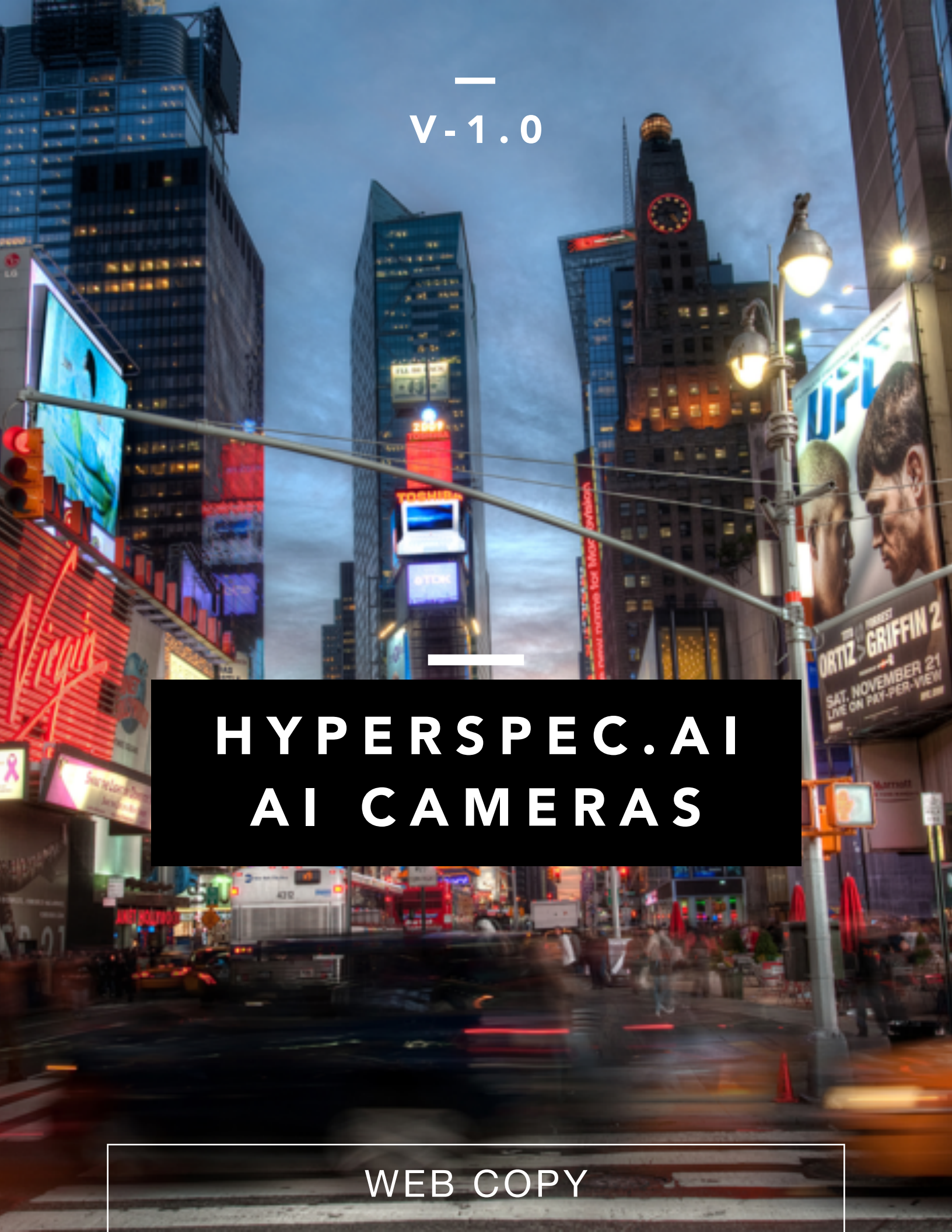

V-1.0
HYPERSPEC.AI
AI CAMERAS
WEB COPY

ABSTRACT

Self driving cars are going to drive the next industrial revolution by freeing up valuable human resources and by democratizing mobility & accessibility. The technology is quite promising and has tremendous amount of scope. However, there are key bottlenecks that need to be addressed, in order for the systems to be economically viable. One of the key bottlenecks is the High Definition Map requirement to enable the car to have scene context. In this paper we describe the state of the art and present a next generation orthogonal approach to creating scene context for self driving cars.

Hyperspec.ai is developing a next generation vision pipeline that will utilize equirectangular imagery instead of standard pin-hole images to gain an understanding of the scene. We present novel innovations that will contribute to the perception stacks which will dramatically revolutionize how self driving cars create scene context while navigating in complex environments

TABLE OF CONTENTS

Abstract 2

Table of Contents 3

State of the art
- Architecture 4
- Sensors 5
- High Definition Maps 6
- Localization 7
- Real Time Perception 8

Hyperspec Stack
- Architecture 9
- Real Time Scene Construction 10
- Depth Estimation 11
- Cross-View Localization 12
- Real Time Perception 13
- Reinforcement Learning Based Safety Envelope 14

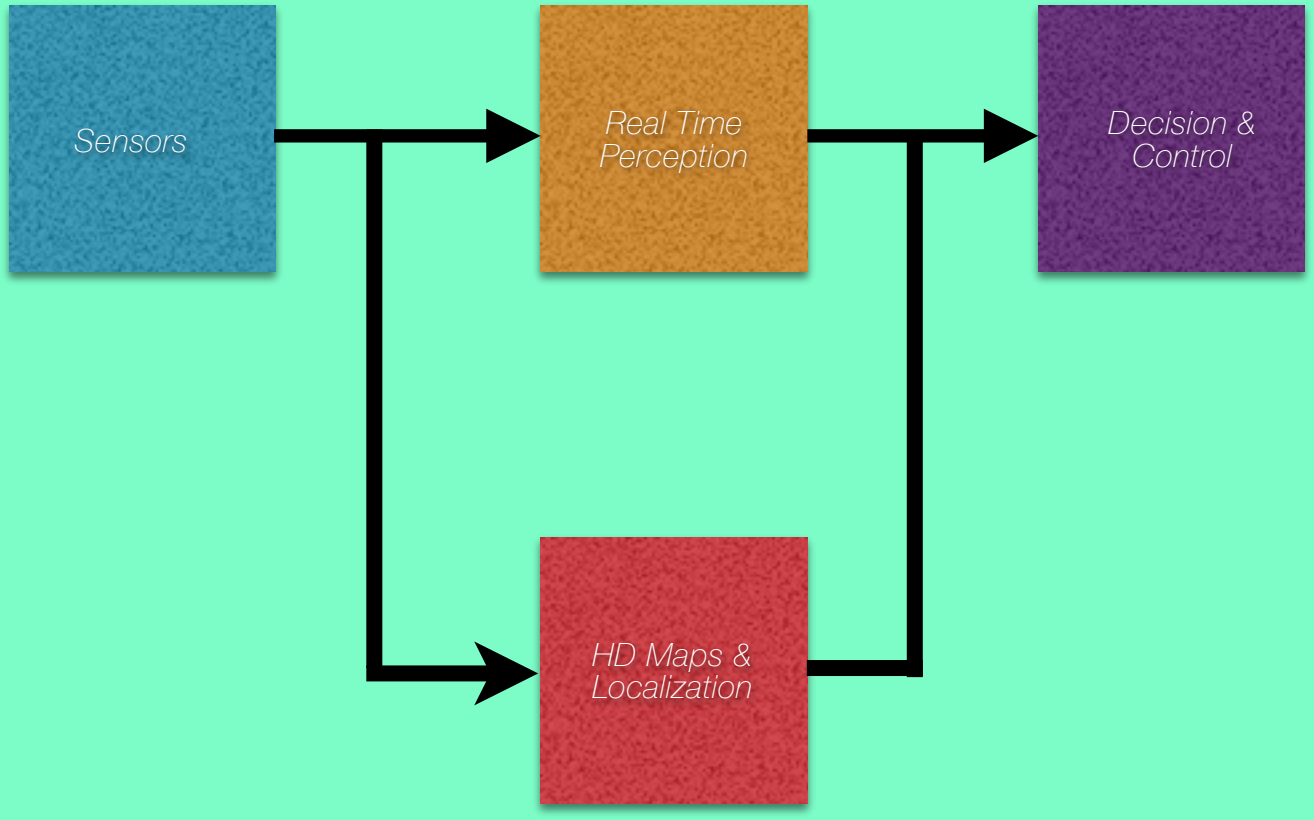
Benefits 15

Glossary TBD

STATE OF THE ART

OVERVIEW OF SELF DRIVING STACK USING HIGH DEFINITION MAPS AND LOCALIZATION. IN DEPTH VIEW OF SUBSYSTEMS AND CONSTRAINTS.

ARCHITECTURE



SENSORS

Typical self driving car will have Cameras, LiDAR, Radars, IMU, GPS, Ultrasonic, OBD sensors. The sensor stack will likely have a 360 perspective in a full self driving vehicle and a more limited perspective in ADAS applications. In addition to the sensors, specialized hardware for edge computing, networking and storage is required to load reference map data from disk into memory. This is typically very costly and operationally intensive to maintain.

SENSORS	DESCRIPTION
HDL-32 Velodyne LiDAR	4-6 Velodyne LiDAR sensors on each corner.
360 Camera Perspective	6-10 2MP - 1MP camera sensors with 30-60 FPS
XSens IMU	Low drift MEMs IMU / GPS sensor
Ultrasonic	Low power active sensors for short range
77 GHz Radar	Ego-motion and object tracking sensor
Universal OBD Sensor	Sensor for Vehicle Speed, Steering Angle, GPS Loc
Edge Computing Nodes	Computing infrastructure with GPUs for processing
Networking Device	Switching Fabric to transfer data between nodes
Storage Nodes	Storage nodes that integrate with computing nodes
Power Distribution System	Power management infrastructure for all sensors and computing infrastructure.

HIGH DEFINITION MAPS

Traditionally HD Maps are stored on disk inside the vehicle on the hard drive. The self driving car uses the GPS sensor to obtain a rough location from the GPS sensor and then proceeds to load sections from the prior HD Map which match the Lat, Long location data from the GPS sensor from the car's storage into memory. The car then utilizes its real time perception stack to determine the location of the car within the map by comparing its real time sensor output with the HD maps data stored in memory. This error in localization is computed and the car is placed within the map.

After this stage, the car utilizes the geometric and semantic data from the HD map to navigate in the environment. The map mostly describes the semantic context of the scene with respect to the static objects. The dynamic objects in the scene are obtained from the real time perception stack. The two views are fused using the 6-DoF pose estimation from the localization stack. The comprehensive view can be described as the scene context which is used by the self driving car to operate.

Weakness of Approach:

- *Staleness of Data:* The frequency between map updates is very high, leading to a high chance that some portions of the scene context is no longer in sync with the actual environment
- *Localization error:* The 6 DoF position estimation from localization is used to superimpose the map on the perception stack. Any misalignment in localization causes the geometry in the scene to be misaligned.
- *Operationally complicated:* The automatic generation of map data in the cloud has gaps, leading to manual annotation efforts to fix the gaps. A large portion of the scene is manually annotated, leading to longer turn around times.
- *Restricted Areas:* The self driving car deployment is restricted to areas that have HD maps
- *Bad Pricing:* Map pricing is on a per km basis and the going rate is $5,000 per km. The end user is not willing to pay this at a large scale.
- *Hammer Approach:* The HD maps approach is overkill for 95% of road infrastructure. Most roads with four way stops and simple intersections don't need HD maps.

LOCALIZATION

During the creation of the HD maps, the reference map data is stored for use later in order to localize the self driving car. The most common reference map data format is a point cloud. When generating point clouds, the reference map needs to undergo a loop closure process in order to make sure the scene is consistent across multiple arrivals. Loop closure is performed by comparing the perspectives of the same location across different arrivals and computing the scene inconsistency between them. The transformation matrix which describes the rotational and translation error is then used to create a bundle adjustment across both arrivals. This allows the scene to become consistent wherein, the geometric integrity is maintained across different arrivals. Loop closure becomes more complex when the point cloud has a high amount of drift. If the drift is encoded into the reference map, the same drift artifact will create a rubber band effect in localization as well.

Once the loop closed reference map is published to the car, the car can utilize the map to register its real time sensor data to the stored map. The transform from the IMU/GPS Lat Long position to the registered map position is the localization correction.

Weakness of Approach:

- *Reference Scene Distortion:* Drift in the reference map creation negatively impacts the localization performance and matching capabilities. Real time sensor data will not have drift if it doesn't accumulate points over time using memory, therefore trying to match that scene with a reference map that has drift will cause localization to lose the lock on the vehicle's position.
- *Changes in the environment:* Changes to the environment will also have poor localization performance. A reference map that has one snapshot of the environment might not match the snapshot from the sensor data stream if there are physical changes in scenery. Localization is negatively impacted by scene inconsistencies between the reference map and the sensor data stream.
- *Limited Coverage:* The self driving car can only localize itself in areas where a reference 3D map is available.

REAL TIME PERCEPTION

The real time perception stack processes sensor data from the vehicle to detect and track all objects of interest. The main pipeline is extracting information such as the lane geometry, vehicles, pedestrians, bicyclists, etc. In addition to the object detection, vision odometry, scene segmentation, free space detection are some core modules that are part of the real time perception stack.

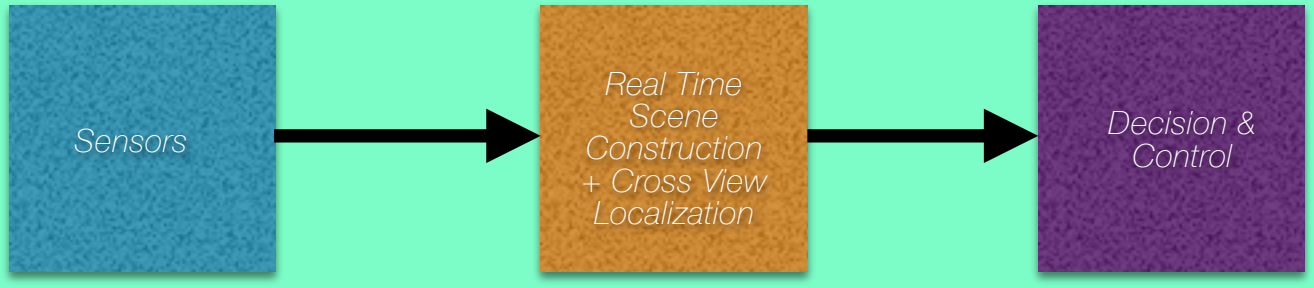
Weakness of Approach:

- *Scene inconsistency:* If localization or the reference map is compromised, then the scene context which combines the reference map with the object lists produced by the real time perception stack will not be accurate; leading to disengagements.
- *Changes in the environment:* Changes to the environment will also have poor localization performance. A reference map that has one snapshot of the environment might not match the snapshot from the sensor data stream if there are physical changes in scenery. Localization is negatively impacted by scene inconsistencies between the reference map and the sensor data stream.

HYPERSPEC STACK

OVERVIEW OF SELF DRIVING STACK USING REAL TIME SCENE CONSTRUCTION AND CROSS VIEW LOCALIZATION.

ARCHITECTURE



REAL TIME SCENE CONSTRUCTION

Real time scene construction relies on the sensor data streams to create a scene context without using High Definition Maps. Fundamentally, the stack is required to provide the geometry of the road furniture the scene, the dynamic objects, the state estimation of each object, the classification of objects. The scene context is derived from an underlying data structure. The colorized 3D mesh can be created in three different ways: 360 camera, 360 stereo cameras, 360 camera + radar fusion or 360 camera + LiDAR fusion.

Once the mesh is created, it can be rendered in a game engine as an object. Game engines allow you to decouple camera perspectives from the object files, allowing the vehicle to generate orthographic top down perspectives from equirectangular depth imagery. Localization uses cross view localization for global corrections and vision odometry for dead reckoning between global corrections.

Strengths of Approach:

- *Low Latency Data:* Locally generated sensor data is close to real time.
- *Increased Localization Tolerances:* High definition maps require precise localization, however using real time scene construction omits the high precision localization requirement since we are only using the local sensor data to construct the scene. Since the sensors are directly mounted on the vehicle frame, both the perception and scene construction stack use the locally sourced sensor data without prior map information, making high precision localization unnecessary.
- *Operationally simple:* There is no offloading of map data since everything is locally generated and processed.
- *Ubiquitous operations:* Since the scene context is generated real time, there are no restrictions in terms of where the self driving car can operate.
- *Low Cost Pricing: Generating the scene context locally drastically reduces the cost.*
- *Elegant Approach:* 95% of the road network can be described using real time scene construction and the remaining 5% may rely on HD maps.

DEPTH ESTIMATION

Initially the sensors are calibrated by co-mounting a LiDAR adjacently. The point cloud generated by the LiDAR sensors acts like the ground truth. The Camera data and point cloud data is loaded into a game engine. The LiDAR data is rendered as particles in the scene using cube materials that span 5cm each. Additionally, the camera data is assigned to a camera perspective object within the game engine and given extrinsic and intrinsic parameters. The camera perspective is then modified to spatially align the pixel data with the point cloud data. Edge features are used to modulate the horizontal field of view, vertical field of view, distortion matrix, near depth field of view, lens size and the sensor size. Each calibrated camera then publishes it's parameters to an initialization config file. Once all individual cameras are calibrated, an image stitching algorithm fuses the images from each individual cameras into a single frame. The stitched equirectangular imagery provides a single frame buffer that is exposed to the real time scene construction stack. Each time the system initializes, it runs an auto-calibration script to tune and update the extrinsic and intrinsic parameters. For time alignment a GPS / PPS signal is sent via GPIO to each camera to time sync the LiDAR and camera data.

Pseudo LiDAR is used to train a neural network to estimate depth whilst using the LiDAR as a ground truth sensor. This is an unsupervised machine learning approach to machine learning since the reference ground truth data is generated using an orthogonal form of measurement. Additionally, the point cloud that is generated is compared against the LiDAR point cloud in real time to enable a continuous machine learning pipeline. The depth estimation gets better as more data is collected from both sensors.

Strengths of Approach:

- *Reduced Sensor Cost:* Pseudo LiDAR generates point clouds from cameras without using LiDAR
- *More Robustness:* Cameras already have a lot of classification and detection algorithms that can be used to detect objects in pixel space and then assign the object pose estimation in point cloud space.

CROSS VIEW LOCALIZATION

Once the depth estimation is complete, a game engine can be used to compute the orthographic top down perspective of each real time scene construction snapshot. The data structure would be a colorized 3D mesh object loaded into a game engine scene. The game engine has the flexibility to subscribe an orthographic top down camera which is perpendicular along the X axis of the vehicle. The top down perspective would capture features such as lane markings, road textures, stop bars, side walks, edges of buildings etc. Existing SatNav data has aerial and satellite imagery which also provides a top down perspective. The localization stack in the car compares the top down orthographic perspective from the equirectangular imagery with the existing SatNav reference to publish corrections in Lat, Long, and directional heading (global corrections). These global corrections can be computed at a lower frequency ranging from 2Hz to .1Hz based on the availability of good features.

For the position estimations between the global corrections, local corrections need to be computed using vision odometry. For this the equirectangular image is processed to compute salient features that are persistent across multiple frames. The features are then tracked on a frame to frame basis to compute a position offset. The 360 coverage makes the dead reckoning more robust to occlusions introduced by large objects like busses.

Strengths of Approach:

- *Worldwide Coverage:* 20cm and 50cm satellite and aerial imagery is available worldwide. The stack can be easily deployed to any location with reference imagery without a prior HD map.
- *Less Drift:* 360 equirectangular images have a lower rate of drift in comparison to an IMU sensor. We utilize the sensor robustness to create a low drift rate vision based inertial information that can track the car's movement against a traditional SatNav map. The precision is more than sufficient to track a vehicle occupying space with a set of lane markings.

REAL TIME PERCEPTION

The real time perception stack processes sensor data from the vehicle to detect and track all objects of interest. The main pipeline is extracting information such as the lane geometry, vehicles, pedestrians, bicyclists, etc. In addition to the object detection, vision odometry, scene segmentation, free space detection are some core modules that are part of the real time perception stack.

Strength of Approach:

- *Scene consistency:* With localization no longer a requirement for scene context, the self contained real time scene construction pipeline can guarantee scene consistency between the outside environment and the scene context.
- *Flexible data structures:* From a pure software perspective the real time scene constructions does not need to support legacy data structures if the perception stack and decision stack are updated in a lock-step deployment.

REINFORCEMENT LEARNING BASED SAFETY ENVELOPE

There might be cases where the real time scene construction or cross view localization fall outside of the operating parameters. Typically this triggers the disengagement of the car. The disengagement can be avoided if there is a safety envelope giving the vehicle a safety rating in real time to denote if it was operating within or outside the safety envelope. A safety envelope operates outside the scope of the real time scene constructions and operates independently. This module uses an end to end neural network to attribute a safety rating to the vehicle by only using the sensor data.

The training data will be created using dash cam footage from student drivers with safety rating annotations from driving instructors. Whenever a student driver encounters an unsafe situation or preforms an unsafe maneuver the driving instructor will annotate that specific section of video with an unsafe rating. Whenever the student is operating safely, the driving instructor will use a safe rating. Over time the videos and the annotations will be fed into a reinforcement learning framework to define a safety envelope using an end to end neural network. The safety envelope enables the self driving car to rank the situation in the environment and its own driving performance using a safety rating. This allows the car to possibly continue the self driving functionality with the real time scene construction and cross view localization aren't performant, by using the standard ADAS functionality and the safety envelope. This would prevent unnecessary disengagements and improve the robustness of the car.

Strength of Approach:

- *Reduced Disengagements:* The self driving stack has capabilities to operate event when the real time scene construction and localization aren't performant by relying on the ADAS functionality and Safety Envelope
- *Avoiding rules based autonomy:* HD maps defining the constraint set for what the car is allowed to do is a failed approach, since there are too many long tail exceptions in driving. Therefore the HD maps will never be able to completely enable a disengagement free experience if the scene context is out of sync with the environment.

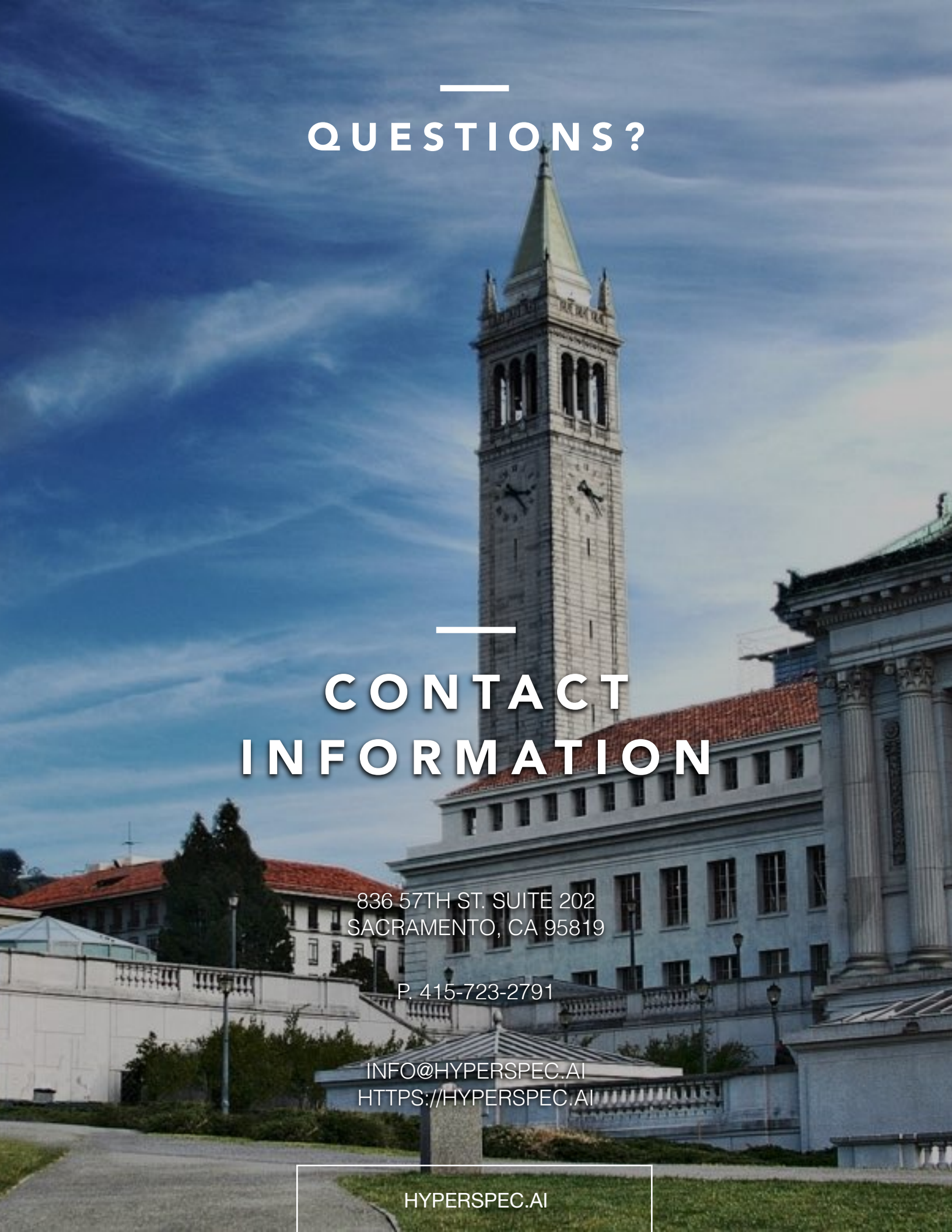
BENEFITS

OVERVIEW OF BENEFITS FROM USING REAL TIME SCENE CONSTRUCTION AND CROSS VIEW LOCALIZATION.

GEOFENCED AREAS → **WORLDWIDE**

$5000/KM → **$0/KM**

2-3 MONTH FREQ → **REAL TIME FREQ**



QUESTIONS?

CONTACT INFORMATION

836 57TH ST. SUITE 202
SACRAMENTO, CA 95819

P. 415-723-2791

INFO@HYPERSPEC.AI
HTTPS://HYPERSPEC.AI