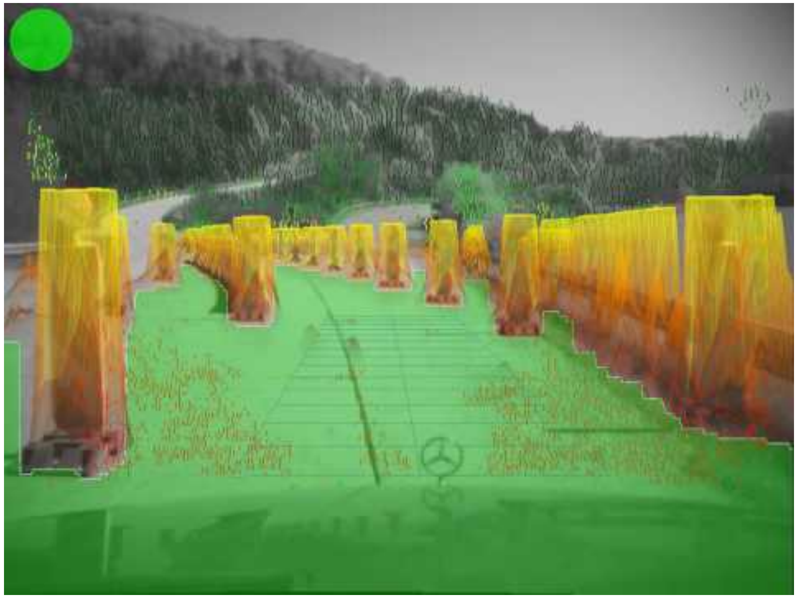
Free-Space Estimation

HAO WU
UNIVERSITY OF TORONTO
MARCH 1ST, 2016

OUTLINE

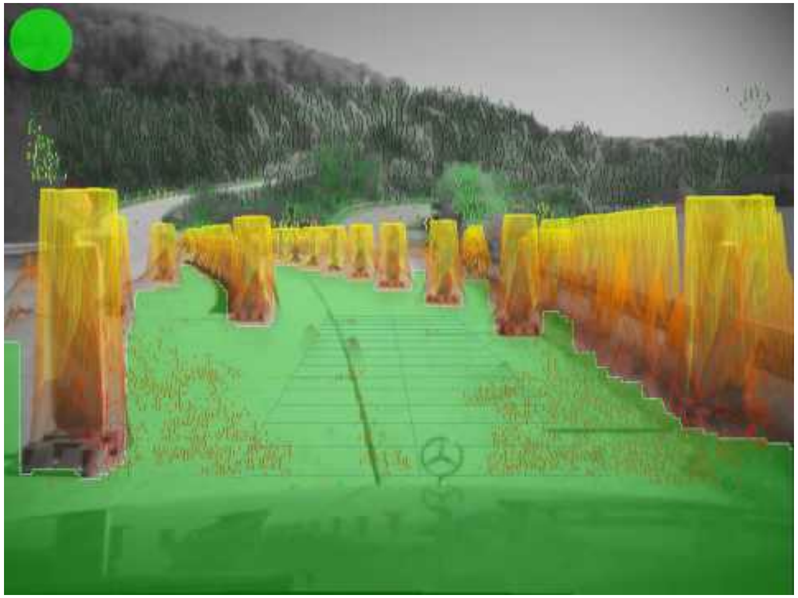
- What is Free-Space Estimation?
- Challenges
- Occupancy Grid Mapping Algorithm
- Approaches to the Problem
  - Dense Stereo
  - Occupancy Grid
  - Free-Space Computation by DP
  - Height Segmentation
  - “Stixel” Extraction
- Future Work



Source: https://www.youtube.com/watch?v=tiRi3I_wbNk

OUTLINE

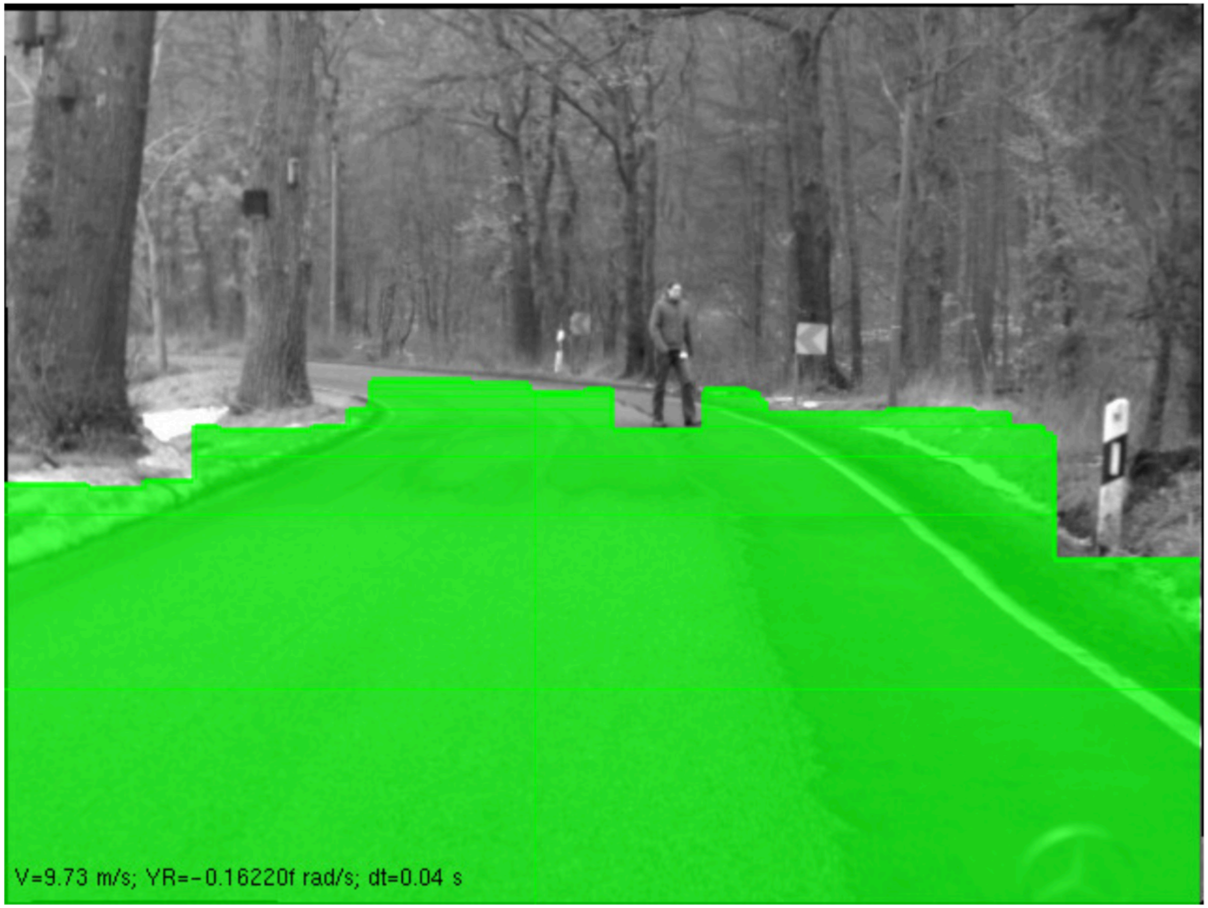
- **What is Free-Space Estimation?**
- Challenges
- Occupancy Grid Mapping Algorithm
- Approaches to the Problem
  - Dense Stereo
  - Occupancy Grid
  - Free-Space Computation by DP
  - Height Segmentation
  - “Stixel” Extraction
- Future Work



Source: https://www.youtube.com/watch?v=tiRi3I_wbNk

Free-Space Estimation

- The world region where navigation **without collision** is guaranteed
- In robotics, free space is required when planning the path between 2 points
- Vision-based?



Occupancy Grid Mapping

- Why Occupancy Grid Mapping?
  - No robot's odometry is perfect!
  - Many of the SLAM techniques do not generate maps fit for path planning and navigation.
- The main utility of the occupancy grid technique is in post-processing.
- Occupancy grid maps are often used after solving the SLAM problem by some other means, and taking the resulting path estimates for granted.

Occupancy Grid

Definition.

An occupancy grid M is a 2-D array / grid, which models occupancy evidence of the environment.

The 3D world is orthographically projected on a plane P parallel to the road (assuming the floor surface is planar).

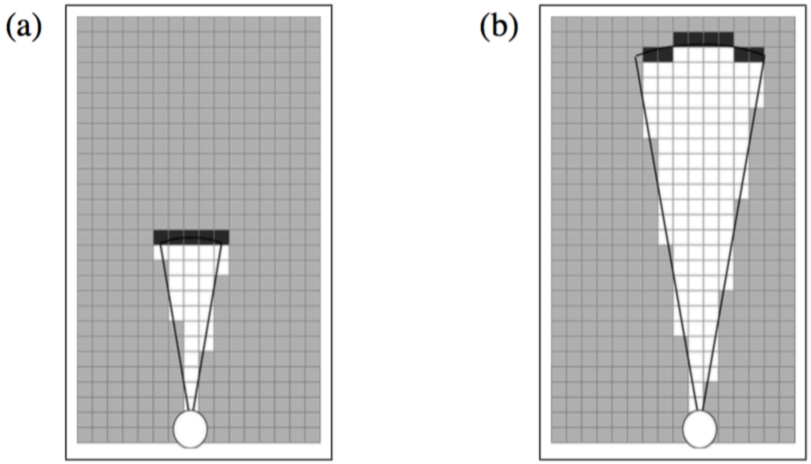
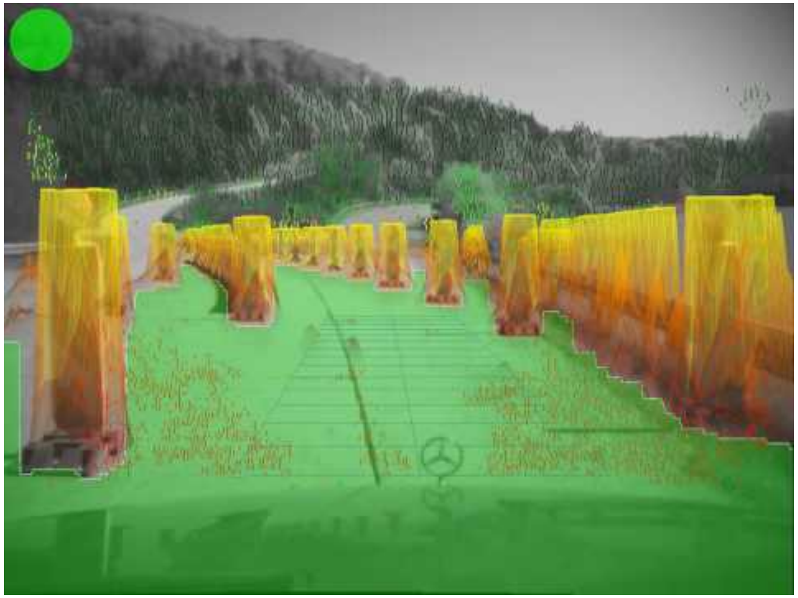


Figure 9.2 Two examples of our inverse measurement model **inverse_range_sensor_model** for two different measurement ranges. The darkness of each grid cell corresponds to the likelihood of occupancy.

OUTLINE

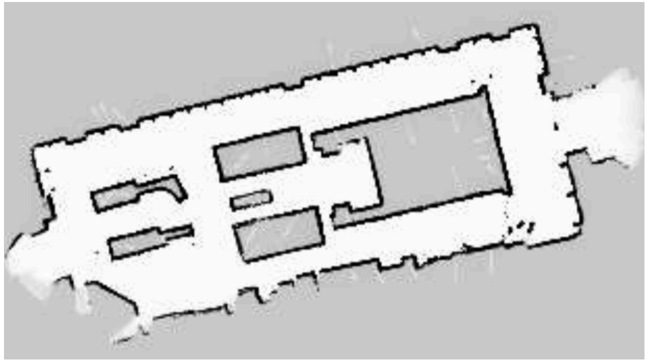
- What is Free-Space Estimation?
- **Challenges**
- Occupancy Grid Mapping Algorithm
- Approaches to the Problem
  - Dense Stereo
  - Occupancy Grid
  - Free-Space Computation by DP
  - Height Segmentation
  - “Stixel” Extraction
- Future Work

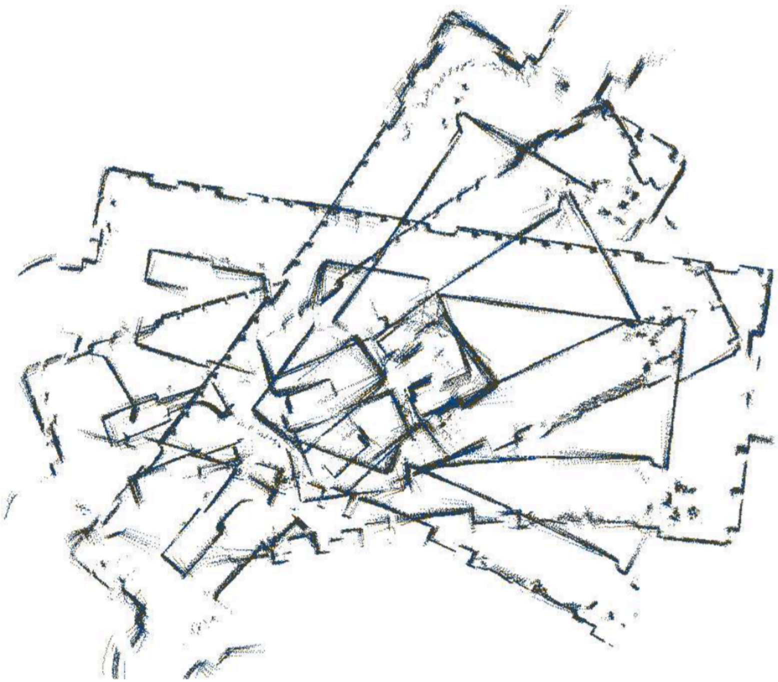


Source: https://www.youtube.com/watch?v=tiRi3I_wbNk

Challenges

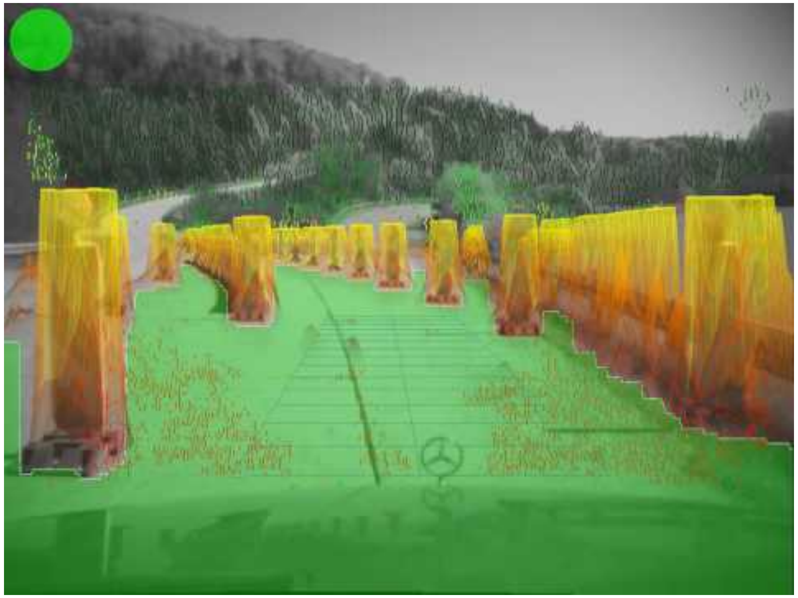
- Hypothesis space is huge
- Learning maps is a chicken and egg problem
- Hardness of the problem
  - Size
  - Noise
  - perceptual ambiguity (pattern)
  - Loop closure





OUTLINE

- What is Free-Space Estimation?
- Challenges
- **Occupancy Grid Mapping Algorithm**
- Approaches to the Problem
  - Dense Stereo
  - Occupancy Grid
  - Free-Space Computation by DP
  - Height Segmentation
  - “Stixel” Extraction
- Future Work



Source: https://www.youtube.com/watch?v=tiRi3I_wbNk

Occupancy Grid Mapping

Algorithm occupancy_grid_mapping($\{l_{t-1,i}\}, x_t, z_t$)**:**

for all cells $\mathbf{m}_i$ do
    if $\mathbf{m}_i$ in perceptual field of z_t then
        $l_{t,i} = l_{t-1,i} + \textbf{inverse\_sensor\_model}(\mathbf{m}_i, x_t, z_t) - l_0$
    else
        $l_{t,i} = l_{t-1,i}$
    endif
endfor
return $\{l_{t,i}\}$

$$\textbf{inverse\_sensor\_model}(\mathbf{m}_i, x_t, z_t) \quad = \quad p(\mathbf{m}_i \mid z_t, x_t)$$

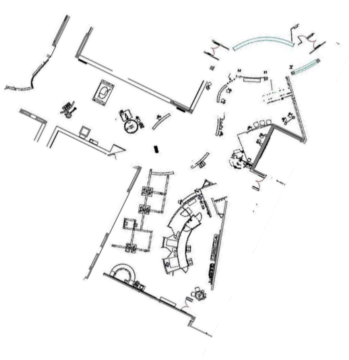
Occupancy Grid Mapping

Algorithm inverse_range_sensor_model(i, x_t, z_t)**:**
  Let x_i, y_i be the center-of-mass of $\mathbf{m}_i$
  $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$
  $\phi = \text{atan2}(y_i - y, x_i - x) - \theta$
  $k = \text{argmin}_j \, |\phi - \theta_{j,\text{sens}}|$
  if $r > \min(z_{\max}, z_t^k + \alpha/2)$ or $|\phi - \theta_{k,\text{sens}}| > \beta/2$ then
    return l_0
  if $z_t^k < z_{\max}$ and $|r - z_{\max}| < \alpha/2$
    return l_{occ}
  if $r \leq z_t^k$
    return l_{free}
  endif

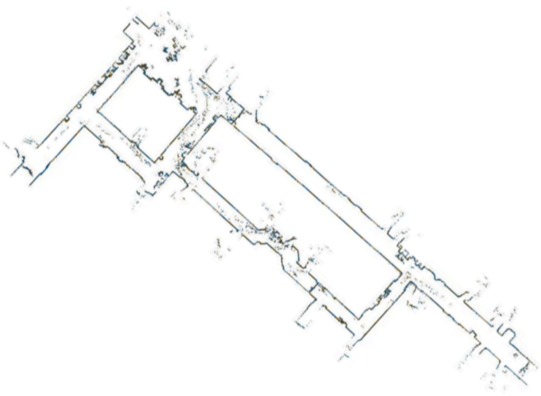
Occupancy Grid Mapping

(a)

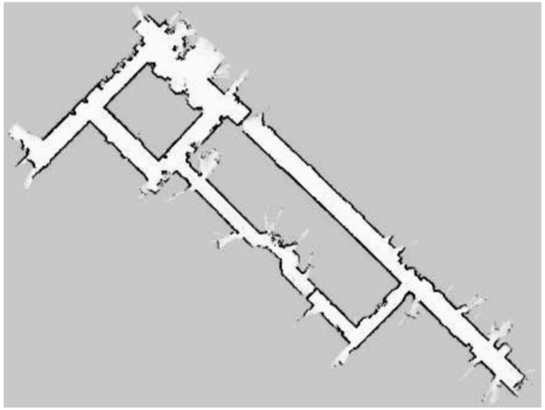
(b)



(a)



(b)



Occupancy Grid Mapping

Error Function

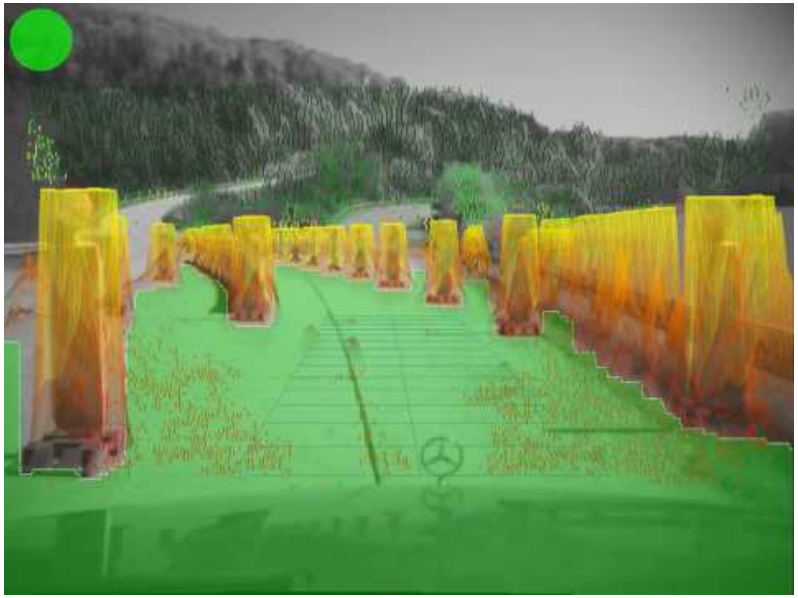
$$p(\mathbf{m}_i^{[k]} \mid \text{input}^{[k]}, W) = \begin{cases} f(\text{input}^{[k]}, W) & \text{if } \mathbf{m}_i^{[k]} = 1 \\ 1 - f(\text{input}^{[k]}, W) & \text{if } \mathbf{m}_i^{[k]} = 0 \end{cases}$$

$$p(\mathbf{m}_i^{[k]} \mid \text{input}^{[k]}, W) = f(\text{input}^{[k]}, W)^{\mathbf{m}_i^{[k]}} \, (1 - f(\text{input}^{[k]}, W))^{1-\mathbf{m}_i^{[k]}}$$

$$\begin{aligned} J(W) &= -\sum_i \log \left[f(\text{input}^{[k]}, W)^{\mathbf{m}_i^{[k]}} \, (1 - f(\text{input}^{[k]}, W))^{1-\mathbf{m}_i^{[k]}} \right] \\ &= -\sum_i \mathbf{m}_i^{[k]} \log f(\text{input}^{[k]}, W) + (1 - \mathbf{m}_i^{[k]}) \log(1 - f(\text{input}^{[k]}, W)) \end{aligned}$$

OUTLINE

- What is Free-Space Estimation?
- Challenges
- Occupancy Grid Mapping Algorithm
- **Approaches to the Problem**
  - Dense Stereo
  - Occupancy Grid
  - Free-Space Computation by DP
  - Height Segmentation
  - “Stixel” Extraction
- Future Work



Source: https://www.youtube.com/watch?v=tiRi3I_wbNk

Algorithm Overview

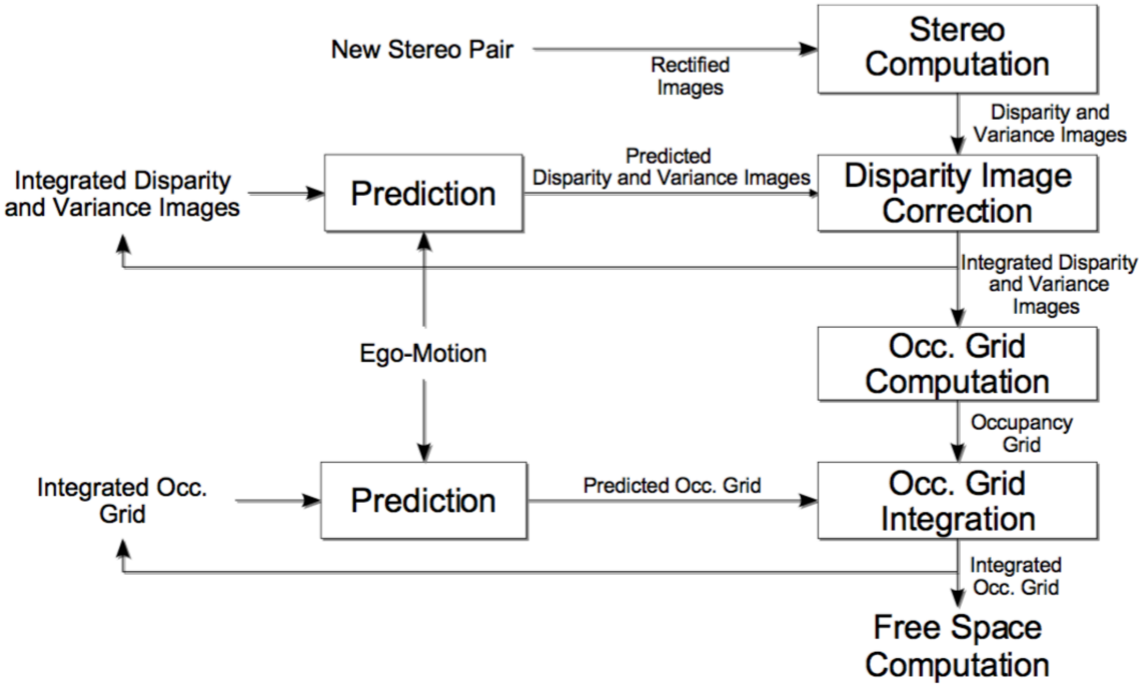


Fig. 1. Block diagram of the algorithm.

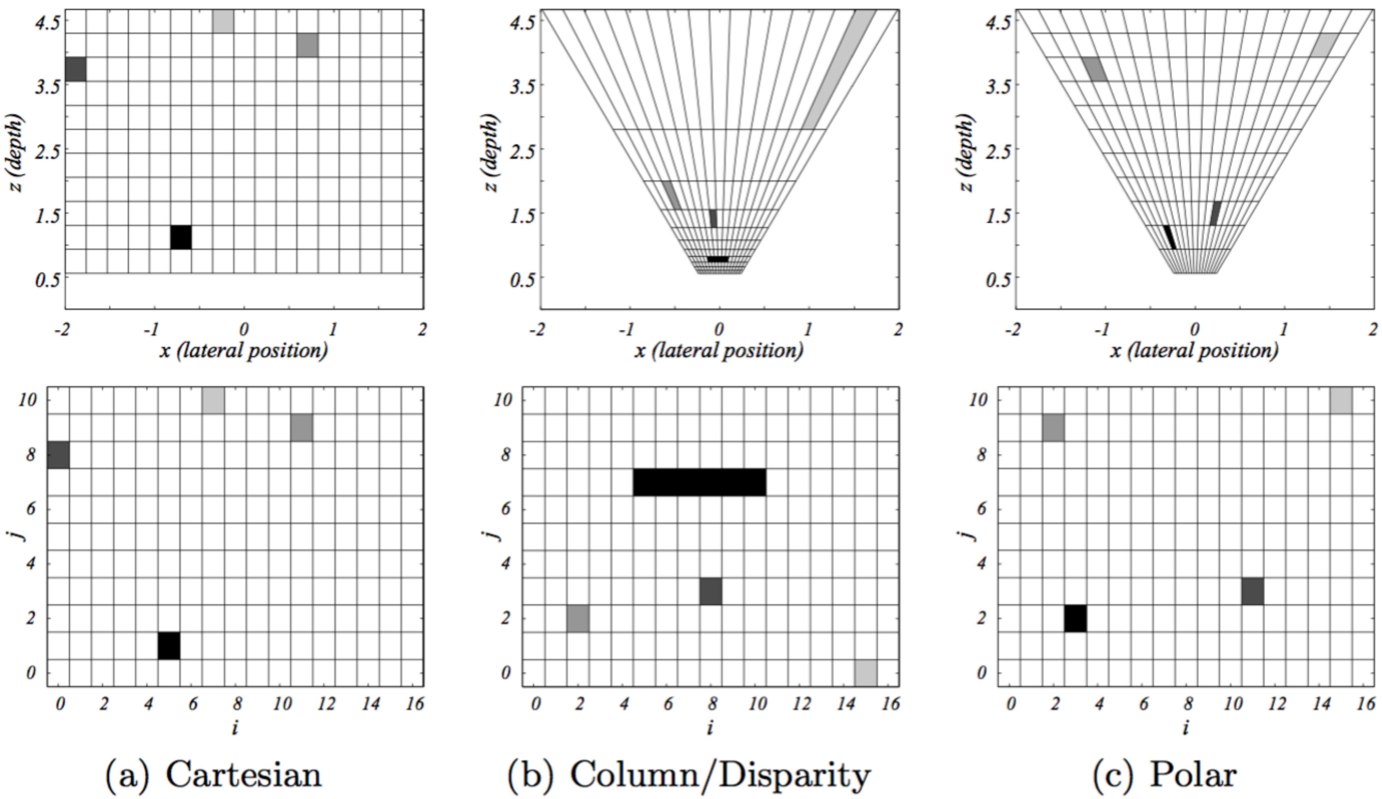
1. Dense Stereo

- Most real time stereo algorithm based on local optimization techniques deliver sparse disparity data.
- The only requirement: computation of enough disparity measurements to capture all relevant objects in a scene.
- The stereo algorithm from previous page generates a disparity and a variance image.
  - The variance image contains the estimated variance of each measured disparity
- Compute the occupancy grid from stereo measurements.

2. Occupancy Grid

- Cell (i, j)
  - i: lateral component
  - j: depth component
- Every cell of the grid maintains an occupancy likelihood $D(i, j)$
- $D(i, j) = \sum_{k=1}^{m} L_{ij}(\boldsymbol{m}_k)$
  - $\boldsymbol{m}_k = (u, v, d)^T$, where u, v: left image coordinate, d: disparity
- 3 types of occupancy grid
  - Cartesian Occupancy Grid: $L_{ij}(\boldsymbol{m}_k) = G_{\boldsymbol{m}_k}\left(P(p_{ij}) - \boldsymbol{m}_k\right)$
  - Column/Disparity Map: $L_{ij}(\boldsymbol{m}_k) = G_{\boldsymbol{m}_k}\left((u_{ij} - u, 0, d_{ij} - d)^T\right)$
  - Polar Occupancy Grid: $L_{ij}(\boldsymbol{m}_k) = G_{\boldsymbol{m}_k}\left((u_{ij} - u, 0, d'_{ij} - d)^T\right)$
    - $d'_{ij} = \frac{f_u B}{z_{ij}}$, disparity corresponding to the cell depth

2. Occupancy Grid

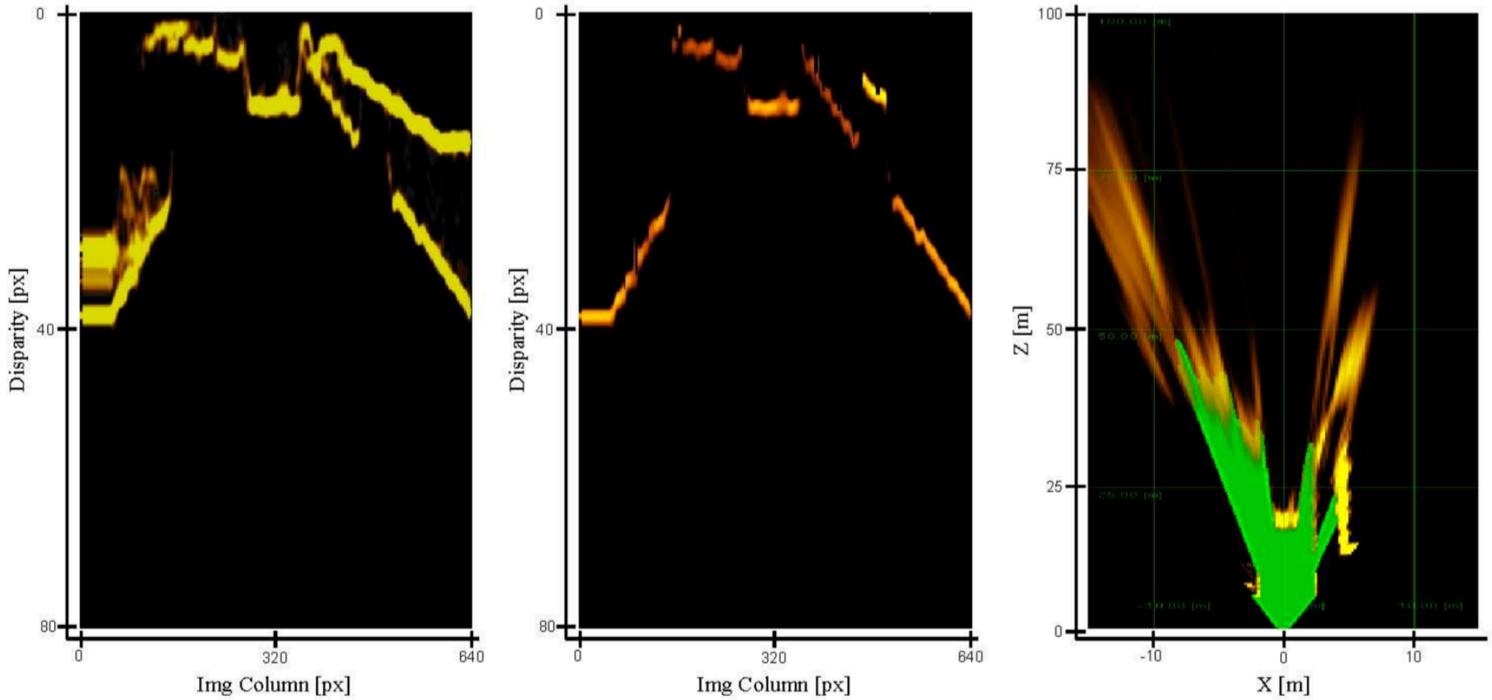


Time

Decreasing resolution

Optimal

2. Occupancy Grid



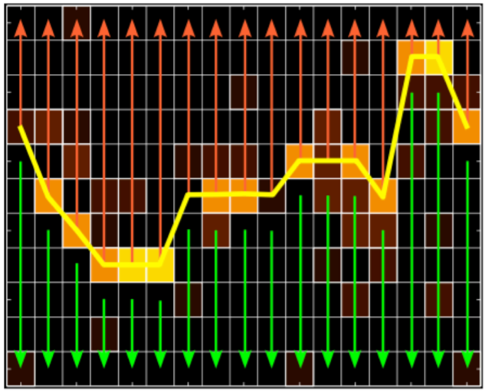
(a) Polar occ. grid. (b) Background subtraction. (c) Obtained free space.

2. Occupancy Grid

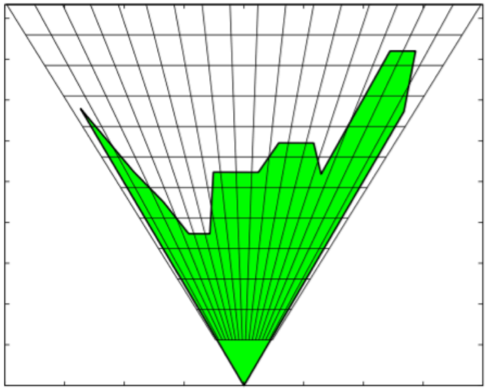
- Iconic Representation*
-

3. Dynamic Programming

- Polar coordinate
  - Every column is already in the direction of a ray, searching is straightforward
- Task: to find the first occupied cell
- Dynamic Programming is applied
  - Global optimization
  - Spatial and temporal smoothness of the solution
  - Preservation of spatial and temporal discontinuities



(a) Polar Occupancy Grid.



(b) Corresponding free space in world coordinates.

3. Dynamic Programming

- The Objective: to find the minimal path
- Cost of each edge:

$$c_{i\,j,k\,l} = E_d(i,j) + E_s(i,j,k,l),$$

- Data term:

$$E_d(i,j) = \frac{1}{D(i,j)},$$

- Smooth term:

$$E_s(i,j,k,l) = S(j,l) + T(i,j),$$

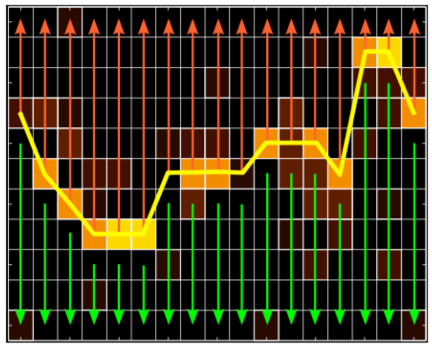
- Spatial term penalizes jumps in depth:

$$S(j,l) = \begin{cases} C_s\ d(j,l) & ;\text{ if } d(j,l) < T_s \\ C_s\ T_s & ;\text{ if } d(j,l) \geq T_s \end{cases}.$$

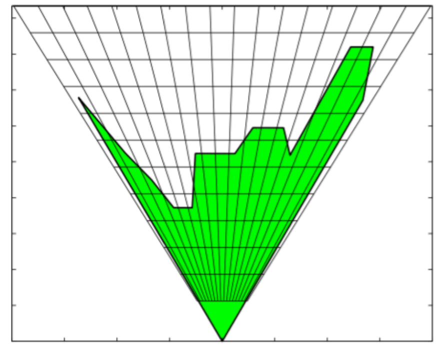
- Temporal term penalizes the deviation from solution and prediction:

$$T(i,j) = \begin{cases} C_t\ d(j,j') & ;\text{ if } d(j,j') < T_t \\ C_t\ T_t & ;\text{ if } d(j,j') \geq T_t \end{cases},$$

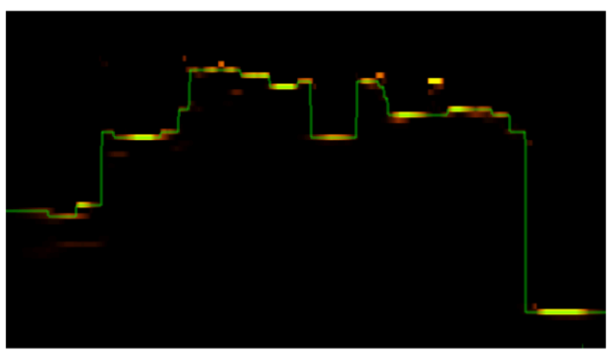
3. Dynamic Programming



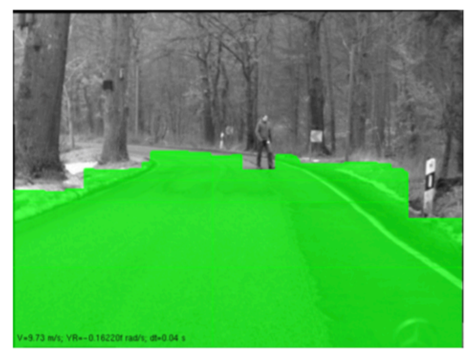
(a) Polar Occupancy Grid.



(b) Corresponding free space in world coordinates.



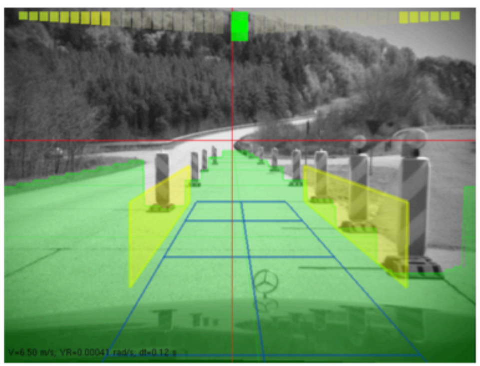
(c) Segmentation result.



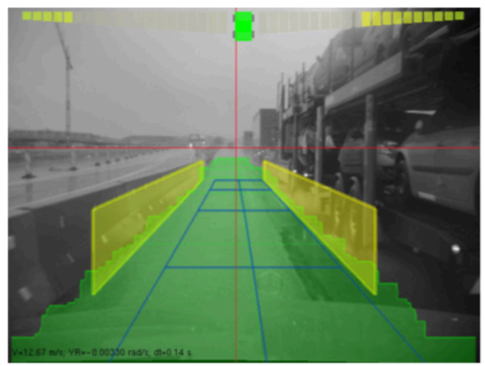
(d) Freespace.

3. Dynamic Programming

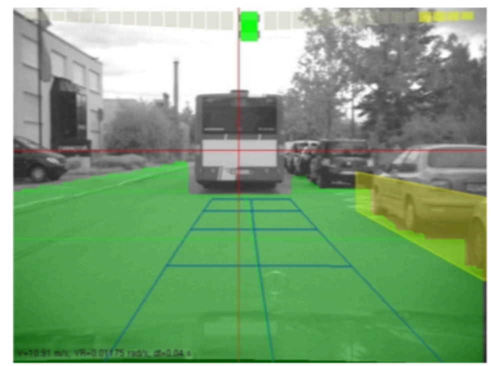
Results:



(a) Highway.



(b) Freeway.

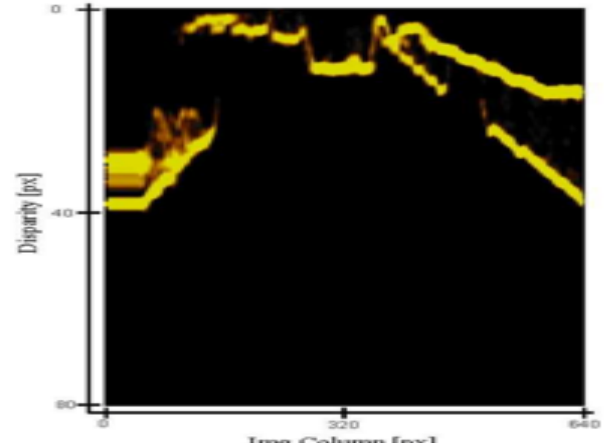


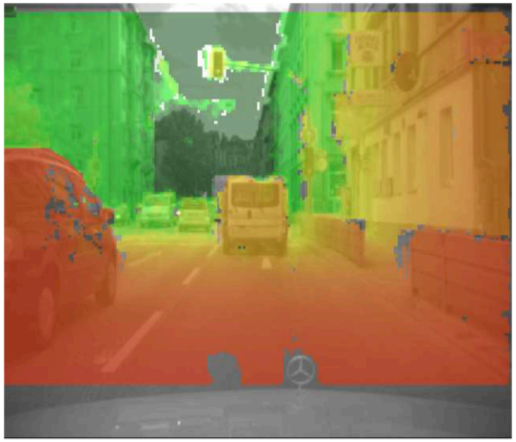
(c) Downtown.

Youtube: https://www.youtube.com/watch?v=e6O-GuI3LzQ

3. Dynamic Programming

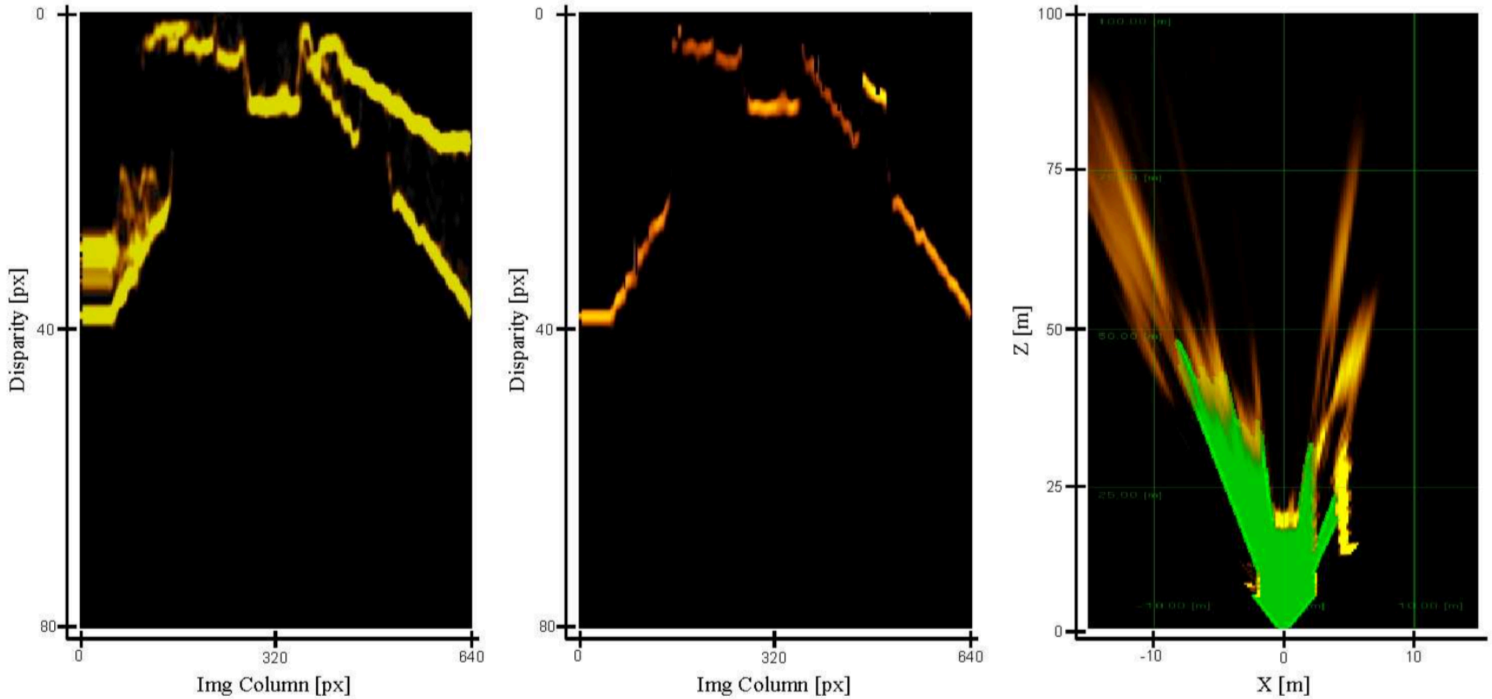
- <u>Problem</u>
  - Applying DP directly on the grid of top image might lead to a solution where the optimal boundary is found on the background object (i.e. the building) and not on the foreground object (i.e. the guardrail).
- Background subtraction
  - All occupied cells behind the first maximum which is above a given threshold are marked as free. The threshold must be selected so that it is quite larger than the occupancy grid noise expected in the grid.





(a) Dense disparity image (SGM result)

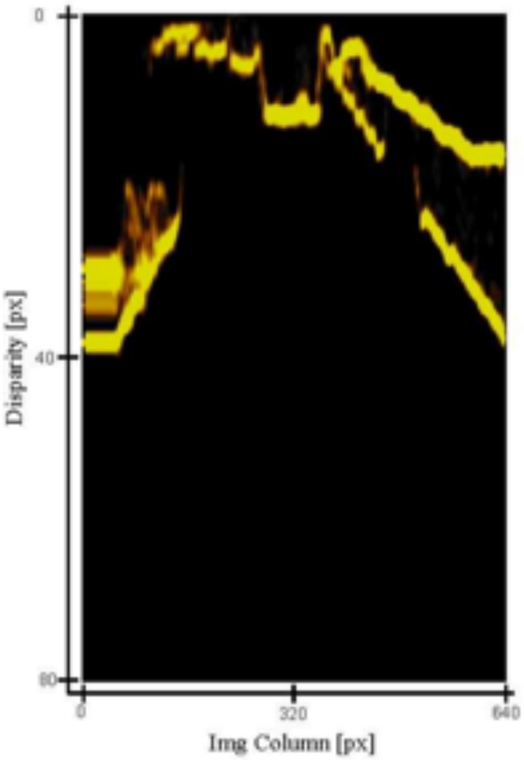
3. Dynamic Programming



(a) Polar occ. grid. (b) Background subtraction. (c) Obtained free space.

4. Height Segmentation

- The height of the obstacles
  - the optimal segmentation between foreground and background disparities.
- Goal: to find upper boundary
- Compute a cost image
- Apply DP to find the upper boundary of the objects



4. Height Segmentation

- Compute a cost image

$$C(u,v) = \sum_{i=0}^{i=v-1} M_{u,v}(d(u,i)) - \sum_{i=v}^{i=v_f} M_{u,v}(d(u,i))$$

$$M_{u,v}(d) = 2^{\left(1-\left(\frac{d-\hat{d}_u}{\Delta D_u}\right)^2\right)} - 1$$

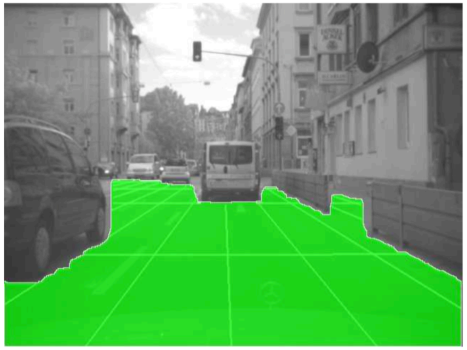
- Cost (data term + smoothness term) minimized by DP

$$c_{u,v_0,v_1} = C(u,v_0) + S(u,v_0,v_1)$$

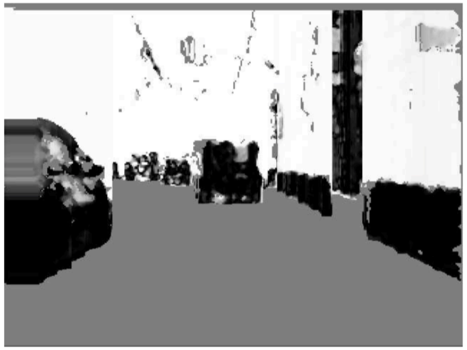
- Smoothness

$$S(u,v_0,v_1) = C_s|v_0 - v_1| \cdot \max\left(0, 1 - \frac{|z_u - z_{u+1}|}{N_Z}\right)$$

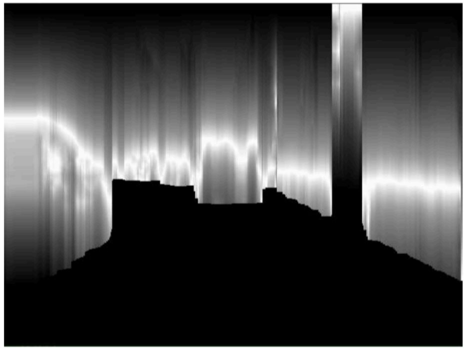
4. Height Segmentation



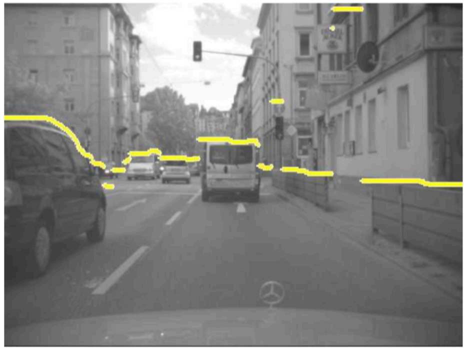
(a) Free space



(b) Membership values



(c) Membership cost image



(d) Height segmentation

5. Stixel Extraction

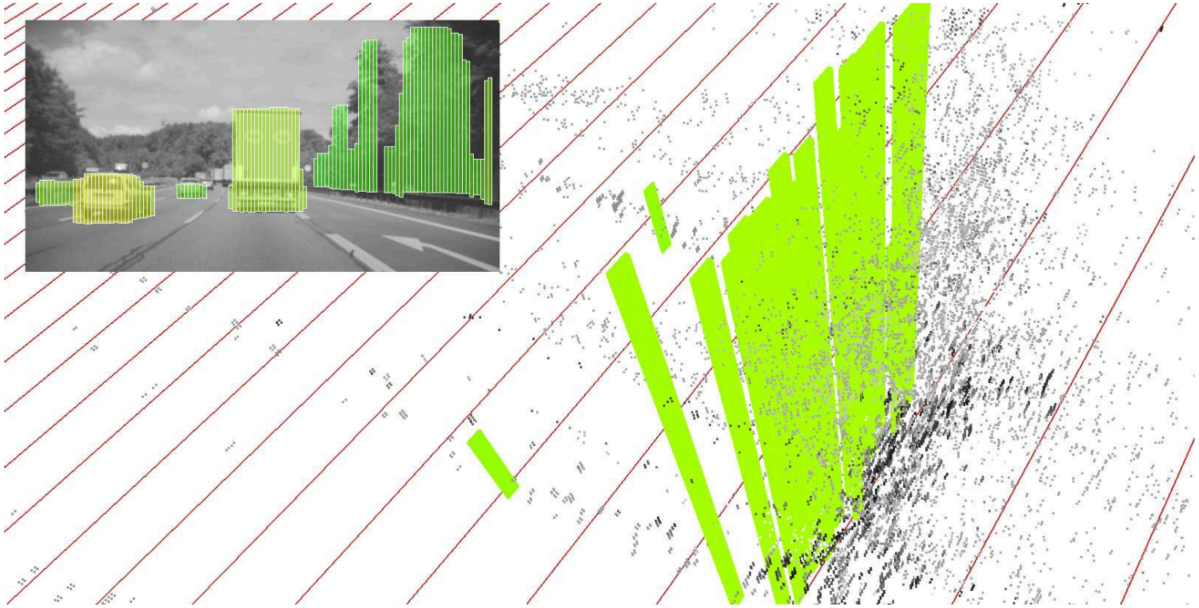
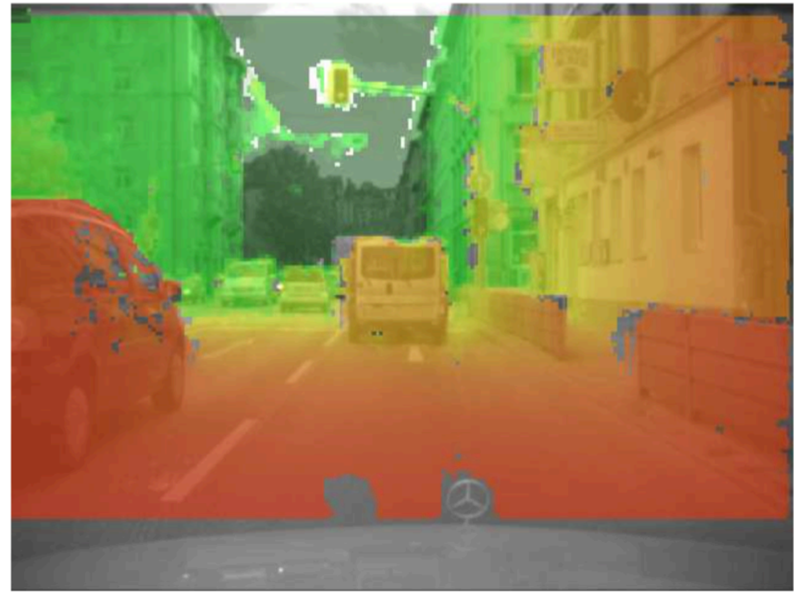


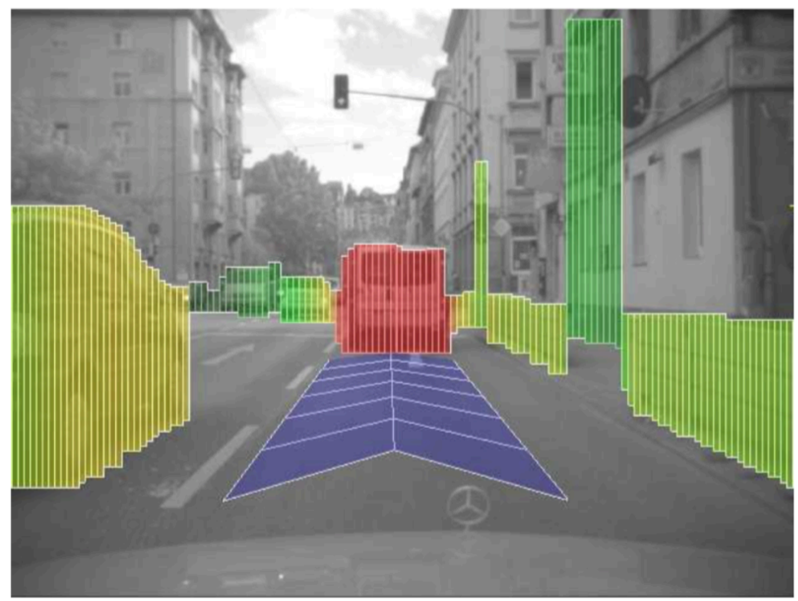
Fig. 4. 3D visualization of the raw stereo data showing a truck driving 28 meters ahead. Each red line represents 1 meter in depth. One can clearly observe the high scattering of the raw stereo data while the stixels remain as a compound and approximate the planar rear of the truck.

Stixel Representation

- “Stick Pixel” Representation
- Compact but flexible representation of the 3D traffic situation.
- Video: https://www.youtube.com/watch?v=j-8zdKq1nnc

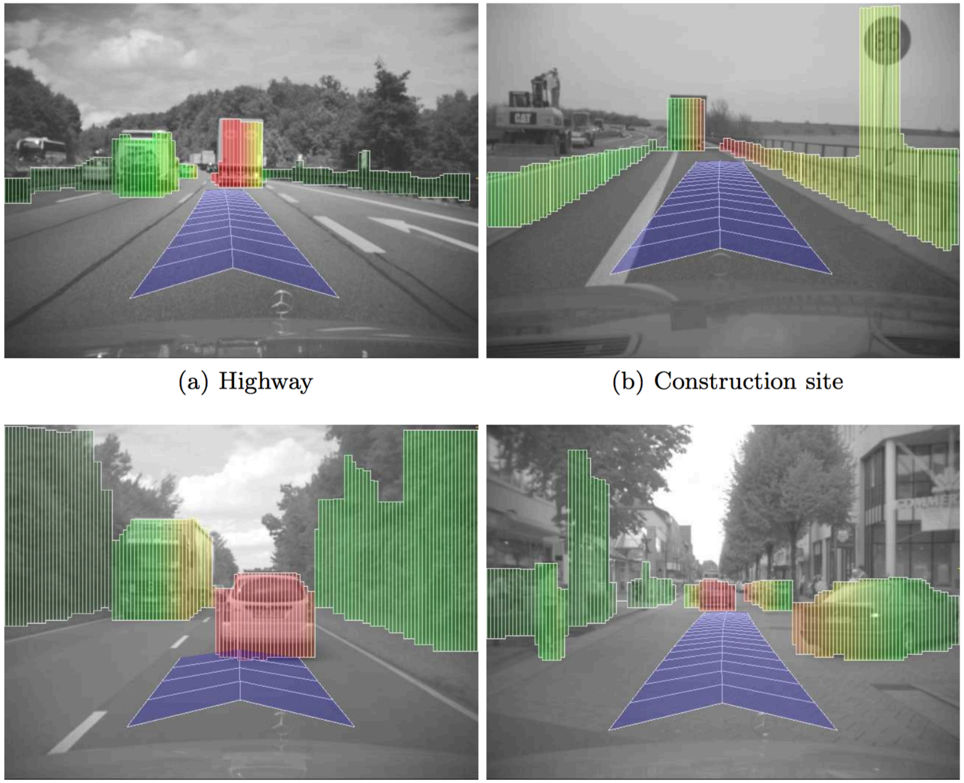


(a) Dense disparity image (SGM result)



(b) Stixel representation

5. Stixel Extraction



(a) Highway

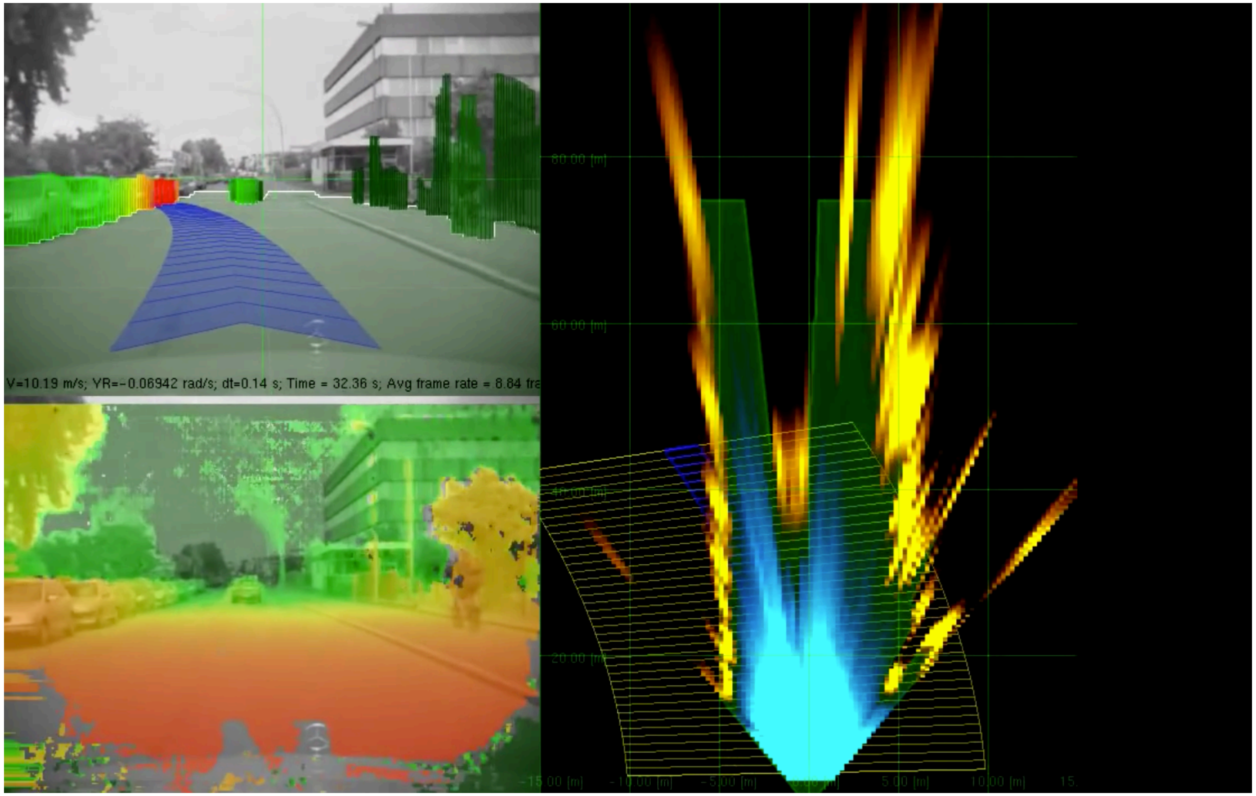
(b) Construction site

(c) Rural road

(d) Urban traffic

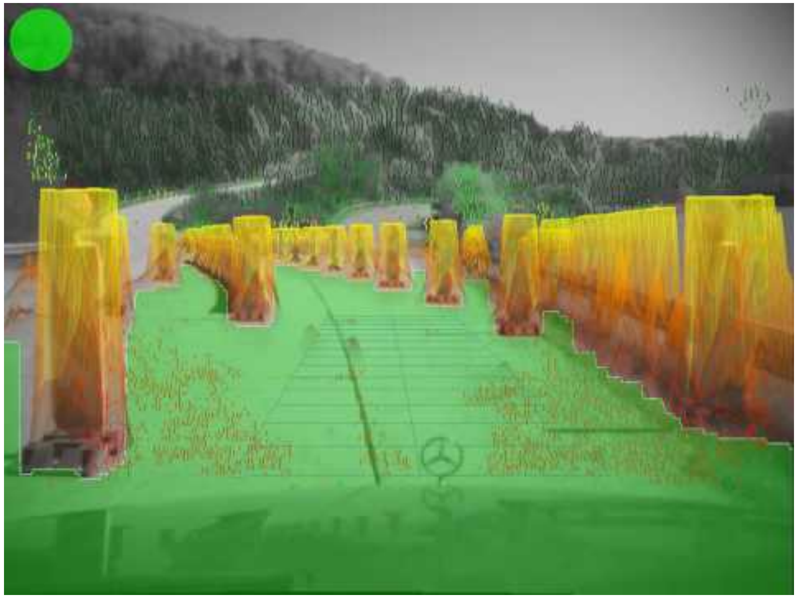
5. Stixel Extraction

Youtube video: https://www.youtube.com/watch?v=FR_mIY34IW0



OUTLINE

- What is Free-Space Estimation?
- Challenges
- Occupancy Grid Mapping Algorithm
- Approaches to the Problem
  - Dense Stereo
  - Occupancy Grid
  - Free-Space Computation by DP
  - Height Segmentation
  - "Stixel" Extraction
- **Future Work**



Source: https://www.youtube.com/watch?v=tiRi3I_wbNk

Future Work

- Apply a tracking for stixels based upon the principles of 6D-Vision, where 3D points are tracked over time and integrated with Kalman filters. The 3D points are tracked over time and integrated with Kalman filters. The integration of stixels over time will lead to further improvement of the position and height.
- Models other than occupancy grid?

Future Consideration

Neural Networks?

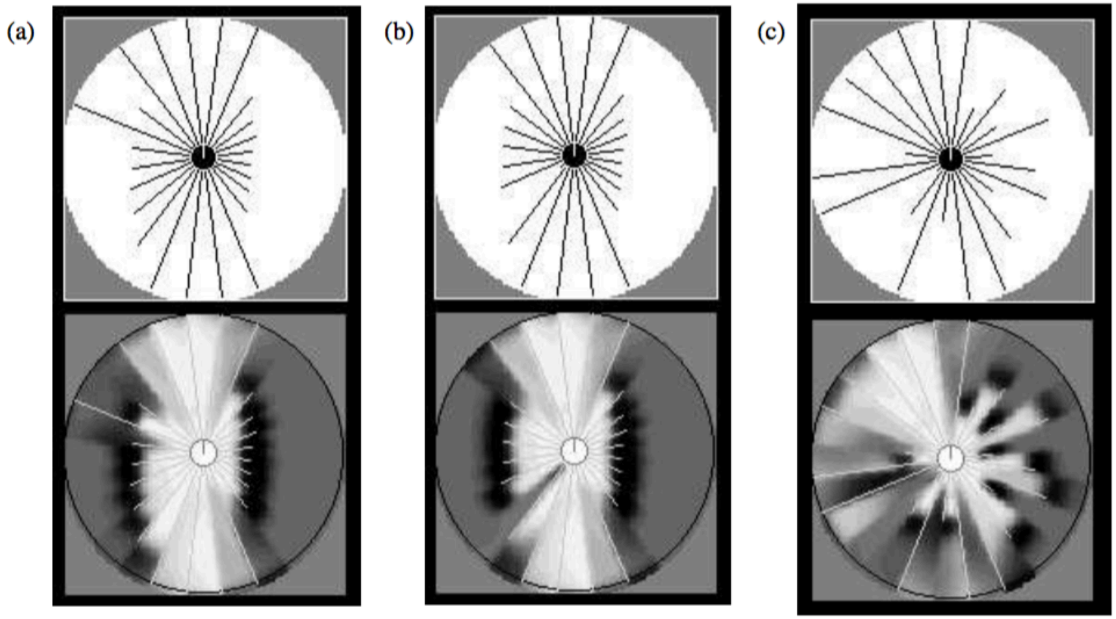


Figure 9.6 Sensor interpretation: Three sample sonar scans (top row) and local occupancy maps (bottom row), as generated by the neural network. Bright regions indicate free-space, and dark regions indicate walls and obstacles (enlarged by a robot diameter).

Thanks!

HAO WU

All the formula and pictures are from the following sources:

- Chapter 9 of Probabilistic Robotics Book, S. Thrun, W. Burgard, D. Fox
- Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming In Workshop Dynamical Vision ICCV 2007, H. Badino, U. Franke and R. Mester
- The Stixel World - A Compact Medium Level Representation of the 3D-World DAGM 2009, H. Badino, U. Franke and D. Pfeiffer