilizó para seguir los puntos de las trayectorias en el dataset original, pero ahora se le entregaron puntos aleatorios en el mapa a múltiples agentes a distintas velocidades. Estas trayectorias sintéticamente generadas tienen la característica de que se pueden crear de manera automática y NavMesh se encarga mediante sus algoritmos de

*pathfinding* de producir trayectorias que evitan la colisión entre agentes, incluso para escenarios con múltiples decenas de ellos. Mediante aquello tenemos acceso a una cantidad teóricamente infinita de ejemplos de navegación con *pathfinding* y evasión de obstáculos, gracias a lo cual contamos con datos suficientes para que el modelo aprendiera una base de navegación. Debido a que estas trayectorias artificiales no aseguran un comportamiento social, el modelo es posteriormente entrenado con los datos reales recreados mediante la técnica de *fine-tuning*. Esto con el objetivo de que el modelo incorpore la componente social presente en el dataset utilizado. De esta forma, los datos sintéticos le dan una base inicial de navegación y evasión de obstáculos al modelo, lo que posteriormente le permite extraer e incorporar con mayor facilidad los patrones sociales presentes en las trayectorias de peatones reales.

## 2.2. DeepSocNav

El problema de la navegación social es visto como un problema de predicción de velocidad dada una secuencia de observaciones. Para cada instante de tiempo $t$, el agente $i$ tendrá una medición de su posición actual $(x_t^i, y_t^i)$, la coordenada de su objetivo final en la trayectoria $(Gx_t^i, Gy_t^i)$, además de una imagen de profundidad $D_t^i$ correspondiente a lo que el agente ve en primera persona en ese instante de tiempo. A partir de una secuencia de observaciones $t - T$ a $t$ se predice su velocidad en el instante siguiente de tiempo $(vx_{t+1}^i, vy_{t+1}^i)$. De esta forma el modelo puede definirse como una función $f(s_{t-T}, ..., s_t, G)$ descrita de la siguiente forma:

$$\begin{aligned} f(s_{t-T}, ..., s_t, G) &= \vec{v}_{t+1} \\ s_t &= (x_t, y_t, D_t) \\ G &= (Gx, Gy) \end{aligned} \tag{2.1}$$

Una vista general de nuestro modelo puede verse en la figura 2.3, donde $s_t$ se presenta como la secuencia de imágenes de profundidad $D_t$ y la información posicional del agente

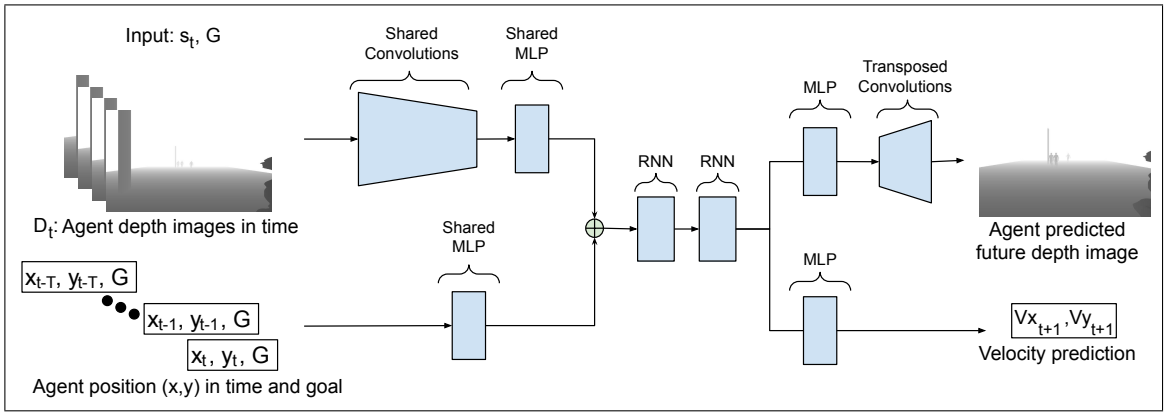



Figura 2.3. Arquitectura general de DeepSocNav. Este modelo predice la velocidad de un agente para el tiempo $t+1$ dado un historial de observación previas desde el paso $t-T$ a $t$.

$(x_t, y_t)$ desde el tiempo pasado $t-T$ hasta el tiempo presente $t$, ambas entradas presentes a la izquierda del diagrama.

Las entradas posicionales $(x_t, y_t)$ y de objetivo $(Gx, Gy)$ están normalizadas a partir del valor más grande y más pequeño posible entre todos los mapas. Debido a que las cámaras de profundidad funcionan en un rango acotado, imitamos este característica definiendo el valor del píxel $p_{\mathbf{x},\mathbf{y},\mathbf{t}}$ en las imágenes de profundidad de la siguiente forma:

$$p_{\mathbf{x},\mathbf{y},\mathbf{t}} = \begin{cases} d_{\mathbf{x},\mathbf{y},\mathbf{t}}/d_{max}, & \text{si } d_{\mathbf{x},\mathbf{y},\mathbf{t}} < d_{max} \\ 1, & \text{en caso contrario} \end{cases} \tag{2.2}$$

donde $d_{\mathbf{x},\mathbf{y}}$ corresponde a la distancia entre la cámara del agente y la superficie ubicada en el píxel $p_{\mathbf{x},\mathbf{y}}$, y $d_{max}$ es la distancia máxima que permite la cámara de profundidad.

La arquitectura de DeepSocNav es la siguiente. En primera instancia una capa *fully connected* recibe de entrada la posición del agente y la del objetivo en el mapa, aumentando su dimensionalidad a 64. De esta etapa se espera que el modelo extraiga una idea de la dirección en la que se encuentra el objetivo. Paralelamente a esto, la imagen de profundidad es procesada por una línea de 3 capas convolucionales seguidas por una capa densa con dimensionalidad de 128. La tarea de esta sección corresponde a reconocer patrones que

le permitirían al modelo reconocer otros agentes, obstáculos y espacios vacíos por donde transitar en su campo visual. Estas capas procesan las últimas T observaciones y mediciones, generando así un tensor de [T, 192]. En base al estudio de Alahi y cols. (2016), aplicamos una LSTM que recibe como entrada el tensor [T, 192]. Utilizamos dos capas de LSTM, cada una con una dimensión de 512 para los estados ocultos. Se espera que la LSTM le otorgue la capacidad al modelo de generar el entendimiento temporal necesario para la navegación social. El estado oculto de la última célula se utilizará como entrada para las dos cabeceras de salida del modelo, las cuales se explicarán a continuación.

### 2.2.1. Predicción de velocidad

El objetivo del modelo es predecir la velocidad $\vec{v}_{t+1}$ que el agente debe tomar dadas las observaciones $s_t$ de los últimos $T$ instantes de tiempo, de manera que el agente llegue a destino de una manera socialmente correcta. Para esto, la salida de la LSTM es procesada por 3 capas *fully connected*, donde la última tiene dimensionalidad 2 y corresponde a la velocidad que el agente debe tomar en $x$ e $y$ respectivamente.

### 2.2.2. Predicción del futuro como tarea auxiliar

Además de la rama que busca predecir la velocidad, se agregó una rama extra que tiene la tarea de predecir el cuadro siguiente de profundidad $\hat{D}_{t+1}$. Ésta es añadida como una tarea auxiliar, la cual busca guiar el aprendizaje alentando al modelo a aprender las features necesarias para ser capaz de predecir cómo se verá la escena un instante de tiempo en el futuro. Parte de nuestra hipótesis es que estas *features* ayudarán a que el modelo aprenda relaciones que facilitarán su tarea al incorporar comportamientos sociales y habilidades de navegación en comparación a cuando esta rama del modelo no se encuentra presente. Esto debido a que la habilidad de saber hacia dónde se moverán los objetos y agentes en la escena está estrechamente vinculada a la habilidad de saber cómo actuar frente a ellos.

### 2.2.3. Función de perdida

La perdidas para la predicción de velocidad y del frame de profundidad futuro se definen en (2.3) y (2.4), respectivamente.

$$L_v = \sum \|v_{t+1} - \hat{v}_{t+1}\|^2 w(t) \tag{2.3}$$

$$L_D = \sum \|D_{t+1} - \hat{D}_{t+1}\|^2 w(t) \tag{2.4}$$

$$w(t) = \begin{cases} c, & \text{if } min(p_{\mathbf{x},\mathbf{y},\mathbf{t}}) < \beta \\ 1, & \text{otherwise} \end{cases} \tag{2.5}$$

Debido a que para el modelo es más importante aprender el comportamiento que tienen los agentes cuando otro agente u obstáculo se encuentra próximo a este, se introduce durante el aprendizaje el ponderador $w(t)$ en (5) el cual asigna un peso $c$, donde $c > 1$, a la salida del modelo cuando la cámara de profundidad detecta un objeto a una distancia menor a $\beta$. De esta manera el modelo priorizará aprender comportamientos que involucren obstáculos cercanos por sobre los comportamientos que no involucran obstáculos o donde estos no son la prioridad.

La pérdida final corresponde a la suma ponderada de ambas pérdidas:

$$L = L_v + kL_D \tag{2.6}$$

# CAPÍTULO 3. IMPLEMENTACIÓN Y METODOLOGÍA DE EVALUACIÓN

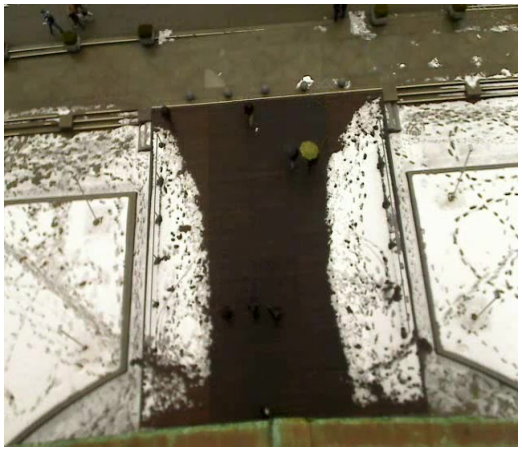
## 3.1. Implementación

El dataset basal utilizado es ETH BIWI Walking Pedestrians dataset (Pellegrini y cols., 2009). Este dataset corresponde a una toma en vista de pájaro de dos escenarios distintos, eth y hotel (ver figura 3.1), en los cuales se incluye la posición $(x_t^i, y_t^i)$ y la velocidad $(vx_t^i, vy_t^i)$ para cada agente $i \in I$ en cada instante de tiempo $t$. En total, el dataset presenta 787 agentes, cada uno con una trayectoria. Estas anotaciones fueron registradas a 2.5 [fps] dando un total de 29.662 imágenes anotadas y se encuentran disponibles para uso público. En este trabajo los datos son replicados en nuestro ambiente virtual, generando así los nuevos datos utilizados para el entrenamiento de nuestro modelo.

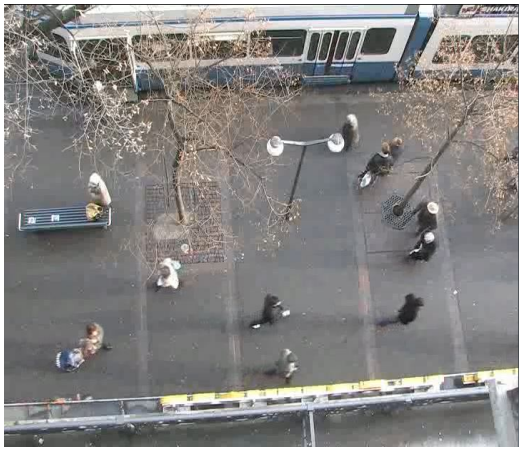
Por simplicidad, se hicieron las siguientes modificaciones para el modelado de los mapas virtuales que se pueden ver en la figura 3.2. Estas son: para el mapa eth, se reemplazó la nieve presente en los costados de la escena por muros. Ello con la justificación de que, como nadie camina por la nieve en el vídeo, ésta actúa de la misma manera que lo haría un muro en un pasillo; para el mapa hotel las formas complejas de las bancas, árboles y postes se simplificaron en la forma de prismas con un volumen similar al obstáculo que buscan representar.

En forma complementaria, se generaron 6000 trayectorias sintéticas en ambos mapas de ETH BIWI Walking Pedestrian dataset con agentes tanto dinámicos como estáticos mediante la técnica de data augmentation. Para los agentes dinámicos, estos se instanciaron con una velocidad aleatoria entre 0.3 y $0.7[m/s]$, y sus trayectorias consisten en alcanzar de 3 a 8 sub-metas designados aleatoriamente dentro en una superficie de 170 y $240[m^2]$ para el mapa hotel y eth respectivamente. El número de agentes por escena fue también alegorizado y va entre 12 a 30 transeúntes.

Para esta investigación, los datos de entrenamiento se generaron en la forma de una imagen de profundidad de 320x240 con un ángulo de visión de $135^\circ$ y $d_{max} = 7[m]$ , en

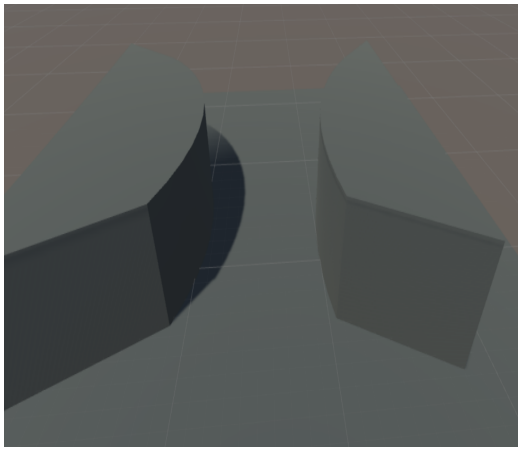


(a) Captura del escenario eth. Fuente: S. Pellegrini

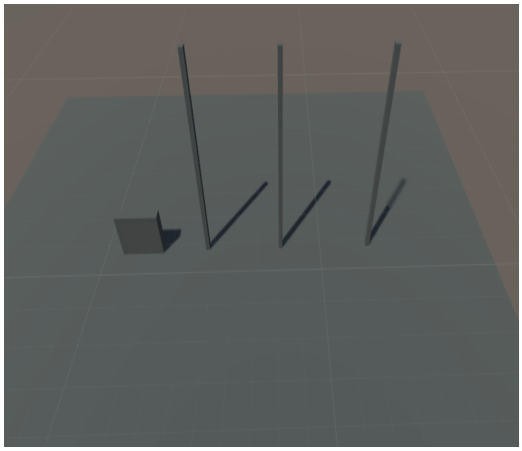


(b) Captura del escenario hotel. Fuente: S. Pellegrini

Figura 3.1. Capturas de ambos mapas de ETH BIWI Walking Pedestrians dataset



(a) Recreación del escenario eth. Fuente: Propia



(b) Recreación del escenario hotel. Fuente: Propia

Figura 3.2. Recreacion de ambos mapas de ETH BIWI Walking Pedestrians dataset

conjunto a la coordenada actual y de destino del agente (ambas en referencia global) y la velocidad que tenía el agente durante el instante de la medición. Todos estos datos son tomados de manera offline en el ambiente virtual con una frecuencia de muestreo de 10 [Hz].

Para entrenar el modelo, los datos se separaron de la siguiente manera. De las 6000 trayectorias sintéticas se separaron 9/10 y 1/10 de manera aleatoria para entrenamiento y evaluación respectivamente. Las 716 trayectorias de BIWI se separaron en la misma proporción también aleatoriamente. Para el modelo se decidió usar $T = 10$, es decir, que éste recibió como entrada las últimas 10 observaciones, lo que corresponde a una memoria de 1 segundo. Durante el entrenamiento se utilizó $c = 2$ como ponderador para las predicciones cuando se encontraba un obstáculo cerca, además de definirse $k = 0{,}1$ para la pérdida de la tarea auxiliar. El modelo DeepSocNav se entrenó, en primer lugar, con las trayectorias sintéticas durante 15 épocas, utilizando un tasa de aprendizaje de 0.001 con Adam como optimizador. Posteriormente se entrenó con los datos reales utilizando la metodología de *fine-tuning* con una tasa de aprendizaje de 0.0001 y Adam por 20 épocas. Se optó por utilizar *fine-tuning* en vez de combinar los datos debido a que la diferencia en la magnitud de los dos tipos de datos tendría como consecuencia que la componente social de los datos reconstruidos se perdiese frente al mayor número de ejemplos no sociales. Para la evaluación, se reemplazó el controlador de movimiento NavMesh de Unity por el modelo a evaluar. Este control está hecho mediante una API, la cual envía las observaciones del agente al modelo corriendo en el fondo, y este retorna la instrucción de velocidad para el agente. Este control es realizado en tiempo real a 10 [Hz].

Durante la evaluación online, el agente debe alcanzar la meta que es una circunferencia de 1.5 metros de radio con el punto final de la trayectoria como centro. Estas trayectorias tienen un largo promedio en el *ground truth* de $9[m]$.

### 3.2. *Baselines*

Para tener un punto comparativo de nuestro modelo, se consideraron las siguientes *baselines*.

- Reciprocal Velocity Obstacles (RVO): RVO es un sistema de navegación en tiempo real que sin utilizar una comunicación directa entre los agentes, asume que todos

los agentes van a reaccionar de una manera similar para evitar colisiones (van den Berg y cols., 2008). Este sistema viene incorporado en el paquete de Unity NavMesh. Como algoritmo de pathfinding, este paquete usa A*.

- Social Force Model (SFM): SFM hace uso de fuerzas de repulsión entre agente-agente y agente-obstáculos en conjunto a fuerzas de atracción agente-objetivo para calcular campos potenciales que guían de forma 'social' a un agente (Helbing y Molnar, 1998). Para su implementación se utilizaron los hiperparámetros obtenidos en (Ferrer y cols., 2016) pero solo se utilizaron las fuerzas de agentes y obstáculos presentes en el rango visual del agente.
- SocioSense: SocioSense es un algoritmo de navegación socialmente consciente basado en Aprendizaje Bayesiano y teoría de Rasgos Psicológicos (Bera y cols., 2017). Este modelo combina restricciones dadas por rasgos psicológicos computados y dinámicas del agente para predecir rutas de navegación utilizando Generalized Velocity Obstacles (GVO), donde se utilizan las posiciones dadas de todos los transeúntes en la escena.
- NaviGAN: NaviGAN es un algoritmo de navegación generativo el cual utiliza generative adversarial networks (GAN) para aprender a generar recorridos que buscan optimizar el confort y la naturalidad de estos (Tsai y Oh, 2020). Para esto, este modelo utiliza las posiciones de todos los agentes en la escena para calcular un descriptor de la Fuerza Social ejercida sobre el agente, en paralelo que predice un descriptor de la intención del agente a controlar.

### 3.3. Variantes para el estudio de sensibilidad

Para estudiar el aporte de las distintas partes del modelo, se entrenaron también las siguientes variaciones de DeepSocNav.

- $DeepSocNav_{noAux}$: esta variante corresponde al modelo completo, pero sin la rama de la tarea auxiliar. Al evaluar este modelo se busca medir el aporte de incluir

la tarea auxiliar de predecir la imagen de profundidad en el frame futuro para facilitar el aprendizaje de la tarea principal.

- $DeepSocNav_{T1}$: esta variante utiliza como ventana de tiempo solo la última medición, es decir $T = 1$. Utilizando esta versión del modelo se podrá evaluar que tan útil es tener una memoria de corto plazo para poder realizar la tarea de navegación social.
- $DeepSocNav_{halfPreTrain}$: corresponde al modelo completo pero solo fue pre-entrenado con la mitad de los datos sintéticos antes de ser entrenado con los datos de BIWI. A partir de esta variante, se busca evaluar el impacto del volumen de datos de pre-entrenamiento en el modelo final.
- $DeepSocNav_{noPreTrain}$: al igual que la variación anterior, esta corresponde al modelo completo pero solo fue entrenado por el dataset BIWI. Es decir, no tuvo el pre entrenamiento con los datos sintéticos. Esto con el fin de verificar si el pre-entrenamiento es realmente un aporte positivo que facilita el posterior aprendizaje de comportamientos sociales presentes en el dataset BIWI.

## 3.4. Métricas de evaluación

Para la evaluación se utilizaron las siguientes métricas.

- Social Score: Esta métrica busca evaluar que tan socialmente correcta es la trayectoria seguida. Hall (1964) define los espacios personales desde la antropología como una serie de círculos concéntricos con el individuo al centro. Estas circunferencias demarcan las distancias que las personas usan para interactuar con otros agentes de acuerdo a su relación con ellos. Basado en este trabajo, nosotros definimos el Social Score también como una serie de círculos concéntricos donde cada uno tiene un costo asignado. Si el agente a evaluar atraviesa la circunferencia de otro agente, se le otorgará una penalización que corresponde al costo más elevado entre las circunferencias atravesadas. Para esto se definieron tres radios, $r_1$, $r_2$ y

$r_3$, donde $r_1 < r_2 < r_3$, y tienen asignados los costos $c_1$, $c_2$ y $c_3$ respectivamente. Para este trabajo, se definieron los radios como $r_1 = 0{,}5[m]$, $r_2 = 0{,}75[m]$ y $r_3 = 1{,}0[m]$, y los costos como $c_1 = 1$, $c_2 = 0{,}5$ y $c_3 = 0{,}1$. De esta manera, un modelo que mantiene una distancia prudente del resto de las personas, tendrá un Social Score menor en comparación a un agente que se mueve demasiado cerca del resto de transeúntes.

- Average Distance Error (ADE): Se calcula el área entre la trayectoria seguida por el modelo a evaluar y la trayectoria del ground truth (GT). Ésta luego se divide por el largo de la trayectoria del GT, quedando así la diferencia promedio entre ambas trayectorias. La unidad para esta métrica es metro.
- Final Distance Error (FDE): Medición de la diferencia entre el punto final de la trayectoria tomada por el modelo y el GT en metros.
- Partial Distance Accuracy (PDA): Tomando en consideración que el paso promedio del ser humano es de aproximadamente 0.8 metros, para cualquier instante de tiempo se puede considerar como exitosa la posición de una trayectoria si la diferencia de esta con la del GT es menor a la del paso promedio. Tal como lo hacen Bera y cols. (2017), nosotros mediremos el porcentaje de éxitos en el instante de tiempo 1[s] y 5[s] a partir del inicio de la trayectoria.
- Colisiones: Porcentaje de colisiones con otros agentes.
- Éxitos: Porcentaje de trayectorias exitosas donde el agente logra llegar al punto objetivo.

# CAPÍTULO 4. RESULTADOS EXPERIMENTALES Y ANÁLISIS

## 4.1. Resultados experimentales

Cuadro 4.1. Resultados en métricas sociales en evaluación online.

| Modelo | Social Score | Colisiones | Éxito |
|---|---|---|---|
| GT | 0.034 | - | - |
| RVO | 0.067 | - | **1** |
| SFM | 0.054 | 0.037 | **1** |
| DeepSocNav | **0.040** | **0.018** | **1** |

Cuadro 4.2. Resultados en métricas de distancia con respecto al GT en evaluación online.

| Modelo | ADE [m] | FDE [m] | PDA 1[s] | PDA 5[s] |
|---|---|---|---|---|
| RVO | 0.203 | 0.203 | **1** | 0.069 |
| SFM | **0.202** | **0.164** | **1** | 0.46 |
| SocioSense (Bera y cols., 2017) | - | - | 0.81 | **0.73** |
| NaviGAN (Tsai, 2019) | 0.430 | 0.740 | - | - |
| DeepSocNav | 0.249 | 0.344 | **1** | 0.23 |

## 4.2. Análisis de los resultados

De la tabla 4.1 que contiene las métricas con carácter social, es posible notar que DeepSocNav pudo imitar el aspecto social de las personas de mejor manera que todos los otros algoritmos, lo que se ve representado con el mejor rendimiento en Social Score y colisiones. En termino de éxitos, DeepSocNav tuvo un puntaje perfecto. Es importante mencionar que aunque SFM y RVO son algoritmos heurísticos basados en reglas predefinidas, la componente social de las reglas de SFM lo ayudan a tener un Social Score

Cuadro 4.3. Resultados de sensibilidad en evaluación online.

| Modelo | Social Score | Colisiones | ADE [m] | FDE [m] | PDA (1s) | PDA (5s) |
|---|---|---|---|---|---|---|
| DeepSocNav | 0.040 | **0.018** | 0.249 | 0.344 | **1** | 0.23 |
| $DeepSocNav_{noAux}$ | 0.042 | 0.037 | **0.244** | **0.201** | **1** | 0.16 |
| $DeepSocNav_{T1}$ | 0.051 | **0.018** | 0.322 | 0.523 | **1** | 0.18 |
| $DeepSocNav_{halfPreTrain}$ | **0.038** | 0.056 | 0.274 | 0.345 | **1** | 0.31 |
| $DeepSocNav_{noPreTrain}$ | 0.047 | 0.094 | 0.337 | 0.415 | **1** | **0.48** |

considerablemente mejor que RVO. Respecto a las colisiones, SFM a pesar de que tuvo el doble de colisiones que DeepSocNav, de todas formas tuvo un buen rendimiento colisionando en menos del 4 % de las trayectorias. RVO al actuar en modo de oráculo dentro del ambiente simulado, es incapaz de tener colisiones. A causa de ello no se considero esta métrica en sus resultados.

A pesar de que SFM tuvo un peor rendimiento en Social Score que DeepSocNav, se puede observar en la tabla 4.2 -que contiene los resultados con carácter de distancia respecto al GT- que este fue capaz de generar las trayectorias más cercanas al GT en términos de camino recorrido al tener los valores más bajos en ADE y FDE. Es por este motivo que modelos basados en SFM siguen siendo considerados como alternativas viables hasta el día de hoy. DeepSocNav no se queda atrás (obteniendo un rendimiento similar en ADE y FDE), lo que no es menor al considerar que DeepSocNav no utiliza las coordenadas de los otros agentes para navegar como lo hace SFM, sino que debe valerse de extraer información a partir de la vista en primera persona para navegar y evadir obstáculos. Creemos que es esta misma vista en primera persona lo que le permite a nuestro modelo generar relaciones más ricas en información para obtener un rendimiento superior a NaviGAN, el cual también está basado en modelos de aprendizaje profundo supervisado, pero que utiliza información de los otros transeúntes sólo a nivel de coordenadas. A pesar de ser el único método con un PDA menor a 1 en el instante 1 [s], SocioSense fue el que tuvo el mejor rendimiento en el instante 5 [s]. De aquí se puede extraer que aunque tienda a desviarse más al inicio de la trayectoria que los otros métodos, suele tener una trayectoria más fiel a la del GT en el largo plazo, lugar donde DeepSocNav presenta su mayor falencia.

### 4.2.1. Análisis de sensibilidad

Al analizar el rendimiento que tuvieron las diferentes versiones de DeepSocNav podemos obtener una noción del aporte de cada parte en el modelo completo. Estos resultados de sensibilidad pueden verse en la tabla 4.3. Es posible notar que aunque el modelo completo haya tenido las mejores métricas sociales (Social Score, colisiones y porcentaje de éxitos), $DeepSocNav_{noAux}$ tuvo el mejor rendimiento en casi todas las métricas de distancia. Esto significa que aunque la tarea auxiliar -es decir predecir como se verá la escena en el instante siguiente de tiempo- ayuda a evitar colisiones y a tener un mejor desempeño social, existe un compromiso respecto a la capacidad de generar trayectorias más similares a las del GT.

Por otro lado, la versión de DeepSocNav con el peor rendimiento en las métricas sociales y similitud a las trayectorias originales fue $DeepSocNav_{T1}$. Aquello acredita la importancia de tener acceso a una pequeña memoria para poder incorporar una navegación más cercana a la humana. Cabe mencionar que al mantener la tarea auxiliar, la componente reactiva de esta variación del modelo sigue siendo equivalente a la del modelo completo, ya que estos dos comparten el mismo rendimiento en colisiones.

Finalmente $DeepSocNav_{noPreTrain}$ fue la variante con el mayor número de colisiones. Esto puede atribuirse a que, aunque el dataset sintético no genera trayectorias con normas sociales, estas sirven de ejemplo sobre cómo transitar de un punto a otro sin colisionar con otros agentes. Esto puede observarse de manera mas gradual al analizar las diferencias entre DeepSocNav y $DeepSocNav_{halfPreTrain}$, donde $DeepSocNav_{halfPreTrain}$ al haber visto solo la mitad de los ejemplos sintéticos tuvo casi el mismo Social Score que $DeepSocNav$ pero mas del doble de colisiones. Por otro lado, la diferencia que tuvieron estos dos modelos en la métrica ADE puede explicarse al analizar la naturaleza del dataset BIWI utilizado. Este es un ambiente exterior donde los agentes, para transitar desde el origen a su destino, siguen generalmente una linea recta solo desviándose para no colisionar con otros agentes. Este tipo de navegación es exactamente la misma utilizada en el generador de trayectorias sintéticas, a excepción del carácter social que tienen las personas

reales. De esta forma al aumentar los datos sintéticos para el pre-entrenamiento el modelo aprende una base de navegación más robusta para ambientes exteriores como los presentes en BIWI.

# CAPÍTULO 5. CONCLUSIÓN Y TRABAJO FUTURO

## 5.1. Observaciones generales

Las redes neuronales profundas son una herramienta útil para extraer *features* y tomar decisiones *end-to-end* sin la necesidad de que los desarrolladores tengan que diseñar manualmente los algoritmos de toma de decisiones. Esto es de especial utilidad en la navegación social debido a que resulta difícil describir manualmente el proceso mental que toman los peatones al transitar en ambientes públicos, lo que solo se dificulta cuando aumentamos el número de agentes externos -como es el caso de los ambientes concurridos-. Modelos como DeepSocNav son capaces de aprovechar capacidades similares a las que tiene un humano en la forma de una vista en primera persona, una memoria de corto plazo (LSTM) y la capacidad de intentar predecir cómo se verá el futuro inmediato de acuerdo a ésta como tarea auxiliar. Éstas le permitieron al modelo completo tener un comportamiento social más cercano a los transeúntes en el *ground truth* en comparación a las *baselines*, tal como se observó en los resultados a través del social score, el número de colisiones y porcentaje de éxitos. Por otro lado, a pesar de que DeepSocNav no haya obtenido el mejor rendimiento en las métricas de distancia, la diferencia del error promedio -o ADE-, obtenido entre el mejor resultado y DeepSocNav, fue de tan solo 0.047 metros, diferencia despreciable al considerar que las trayectorias tienen un largo promedio de 9 metros. De esta forma, a través de DeepSocNav logramos cumplir el objetivo de desarrollar un algoritmo de navegación social basado en un *input* de imágenes de profundidad, el cual está a la altura del estado del arte. A la vez, poder utilizar ambientes simulados para recrear datos reales permite desarrollar modelos con entradas más cercanas a las que tiene un humano, sin la necesidad de costear la toma de datos nuevos en el mundo real. En particular, el desarrollo de éste ambiente simulado nos permitió cumplir el segundo objetivo de esta investigación, que consistía en poder extraer imágenes en primera persona a partir de datos en vista de pájaro. A través de este ambiente simulado, el diseñador del modelo tiene un mayor nivel de control sobre los datos y la escena, además de otorgarle la posibilidad de

generar datos nuevos que, incluso sin tener un componente social, le permiten al modelo aprender una base de *pathfinding* y *obstacle avoidance* sobre los cuales se podrá construir la navegación social, tal como quedó demostrado en los resultados.

## 5.2. Trabajo futuro

Habiéndose comprobado la utilidad de utilizar una vista en primera persona como entrada de un modelo de aprendizaje profundo para navegación social, y teniendo la capacidad de obtener datos de este tipo a través de la replicación en un ambiente simulado, queda como trabajo futuro la implementación de un modelo más avanzado, que incluya herramientas como atención o GANs para poder extraer relaciones más complejas de las que puede aprender una LSTM.

# BIBLIOGRAFÍA


Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., y Savarese, S. (2016). Social lstm: Human trajectory prediction in crowded spaces. En *2016 ieee conference on computer vision and pattern recognition (cvpr)* (p. 961-971).

Alahi, A., Ramanathan, V., y Fei-Fei, L. (2014). Socially-aware large-scale crowd forecasting. En *2014 ieee conference on computer vision and pattern recognition* (p. 2211-2218).

Bera, A., Randhavane, T., Prinja, R., y Manocha, D. (2017). Sociosense: Robot navigation amongst pedestrians with social and psychological constraints. En *2017 ieee/rsj international conference on intelligent robots and systems (iros)* (p. 7018-7025).

Chen, Y. F., Everett, M., Liu, M., y How, J. P. (2017). Socially aware motion planning with deep reinforcement learning. En *2017 ieee/rsj international conference on intelligent robots and systems (iros)* (p. 1343-1350).

Chen, Y. F., Liu, M., Everett, M., y How, J. P. (2016). Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. *CoRR*, *abs/1609.07845*. Descargado de http://arxiv.org/abs/1609.07845

Fahad, M., Chen, Z., y Guo, Y. (2018). Learning how pedestrians navigate: A deep inverse reinforcement learning approach. En *2018 ieee/rsj international conference on intelligent robots and systems (iros)* (p. 819-826).

Ferrer, G., Zulueta, A., Cotarelo, F., y Sanfeliu, A. (2016, julio). Robot social-aware navigation framework to accompany people walking side-by-side. *Autonomous Robots*, *41*. doi: 10.1007/s10514-016-9584-y

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., y Alahi, A. (2018). Social GAN: socially acceptable trajectories with generative adversarial networks. *CoRR*, *abs/1803.10892*. Descargado de http://arxiv.org/abs/1803.10892

Hall, E. T. (1964). Silent assumptions in social communication. *American Anthropologist*, *66*(6), 154-163.

Hamandi, M., D'Arcy, M., y Fazli, P. (2019). Deepmotion: Learning to navigate like

humans. En *2019 28th ieee international conference on robot and human interactive communication (ro-man)* (p. 1-7).

Helbing, D., y Molnar, P. (1998, mayo). Social force model for pedestrian dynamics. *Physical Review E*, *51*. doi: 10.1103/PhysRevE.51.4282

Hochreiter, S., y Schmidhuber, J. (1997, diciembre). Long short-term memory. *Neural computation*, *9*, 1735-80. doi: 10.1162/neco.1997.9.8.1735

Joo, H., Simon, T., Cikara, M., y Sheikh, Y. (2019). Towards social artificial intelligence: Nonverbal social signal prediction in A triadic interaction. *CoRR*, *abs/1906.04158*. Descargado de http://arxiv.org/abs/1906.04158

Kitani, K., Ziebart, B., Bagnell, J., y Hebert, M. (2012, octubre). Activity forecasting. En *Eccv (4)* (p. 201-214). doi: 10.1007/978-3-642-33765-9_15

Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I. D., Rezatofighi, S. H., y Savarese, S. (2019). Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *CoRR*, *abs/1907.03395*. Descargado de http://arxiv.org/abs/1907.03395

Kretzschmar, H., Spies, M., Sprunk, C., y Burgard, W. (2016). Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, *35*, 1289 - 1307.

Kuderer, M., Kretzschmar, H., Sprunk, C., y Burgard, W. (2012, julio). Feature-based prediction of trajectories for socially compliant navigation. En *Robotics: Science and systems*. doi: 10.15607/RSS.2012.VIII.025

Lai, C.-J., y Tsai, C.-P. (2018, marzo). Design of introducing service robot into catering services. En *Icsrt '18* (p. 62-66). doi: 10.1145/3208833.3208837

Leal-Taixé, L., Pons-Moll, G., y Rosenhahn, B. (2011). Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. En *2011 ieee international conference on computer vision workshops (iccv workshops)* (p. 120-127).

Lerner, A., Chrysanthou, Y., y Lischinski, D. (2007, septiembre). Crowds by example. *Comput. Graph. Forum*, *26*, 655-664. doi: 10.1111/j.1467-8659.2007.01089.x

Luber, M., Stork, J. A., Tipaldi, G. D., y Arras, K. O. (2010). People tracking with human motion predictions from social forces. En *2010 ieee international conference on robotics and automation* (p. 464-469).

Pellegrini, S., Ess, A., Schindler, K., y van Gool, L. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. En *2009 ieee 12th international conference on computer vision* (p. 261-268).

Pellegrini, S., Ess, A., y Van Gool, L. (2010). Improving data association by joint modeling of pedestrian trajectories and groupings. En K. Daniilidis, P. Maragos, y N. Paragios (Eds.), *Computer vision – eccv 2010* (pp. 452–465). Berlin, Heidelberg: Springer Berlin Heidelberg.

Robicquet, A., Sadeghian, A., Alahi, A., y Savarese, S. (2016, octubre). Learning social etiquette: Human trajectory understanding in crowded scenes. En *Computer vision – eccv 2016* (Vol. 9912, p. 549-565). doi: 10.1007/978-3-319-46484-8_33

Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., y Savarese, S. (2018). Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. *CoRR*, *abs/1806.01482*. Descargado de http://arxiv.org/abs/1806.01482

Sadeghian, A., Legros, F., Voisin, M., Vesel, R., Alahi, A., y Savarese, S. (2017). Car-net: Clairvoyant attentive recurrent network. *CoRR*, *abs/1711.10061*. Descargado de http://arxiv.org/abs/1711.10061

Siegwart, R., Nourbakhsh, I., y Scaramuzza, D. (2004). Introduction to autonomous mobile robots..

Tai, L., Zhang, J., Liu, M., y Burgard, W. (2017). Socially-compliant navigation through raw depth inputs with generative adversarial imitation learning. *CoRR*, *abs/1710.02543*. Descargado de http://arxiv.org/abs/1710.02543

Technologies, U. (2018). Unity user manual [Manual de software informático]. Descargado de https://docs.unity3d.com/Manual/index.html

Trautman, P., y Krause, A. (2010). Unfreezing the robot: Navigation in dense, interacting crowds. En *2010 ieee/rsj international conference on intelligent robots and systems* (p. 797-803).

Treuille, A., Cooper, S., y Popović, Z. (2006, julio). Continuum crowds. *ACM Trans. Graph.*, *25*(3), 1160–1168. Descargado de https://doi.org/10.1145/1141911.1142008 doi: 10.1145/1141911.1142008

Triebel, R., Arras, K., Alami, R., Beyer, L., Breuers, S., Chatila, R., ... Zhang, L. (2016, marzo). Spencer: A socially aware service robot for passenger guidance and help in busy airports. En (Vol. 113, p. 607-622). Springer. doi: 10.1007/978-3-319-27702-8_40

Tsai, C.-E. (2019). *A generative approach for socially compliant navigation* (Tesis de Master no publicada). Pittsburgh, PA.

Tsai, C.-E., y Oh, J. (2020). *Navigan: A generative approach for socially compliant navigation.*

van den Berg, J., Ming Lin, y Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. En *2008 ieee international conference on robotics and automation* (p. 1928-1935).

Vemula, A., Mülling, K., y Oh, J. (2017). Social attention: Modeling attention in human crowds. *CoRR*, *abs/1710.04689*. Descargado de http://arxiv.org/abs/1710.04689

The wisdom of crowds.

. Descargado de https://www.economist.com/christmas-specials/2011/12/17/the-wisdom-of-crowds

Yamaguchi, K., Berg, A. C., Ortiz, L. E., y Berg, T. L. (2011). Who are you with and where are you going? En *Cvpr 2011* (p. 1345-1352).

Zhong, L., y Verma, R. (2019). robot rooms": How guests use and perceive hotel robots.

# APÉNDICE

# A. PAPER CONFERENCIA

A continuación se presenta el paper con los mismos contenidos de esta tesis, Deep-SocNav: Social navigation by imitating human behaviors, enviado a la conferencia IEEE International Conference on Robotics and Automation - ICRA 2021.

# DeepSocNav: Social navigation by imitating human behaviors

Juan Pablo de Vicente and Alvaro Soto[1]

*Abstract*— **Current 2D representations and coordinate based interactions for social navigation leave aside precious relationships and visual cues that could only be captured through a first-person view of the scene. This limits the performance of data-driven social navigation models. In this work, we propose to exploit the power of current game engines, such as Unity, to transform pre-existing bird's-eye view datasets into a first-person view, in particular, a depth view. Furthermore, we show the benefits of using synthetic data from a game engine to pre-train a navigational model. To test our ideas, we present DeepSocNav, a deep learning model that takes advantage of the transformed and synthetic datasets, as well as, a self-supervised strategy that is included as an auxiliary task. This task consists of anticipating the next depth frame that the agent will face. Our experiments show the benefits of the proposed model that is able to outperform relevant baselines in terms of social navigation scores.**

## I. Introduction

In recent times there has been an increasing diversification of the tasks performed by robots. In particular, there are new applications in natural environments where a robot needs to interact with humans. This is the case of service robots that can perform tasks such as guidance in airports [1], food delivery [2], or room attention in a hotel [3].

A desirable skill to operate in a human inhabited environment is to exercise social navigation, i.e., navigate in a way that does not hinder other humans present in the environment. Due to this, traditional navigation techniques based on algorithms such as A* or potential field are not applicable [4]. This is because they do not take into consideration people's intrinsic social behaviors.

Previous works have tried to solve this problem. In particular, previous attempts can be classified into two main groups: model-driven and data-driven approaches. On the one hand, model-driven approaches are characterized by an explicit use of relevant relationships and rules that guide navigational behaviors in a social context. Examples of model-driven approaches for social navigation are [5], [6], [7], [8], [9], [10], [11], [12], [13], which mainly use the so-called social force [14] to model the social navigation of passers-by. As a major limitation, these approaches lack enough flexibility to handle unexpected or complex situations.

On the other hand, data-driven approaches seek to learn social behaviors by extracting knowledge from a large number of examples. This has the advantage to automatically infer complex relationships from data. Human behavior is highly stochastic [15], [16] and varies depending on factors such as culture [17]. As a consequence, the increased flexibility of data-driven techniques usually leads to models that outperform the model-driven counterparts [18], [15], [19], [20], [21].

A relevant complexity to implement a data-driven approach for social navigation is the need for a large volume of data. Furthermore, the type of data collected plays a significant role in the quality of the final solution. In terms of social navigation, currently most public datasets consist of third-person recordings, usually in bird's-eye view. This means that models that directly use this data are limited to a 2D representation of the scene, which leaves aside useful relationships that can be captured through 3D observations. A possible way to alleviate this problem is to record and tag first-person navigational data. Unfortunately, this approach does not scale properly, mainly due to the complexity of measuring the velocity and position of the first-person agent and the cost associated to capture a sizable amount of data.

This work helps to overcome this limitation by developing a virtual environment that allows us to transform the abundant bird's-eye view data to first-person views of all agents present in a giving scene. The simulated environment also features realistic human models and animations, increasing the information available to implement social navigational behaviors. Furthermore, it is also possible to include artificial agents to enrich the original data. Motions of these artificial agents do not ensure social behaviors but, as we show in this work, are useful to pre-train a model to acquire basic navigation skills.

In addition with the simulated environment, we also present DeepSocNav, a supervised learning model that seeks to imitate the social behaviors of real humans in crowded environments, using a first-person depth view. This model is capable of learning to replicate the navigational patterns of the agents present in the data. In this way, the model is encouraged to develop a social policy closer to what people do. Furthermore, to guide learning, DeepSocNav also learns the auxiliary task of predicting what the scene will look in the next time step. This fosters the model to acquire a notion of space and agents in its visual field, thus improving its social performance and its ability to avoid collisions.

In summary, the main contributions of this paper are:

- We present a method that allows us to take advantage of the abundant quantity of data in bird's-eye view to generate data under a first-person view of the scene. In addition, this virtual environment allows us to include virtual agents to generate new synthetic data.
- We present DeepSocNav, a supervised learning model that aims to imitate the social behavior of humans in

*This work was partially funded by FONDECYT grant 1181739

[1]J. de Vicente and A. Soto, Dept. of Computer Science, Pontificia Universidad Catolica de Chile, (jpdevicente@uc.cl, asoto@ing.puc.cl)

crowded environments using first-person depth views generated by a simulated environment. As an auxiliary task, this model also includes a self-supervised learning strategy that consists of anticipating its next view of the environment.

We support our contributions with suitable experiments to evaluate the main components of our model, conducting an ablation analysis to demonstrate its benefits.

## II. Previous work

In [14], Helbing and Molnar present a seminal work that guides most of the initial methods devoted to social navigation. Using the idea of a social force, social behaviors are presented as a force of attraction or repulsion according to a potential field generated by the pedestrians present in the scene. Leal-Taixé et al. [6] use social force to improve the performance of people's trackers in crowded environments. Luber et al. [8] model the dynamics of people using a social force to predict their motions. Pellegrini et al. [9] demonstrate good performance in the task of person tracking by combining a social force in conjunction with knowledge of the underlying scene even in conditions of heavy occlusion. Treuille et al. [12] use a potential field approach based on a social force to implement obstacle avoidance behaviors. All the works above follow a model-driven approach where relevant rules and representations are manually defined. Unfortunately, as the complexity of the task increases, the limitations of a hand-based modeling begin to appear, generating problems such as the Freezing Robot Problem [11].

Lately data-driven approaches have become the usual technique to provide with social navigation skills to a mobile robot. One of the first attempts is presented by Kuderer et al. [22]. They make use of Inverse Reinforcement Learning (IRL) and the principle of maximum entropy to learn features that capture relevant aspects of human navigation. At the same time, Kitani et al. [23] combine the use of IRL and semantic knowledge of the scene to include the effect of the environment on people's navigational decisions. Later, Fahad et al. [19] present an IRL model capable of generating trajectories that imitate human-like social behaviors using a large volume of data and social affinity maps (SAM). An operational requirement of IRL is to learn first a suitable reward function. This function is then used to learn an action policy that implements social navigational behaviors. A closely related line of research is deep reinforcement learning (DRL), which needs rewards and penalizers designed by hand to learn a policy. An example of this is found in Chen et al. [15], where they design penalizers around what the robot should not do, in contrast to rewarding desired behaviors, to obtain a social navigation behavior. In contrast to IRL and DRL, in this work, we propose an end-to-end model that directly learns to predict robot actions from data.

There are also works based on supervised learning. These do not require features or rewards designed by hand. Alahi et al. [24] make use of supervised learning and recurrent networks, specifically LSTMs [25], to predict social trajectories of relevant agents present in the scene. Vemula et al. [26] use a LSTM with attention on a graphical representation of passers-by so that the model learns that not all passers-by are equally important to predict social trajectories. In [24] and [26], agents are represented as coordinates on a 2D map, while Sadeghian et al. [27] propose using bird's-eye view images to predict future trajectories using a CNN model with attention and a LSTM.

In a parallel line of research, works such as those by Gupta et al. [28], Kosaraju et al. [29], Sadeghian et al. [30], and Tai et al. [31], seek to predict social trajectories using Generative Adversarial Networks (GAN). As an example, Hamandi et al. [21] simulate LiDAR scans from tagged data in bird's-eye view to train the navigation of a robot, learning the social behaviors present in the dataset. In this research we go a step further and simulate first-person views in order to take advantage of visual cues that cannot be seen in a bird's-eye view.

## III. Proposed method

In this section we describe our proposed method to provide a robot with capabilities for social navigation. Section III.A provides a description of the problem and our main assumptions. Then, Section III.B presents the methodology to transform a bird's-eye view dataset to a first-person view, as well as describing the methodology for the generation of artificial data. Finally, Section III.C describes DeepSocNav, our proposed model to implement social navigation.

### A. Description of the problem and assumptions

When navigating in crowded environments, body posture, gaze, and motions of passers-by provide us with clues about their intentions and potential trajectories. In turn, this allows us to plan a suitable route to avoid collisions or hindering other passers-by. However, due to the complexity associated to capture this type of data, most current approaches for social navigation do not consider this type of information. In effect, most current datasets consists only of 2D data captured from a bird's-eye perspective which limits the scope of the model to consider only position and speed of passers-by. This situation motivates our proposed method that starting from a dataset corresponding to a bird's-eye perspective is able to recreate the corresponding first-person views, in particular a view of depth information.

For simplicity we make the following assumptions about the data:

1) Pedestrians in the original dataset only use their sight and do not communicate between them.
2) Each agent in the dataset aims to reach a specific point in the map before its navigation is completed.

### B. First-person view data generation

For the recreation of the datasets using a first-person view, we develop a simulated environment using Unity, a video game development engine [32]. This environment loads previously designed maps and recreate the trajectory of agents using the data sequence of the agents' position in

time. Furthermore, it can also provide autonomous navigation capabilities using the NavMesh navigation library [33]. This library includes tools to provide path-finding and obstacle avoidance capabilities [34]. Specifically, the simulator moves each agent $i$ from its current position $(x_t^i, y_t^i)$ to next position $(x_{t+1}^i, y_{t+1}^i)$ at time instant $t+1$ until the agent reaches a target position $(G_x^i, G_y^i)$. These displacements follow a realistic walking animation through Unity's animation control asset [33].

Using the simulation, for each agent and time instant $t$, a first-person depth view is stored, together with its current coordinate $(x_t^i, y_t^i)$, target coordinate $(G_x^i, G_y^i)$, and velocity $(vx_t^i, vy_t^i)$. From this data, it is expected that the model can learn the implicit social behaviors underlying the trajectories of the agents in the original dataset. An example of a recreated frame in the simulated environment can be seen in Figure 1. Specifically, Figure 1a corresponds to a frame in the original video and Figure 1b to its re-creation in the simulated environment using a first-person depth view.

### C. Data Augmentation

Besides its bird's-eye view, a second limitation of current pedestrian trajectory datasets is the limited size. In effect, currently the most extensive dataset is The Stanford Drone Dataset [35] that includes a total of 11.2K pedestrians distributed in 6 different scenes. This dataset also include bicycles, skateboards, cars, buses, and golf carts, that in our case are not useful, since we only want to model interactions among people. Purely pedestrian datasets have smaller volumes, in the order of 1K people in total. Furthermore, some of these datasets do not include rich interactions among people, where most pedestrian follow straight trajectories to reach their goals. This may produce model that overlook the need to learn social navigation skills, such as turning and deviating from a straight line in order to avoid collisions.

To alleviate the data scarcity problem, we propose a data augmentation scheme that takes advantage of the autonomous navigation capabilities provided by the library NavMesh included in Unity [33]. Specifically, using Unity, we can add virtual agents to a scene that are instructed to reach randomly generated goal positions. Using this scheme, we can generate an almost unlimited amount of navigational trajectories. As a main limitation, this artificial trajectories do not ensure social behaviors. However, we can use them as a massive source of data to train an initial navigational model that is able to reach target positions while avoiding obstacles. Afterwards, we can fine-tune this model using data that incorporates social behaviors. In other words, while the synthetic data provides our model with basic navigational abilities, a posterior fine-tunning step, using trajectories of real pedestrians, enriches our model with social navigational aptitudes. We describe next the details behind our navigational model.

### D. DeepSocNav

The problem of social navigation is seen as a problem of velocity prediction given a sequence of observations. For each time instant $t$ and agent $i$, we have a measurement of its current position $(x_t^i, y_t^i)$, the coordinates of its goal position $(Gx_t^i, Gy_t^i)$, and a depth image $D_t^i$ corresponding to what the agent sees at time $t$ according to a first-person view. From a sequence of observations $t-T$ to $t$, the agent velocity $(vx_{t+1}^i, vy_t^i)$ is predicted for the following time step $t+1$. In this way the model can be defined as a function $f(s_t, G)$ described as follows:

$$\begin{aligned} f(s_t, G) &= \vec{v}_{t+1},\ t \in [t-T, t] \\ s_t &= (x_t, y_t, D_t),\ t \in [t-T, t] \\ G &= (Gx, Gy) \end{aligned} \tag{1}$$

The positional entries $(x_t, y_t)$ and target $(Gx, Gy)$ are normalized from the largest and smallest value possible among all the maps. Because depth cameras have a limited range, we imitate this limitation by defining the pixel value $p_{\mathbf{x},\mathbf{y},\mathbf{t}}$ in depth images as follows:

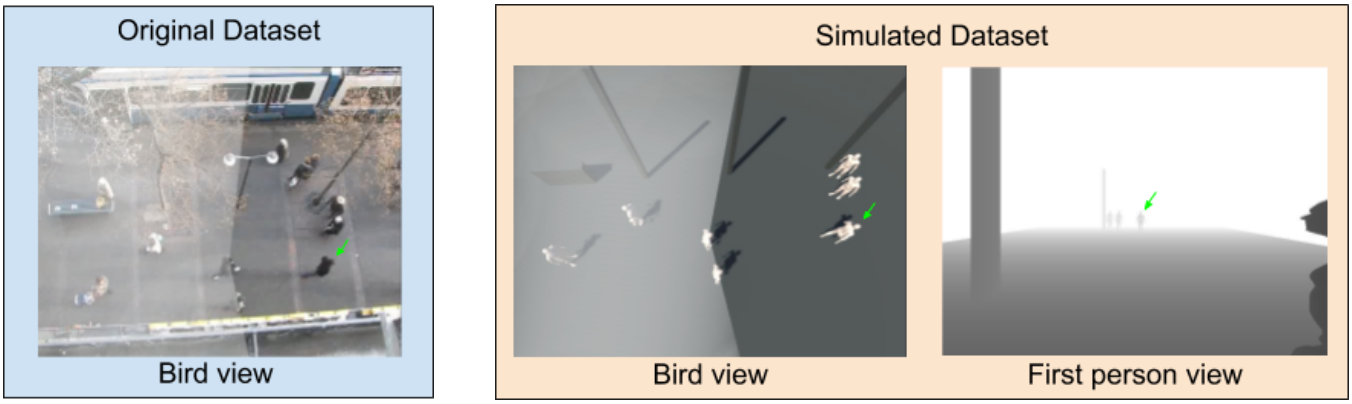
$$p_{\mathbf{x},\mathbf{y},\mathbf{t}} = \begin{cases} d_{\mathbf{x},\mathbf{y},\mathbf{t}}/d_{max}, & \text{if } d_{\mathbf{x},\mathbf{y},\mathbf{t}} < d_{max} \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

where $d_{\mathbf{x},\mathbf{y}}$ corresponds to the distance between the agent's camera and the surface located on the pixel $p_{\mathbf{x},\mathbf{y}}$; $d_{max}$ is the maximum range distance provided by the depth camera.

An overview of the learning architecture behind DeepSocNav can be seen in Figure 2. A dense layer receives the current position and goal location for the agent, increasing its dimensionality to 64. Using this, the model is expected to extract an initial idea of the direction in which the target is located. The depth image is processed through 3 convolutional layers followed by a dense layer of size 128. The goal of these layers is to recognize patterns that would allow the model to identify other agents, obstacles, and empty spaces in its visual field. Both outputs are concatenated for the last T observations to form a tensor of shape [T, 192]. Following [24], we apply a LSTM that receives this as an input. We use two layers of LSTM, each one with a hidden size of 512. It is expected that the LSTM can provide temporal understanding for social navigation. The hidden state of the last cell is used as an input for the two final heads of the model that are explained next.

*1) Velocity prediction:* The goal of this head is to predict the velocity $\vec{v}_{t+1}$ that the agent must take given the observations $s_t$ of the last $T$ instants of time, so that the agent arrives at the destination in a socially correct way. For this, the output of the LSTM is processed by 3 dense layers, where the last one outputs $(vx_{t+1}^i, vy_t^i)$.

*2) Forecasting the future as an auxiliary task:* The goal of this head is to predict the future depth image $\hat{D}_{t+1}$. This is added as an auxiliary task, which seeks to guide learning by encouraging the model to learn the features needed to predict how the scene will look in the close future. Our hypothesis is that the ability to anticipate where objects and agents in a scene will move is closely related to the ability to know how they act.




(a) BIWI dataset: hotel top-down frame.

(b) Simulated top-down view and first-person view.

Fig. 1: a) Representative frame from the real scene present in the original dataset. b) First person depth view of the frame in a) recreated using the simulator.

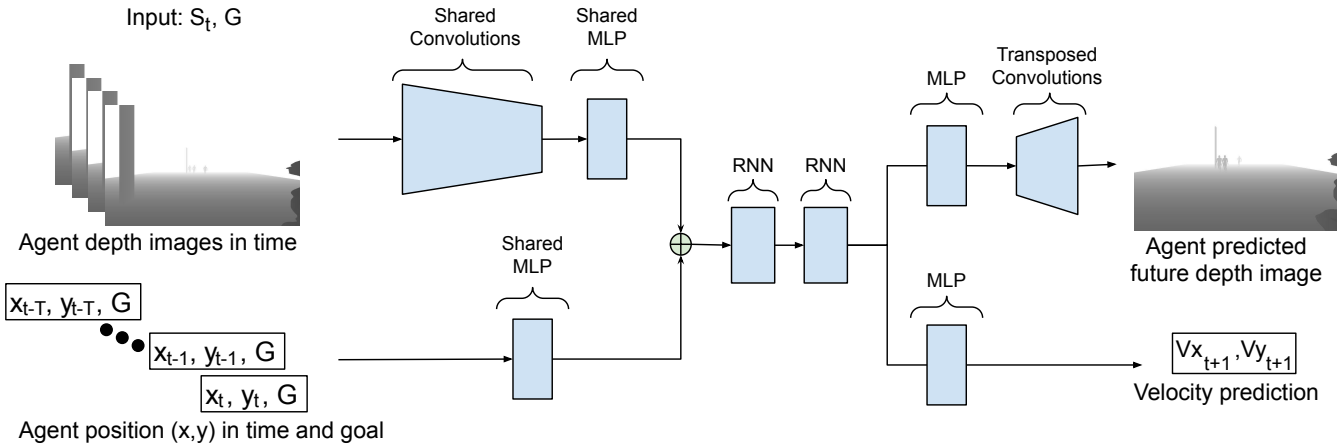



Fig. 2: DeepSocNav overall architecture. This model predicts the velocity of an agent at time $t+1$ given a history of previous observations from steps $t-T$ to $t$.

*3) Loss function:* The losses for the prediction of velocity and depth information are defined in (2) and (3), respectively.

$$L_v = \sum (v_{t+1} - \hat{v}_{t+1})^2 w(t) \tag{2}$$

$$L_D = \sum (D_{t+1} - \hat{D}_{t+1})^2 w(t) \tag{3}$$

$$w(t) = \begin{cases} c_w, & \text{if } min(p_{\mathbf{x},\mathbf{y},\mathbf{t}}) < \beta \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

Because it is more important to pay attention to behaviors when other agents or obstacle are close during training, we introduce the weighting $w(t)$ in (4) which assigns a weight $c$, where $c > 1$, to the output of the model when the depth camera detects an object at a distance less than $\beta$. This is done to prioritize learning behaviors that involve close obstacles over the others.

The final loss corresponds to the weighted sum of both losses:

$$L = L_v + \alpha L_D \tag{5}$$

where due to the varying difference between the magnitude of the two functions, a weighting of $\alpha$ was added, defined as:

$$\alpha = k \frac{L_v}{L_D} \tag{6}$$

so that the model maintains a $k$ relationship between the importance of both tasks during training.

## IV. Experiments

To test our model, we use as a benchmark the ETH BIWI Walking Pedestrians dataset (BIWI) [9]. This dataset corresponds to RGB images from a bird's-eye view of two different scenarios: ETH and Hotel. For each image at time instant $t$, the dataset provides annotations about the position $(x_t^i, y_t^i)$ and velocity $(vx_t^i, vy_t^i)$ of each agent $a \in i$ in the image. In total, the dataset includes 787 agents. Annotations are registered at 2.5 [fps] providing a total of 29,662 annotations.

Using Unity, we recreate the scenes corresponding to the BIWI dataset. Using these synthetic scenes and our proposed

methodology, we transform the RGB images corresponding to bird's-eye view to depth images corresponding to first-person view. As we detail in Section III-B, we use the first-person depth views to train our proposed model: DeepSocNav.

Furthermore, following the data augmentation scheme described in Section III-C, we generate 6,000 trajectories using the synthetic recreation of both scenes: ETH and Hotel. Virtual agents are instantiated using a random velocity between 0.3 and $0.7[m/s]$. Their trajectories consist of 3 to 8 randomly selected sub-goals within a surface of 240 and $170[m^2]$ for ETH and Hotel scenes, respectively. The number of agents per scene is also randomly selected within 12 to 30 passers-by.

In our experiments, we generate depth images using a resolution of 320x240 pixels, a field of view of 135°, and maximum range of $d_{max} = 7[m]$. For each image, we also consider the position and velocity annotations included in the original dataset using a global reference frame. The data is generated offline with a sampling frequency of 10[Hz].

To train DeepSocNav, both synthetic and BIWI trajectories are randomly separated in $\frac{9}{10}$ for training and $\frac{1}{10}$ for evaluation. In our experiments, we use $T = 10$. Also, we use $c = 2$ to weight the predictions when an obstacle is nearby, and $k = 0.1$ to weight the loss function for the auxiliary task.

As we describe in Section III-C, we use the synthetic trajectories to pre-train DeepSocNav. Specifically, we pre-train for 15 epochs, using a learning rate of 0.001 and Adam as the optimizer. Afterwards, DeepSocNav is fine-tuned for 10 epochs using the synthetic depth images corresponding to the BIWI dataset, Adam as the optimizer, and a learning rate of 0.0001. For online evaluation, we replace NavMesh, the default navigation controller in Unity, using instead DeepSocNav to control the navigation policy of the target agent. The controller is implemented at 10 [Hz] through an API. This API sends the agent's observations to the model that runs in the background returning to Unity velocity setups for the target agent. Goal positions follow the BIWI dataset, denoted as a circular area with a radius of $1.5[mts]$. In average, trajectories consist of $9[mts]$ length.

### A. Baselines

In our experiments, we consider the following baselines:

- **Reciprocal Velocity Obstacles (RVO)**. RVO is implemented in NavMesh, the navigational system included in Unity [34]. Specifically, NavMesh uses RVO for obstacle avoidance and A* for path-finding.
- **Social Force Model (SFM)**. SFM makes use of agent-agent and agent-object attraction and repulsive forces. Specifically, SFM calculates potential fields that guide an agent in a social way [14]. In our implementation, we use the hyperparameters described in [36]. Furthermore, we only consider the forces of agents and obstacles present in the visual range of the target agent.
- **SocioSense**. SocioSense is a socially aware navigation algorithm based on Bayesian Learning and Psychological Traits theory [37]. This model combines constraints given by computed and dynamic psychological traits of the agent to predict navigation routes. Specifically, it uses Generalized Velocity Obstacles (GVO), where the given positions of all passers-by in the scene are used.
- **NaviGAN**. NaviGAN is a generative navigation algorithm that uses generative adversarial networks (GANs), to learn how to generate routes that seek to optimize the comfort and naturalness of these routes [38]. To do this, this model uses the positions of all agents in the scene to calculate a descriptor of the social force exerted over the target agent.

### B. DeepSocNav ablation study

To study the contribution of the different parts of the model, the following variations of DeepSocNav are also considered.

- $DeepSocNav_{noAux}$: This variant does not consider the auxiliary task that predicts the depth image for the next frame.
- $DeepSocNav_{T1}$: This variant uses $T = 1$ as the time window.
- $DeepSocNav_{halfPreTrain}$: This variant only uses half of the trajectories for pre-train.
- $DeepSocNav_{noPreTrain}$: This variant does not include pre-training.

| Model | Social Score | Collisions | Success |
|---|---|---|---|
| GT | 0.034 | - | - |
| RVO | 0.067 | - | **1** |
| SFM | 0.054 | 0.037 | **1** |
| DeepSocNav | **0.040** | **0.018** | **1** |

TABLE I: Social results in online evaluation

| Model | ADE [m] | FDE [m] | PDA 1[s] | PDA 5[s] |
|---|---|---|---|---|
| RVO | 0.203 | 0.203 | **1** | 0.069 |
| SFM | **0.202** | **0.164** | **1** | 0.46 |
| SocioSense [37] | - | - | 0.81 | **0.73** |
| NaviGAN [39] | 0.430 | 0.740 | - | - |
| DeepSocNav | 0.249 | 0.344 | **1** | 0.23 |

TABLE II: Distance results relative to GT in online evaluation

### C. Metrics

The following metrics are used for evaluation.

- Social Score: To measure a trajectory's social compliance, we define a series of concentric circles with the individual at the center, each one with an assigned cost. We do this based on the hypothesis of anthropology's personal spaces from Hall [40]. This way, we penalize the target agent using a cost $c_l$ each time he is inside of the personal circle of another agent. For each agent, three radii are defined: $r_1$, $r_2$ and $r_3$, where $r_1 < r_2 < r3$, and they have the costs $c_1$, $c_2$ and $c_3$, respectively. In our experiments, we use the following values: $r_1 =$

| Model | Social Score | Collisions | ADE [m] | FDE [m] | PDA (1s) | PDA (5s) |
|---|---|---|---|---|---|---|
| DeepSocNav | 0.040 | **0.018** | 0.249 | 0.344 | **1** | 0.23 |
| $DeepSocNav_{noAux}$ | 0.042 | 0.037 | **0.244** | **0.201** | **1** | 0.16 |
| $DeepSocNav_{T1}$ | 0.051 | **0.018** | 0.322 | 0.523 | **1** | 0.18 |
| $DeepSocNav_{halfPreTrain}$ | **0.038** | 0.056 | 0.274 | 0.345 | **1** | 0.31 |
| $DeepSocNav_{noPreTrain}$ | 0.047 | 0.094 | 0.337 | 0.415 | **1** | **0.48** |

TABLE III: Ablation results of online evaluation

$0.5[m]$, $r_2 = 0.75[m]$, $r_3 = 1.0[m]$, $c_1 = 1$, $c_2 = 0.5$ and $c_3 = 0.1$.

- Average Distance Error (ADE): The area between the path followed and the ground truth (GT) is divided by the length of the GT to get the average distance error. This is measured in meters.
- Final Distance Error (FDE): Measurement of the difference between the end point of the path taken and the GT in meters.
- Partial Distance Accuracy (PDA): Taking into account that the average human step is approximately 0.8 meters, we consider as a successful trajectory position, if the difference with respect to the GT is less than the average step. Following [37], we measure the percentage of success at time instants 1[s] and 5[s] from the beginning of the trajectory.
- Collisions: Percentage of collisions with other agents.
- Success: Percentage of successful trajectories where the agent manages to reach the target position.

## V. Results and Analysis

Table I shows the performance of DeepSocNav and the baselines. In terms of the social scores (Social Score and Collisions), DeepSocNav outperforms all the baselines. In terms of success score, DeepSocNav has a perfect score. In terms of the distance metrics (ADE and FDE), while DeepSocNav has a competitive performance, SFM is able to generate the most similar trajectories to the GT. However, it is important to note that SFM needs and uses explicit information about the positions of all the other agents, while DeepSocNav only uses a first-person depth view.

We believe that by taking advantage of its first-person depth view, DeepSocNav is able to learn and then infer rich information about the intentions and potential trajectories of other passers-by. This explains its superior performance with respect to state-of-the art techniques such as NaviGAN [38], which corresponds to a deep learning model that uses oracle information from other passers-by at the coordinate level.

### A. Ablation analysis

Table III shows an ablation analysis with respect to the main components behind DeepSocNav. It is possible to note that although the full model achieves the best social scores (Social Score and Collisions), $DeepSocNav_{noAux}$ obtains the best performance on almost all the distance related scores. This means that predicting depth information for the next frame helps to avoid collisions and to improve social navigation, but there is a trade-off in terms of generating trajectories that are similar to the GT.

The version of DeepSocNav with the worst performance overall is $DeepSocNav_{T1}$. This supports the relevance of keeping a small memory to perform correctly in the task. However, by maintaining the auxiliary task, its reactive component is as good as the complete model, as seen in its collisions performance.

Finally, $DeepSocNav_{noPreTrain}$ is the variant with the highest number of collisions. This supports our initial hypothesis that by pre-training the model using simulated data, it is possible to overcome data scarsity issues. In particular, our pre-training strategy is able to generate an initial navigation scheme with obstacle avoidance and goal-reaching capabilities. This can also be seen by considering our full model and $DeepSocNav_{halfPreTrain}$, where $DeepSocNav_{halfPreTrain}$ has almost the same Social Score than the full model, but the number of collisions is more than double.

## VI. Conclusions

In this work, we show the advantage of using the power of current game engines, such as Unity, to improve the generation of data to train social navigation agents. In particular, we show the advantage of exploiting a method to extract first-person view recordings of passers-by from bird-view datasets. Similarly, we show the advantage of using simulated data from a virtual environment to pre-train a model to obtain an initial navigation scheme that include obstacle avoidance and goal-reaching capabilities.

In particular, we demonstrate the advantages of the information-rich first-person view by training DeepSocNav, an LSTM-based model capable of navigating through a crowded environment. Our results validate the impact of our proposed strategy in terms of social navigation by outperforming all baselines considered in this work. We also demonstrate the relevance of using a short-term memory of previous views, as well as, a prediction of the next depth frame, which is included as an auxiliary task. We believe that the use of this type of auxiliary tasks and self-supervised learning strategies might play an important role to improve current social navigation models.

## References


[1] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, H. Hung, O. Ramírez, M. Joosse, H. Khambhaita, T. Kucner, B. Leibe, A. Lilienthal, T. Linder, M. Lohse, and L. Zhang, *SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports*. Springer, Mar. 2016, vol. 113, pp. 607–622.

[2] C.-J. Lai and C.-P. Tsai, "Design of introducing service robot into catering services," in *ICSRT '18*, Mar. 2018, pp. 62–66.

[3] L. Zhong and R. Verma, ""robot rooms": How guests use and perceive hotel robots," 2019.

[4] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, "Introduction to autonomous mobile robots," 2004.

[5] A. Alahi, V. Ramanathan, and L. Fei-Fei, "Socially-aware large-scale crowd forecasting," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2211–2218.

[6] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn, "Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 120–127.

[7] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, pp. 655–664, Sept. 2007.

[8] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 464–469.

[9] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 261–268.

[10] S. Pellegrini, A. Ess, and L. Van Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 452–465.

[11] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 797–803.

[12] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," *ACM Trans. Graph.*, vol. 25, no. 3, p. 1160–1168, July 2006. [Online]. Available: https://doi.org/10.1145/1141911.1142008

[13] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" in *CVPR 2011*, 2011, pp. 1345–1352.

[14] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, May 1998.

[15] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1343–1350.

[16] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, pp. 1289 – 1307, 2016.

[17] *The wisdom of crowds*, The Economist, Dec. 2011. [Online]. Available: https://www.economist.com/christmas-specials/2011/12/17/the-wisdom-of-crowds

[18] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," *CoRR*, vol. abs/1609.07845, 2016. [Online]. Available: http://arxiv.org/abs/1609.07845

[19] M. Fahad, Z. Chen, and Y. Guo, "Learning how pedestrians navigate: A deep inverse reinforcement learning approach," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 819–826.

[20] H. Joo, T. Simon, M. Cikara, and Y. Sheikh, "Towards social artificial intelligence: Nonverbal social signal prediction in A triadic interaction," *CoRR*, vol. abs/1906.04158, 2019. [Online]. Available: http://arxiv.org/abs/1906.04158

[21] M. Hamandi, M. D'Arcy, and P. Fazli, "Deepmotion: Learning to navigate like humans," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019, pp. 1–7.

[22] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," in *Robotics: Science and Systems*, July 2012.

[23] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert, "Activity forecasting," in *ECCV (4)*, Oct. 2012, pp. 201–214.

[24] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997.

[26] A. Vemula, K. Mülling, and J. Oh, "Social attention: Modeling attention in human crowds," *CoRR*, vol. abs/1710.04689, 2017. [Online]. Available: http://arxiv.org/abs/1710.04689

[27] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese, "Car-net: Clairvoyant attentive recurrent network," *CoRR*, vol. abs/1711.10061, 2017. [Online]. Available: http://arxiv.org/abs/1711.10061

[28] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: socially acceptable trajectories with generative adversarial networks," *CoRR*, vol. abs/1803.10892, 2018. [Online]. Available: http://arxiv.org/abs/1803.10892

[29] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. D. Reid, S. H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," *CoRR*, vol. abs/1907.03395, 2019. [Online]. Available: http://arxiv.org/abs/1907.03395

[30] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, and S. Savarese, "Sophie: An attentive GAN for predicting paths compliant to social and physical constraints," *CoRR*, vol. abs/1806.01482, 2018. [Online]. Available: http://arxiv.org/abs/1806.01482

[31] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially-compliant navigation through raw depth inputs with generative adversarial imitation learning," *CoRR*, vol. abs/1710.02543, 2017. [Online]. Available: http://arxiv.org/abs/1710.02543

[32] Unity Technologies, "Unity," May 2018, 2018.1.2f1. [Online]. Available: https://unity.com

[33] U. Technologies, *Unity User Manual*, 2018. [Online]. Available: https://docs.unity3d.com/Manual/index.html

[34] J. van den Berg, Ming Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 1928–1935.

[35] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Computer Vision – ECCV 2016*, vol. 9912, Oct. 2016, pp. 549–565.

[36] G. Ferrer, A. Zulueta, F. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous Robots*, vol. 41, July 2016.

[37] A. Bera, T. Randhavane, R. Prinja, and D. Manocha, "Sociosense: Robot navigation amongst pedestrians with social and psychological constraints," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 7018–7025.

[38] C.-E. Tsai and J. Oh, "Navigan: A generative approach for socially compliant navigation," 2020.

[39] C.-E. Tsai, "A generative approach for socially compliant navigation," Master's thesis, Pittsburgh, PA, June 2019.

[40] E. T. Hall, "Silent assumptions in social communication," *American Anthropologist*, vol. 66, no. 6, pp. 154–163, 1964.