
HE
IG
VD

Intelligence Artificielle pour les systèmes autonomes (IAA)

Modular pipeline: Scene parsing

Prof. Yann Thoma - Prof. Marina Zapater

Février 2024

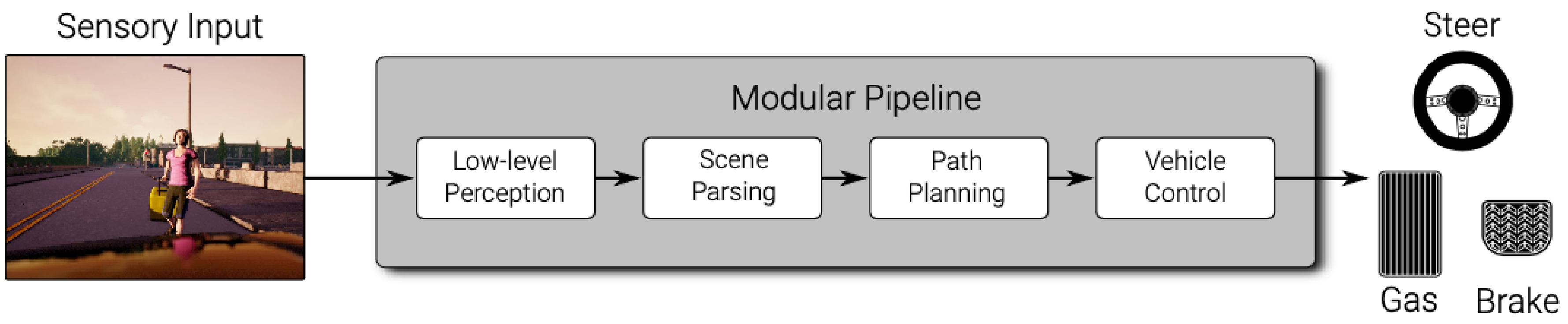
Basé sur le cours du Prof. A. Geiger

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale

ReDS
Reconfigurable & embedded
Digital Systems

Modular Pipeline

Reminder of main blocks



→ Vehicle control

→ Low-level perception : Odometry, SLAM and global localization

→ **Scene Parsing**

→ Path planning

Summary

Today’s lesson

→ **Road and Lane detection**

→ Free space estimation

→ Optical Flow and Scene Flow

Road and lane detection

Representations

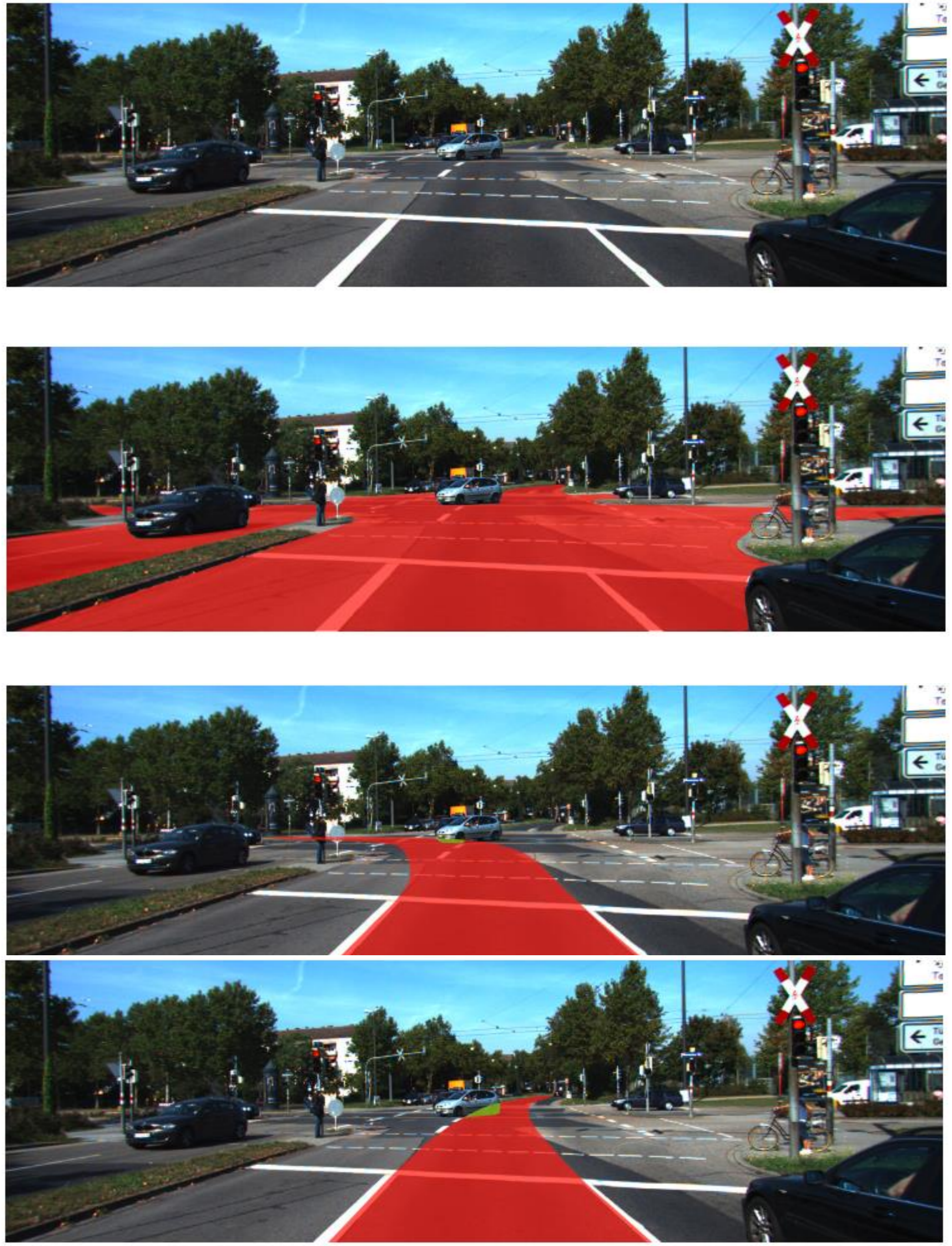
→ Navigate without detailed global map by ”sensing” drivable areas in the car vicinity

→ Road segmentation:

- Classify each pixel in the image as road or non-road

→ Driving corridor prediction:

- Estimate corridor ahead
- Mark obstacles in green within the corridor
- Multiple corridors!



Road and lane detection

Representations

→ Lane marking detection

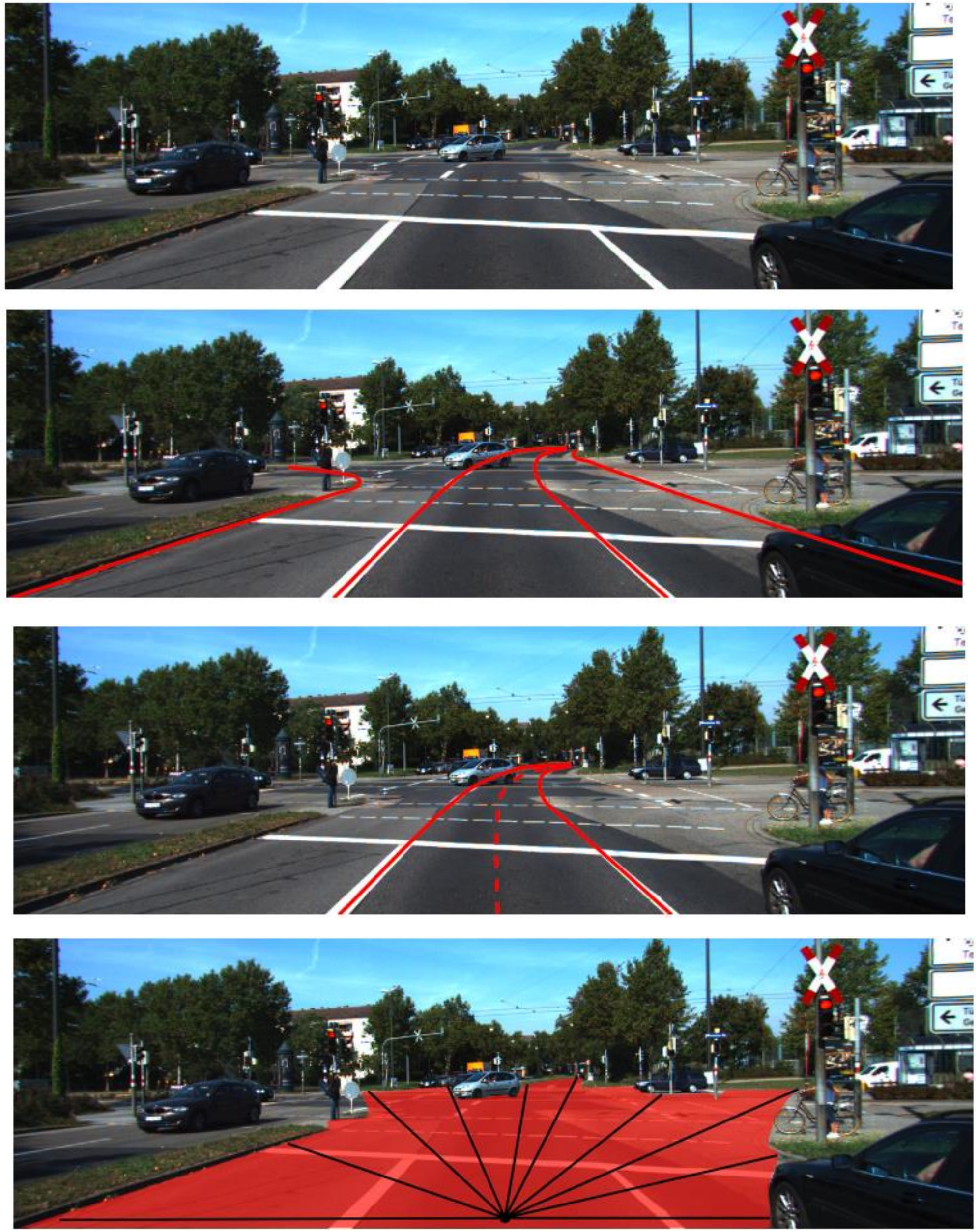
- Detect lane marking and fit a parametric model

→ Lane detection

- Group two lane markings into one lane
- Provide information on the centerline (dashed)

→ Freespace estimation

- Estimate places that can be reached without collision

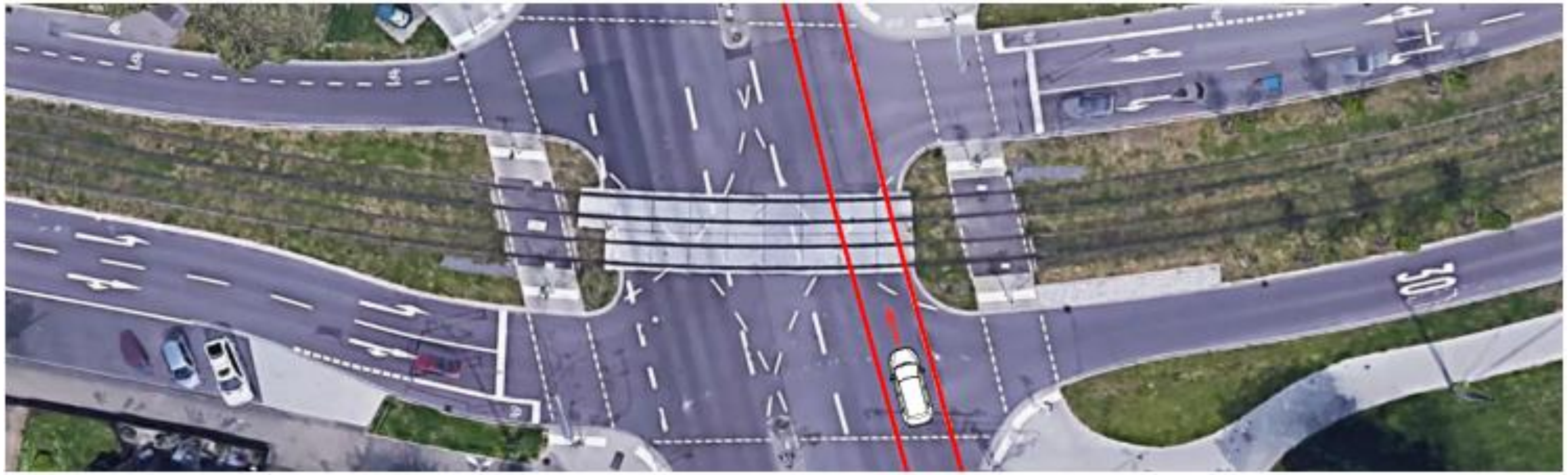


Road and Lane detection

2D vs. 3D representation

→ Estimating quantities in the 2D image domain is hard and not very useful

→ Better to map into 3D or Bird’s Eye View (BEW) where vehicle is controlled



Road Segmentation

Deep Convolutional Image Segmentation

→ Use convolution, pooling and upsampling layers to predict per-pixel class label

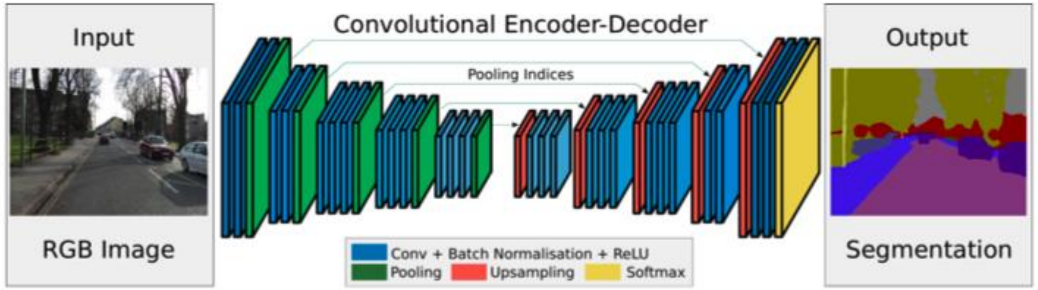
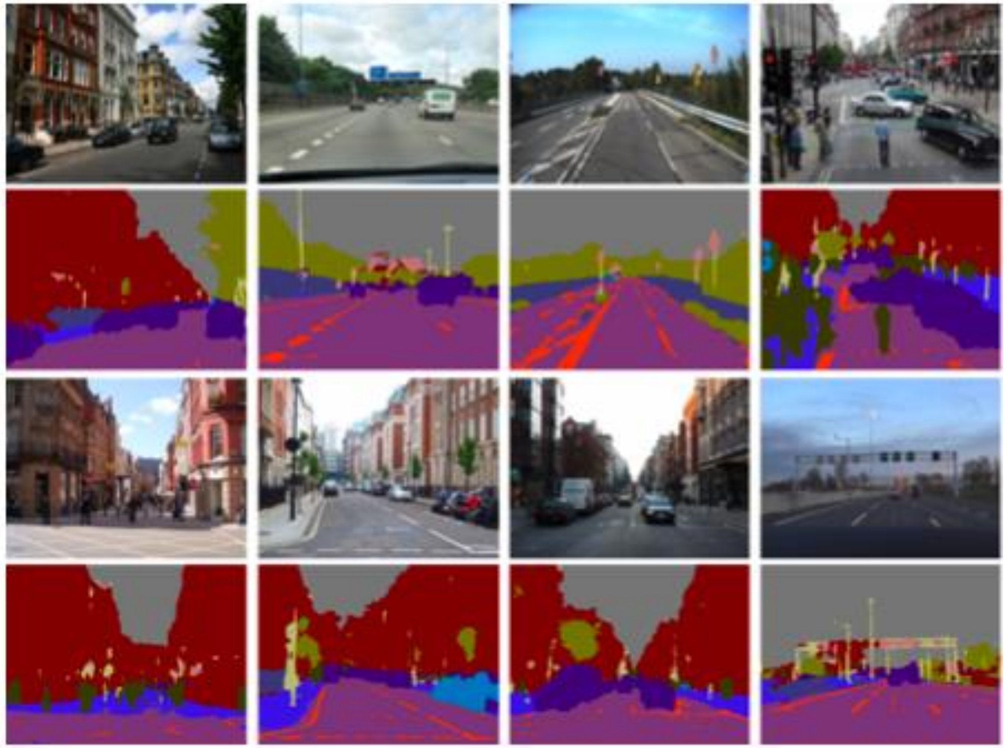


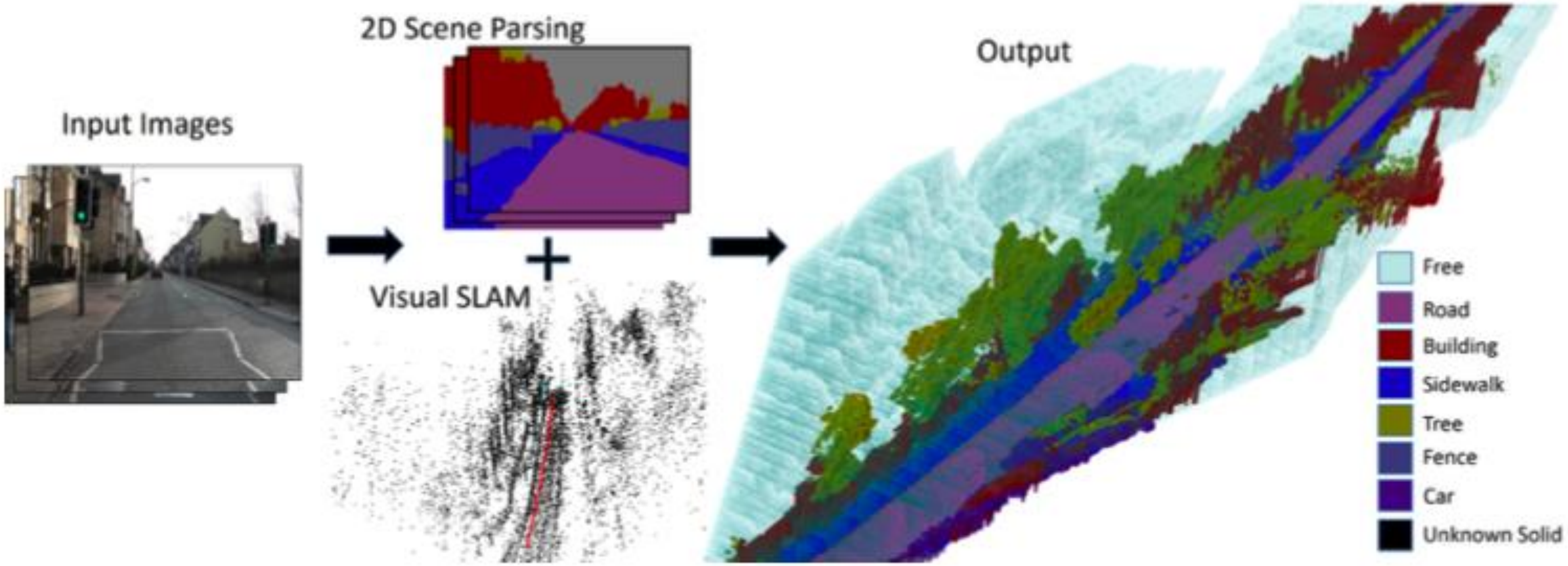
Fig. 2. An illustration of the SegNet architecture. There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification.



https://arxiv.org/pdf/1511.00561.pdf

Road Segmentation

Joint Semantic Segmentation and 3D reconstruction

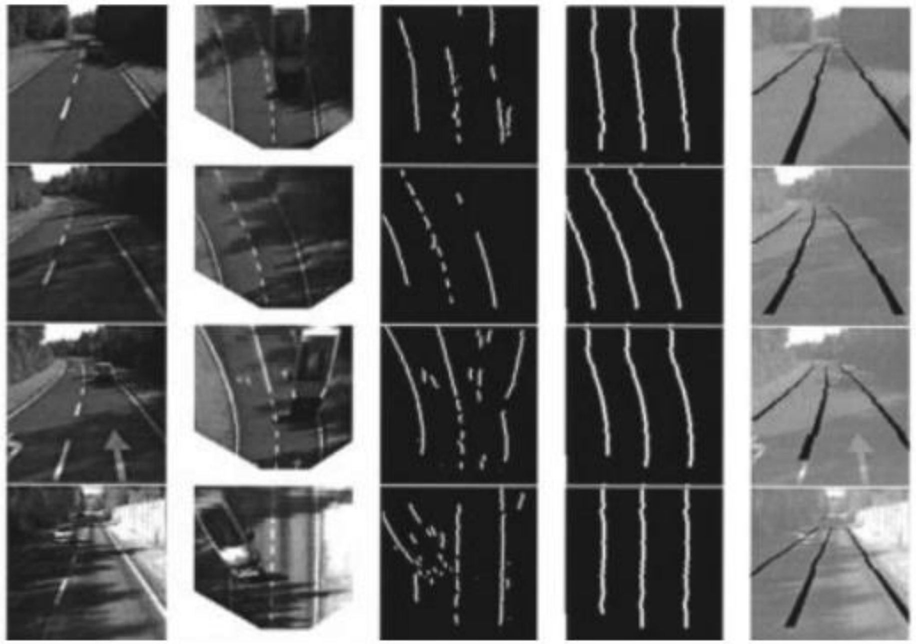


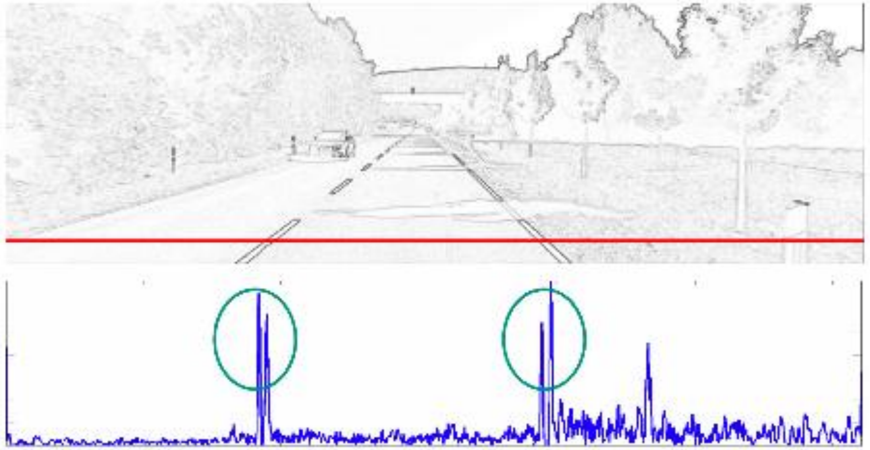
https://abhijitkundu.info/assets/pdf/HybridSFM-ECCV2014.pdf

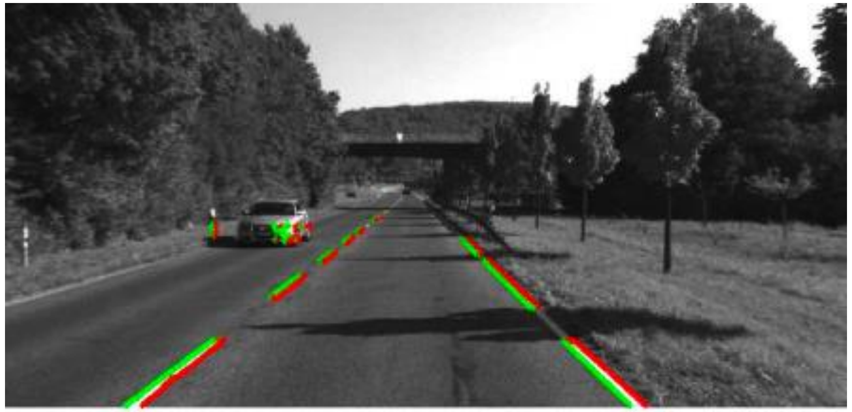
Lane marking detection

Searching for gradients along each image row

→ If depth is available: filter points not on ground plane
→ We will need to map into perspective
→ And then fit pixels into curves







Parametric Lane Marking estimation

Fit Splines into the lines we found

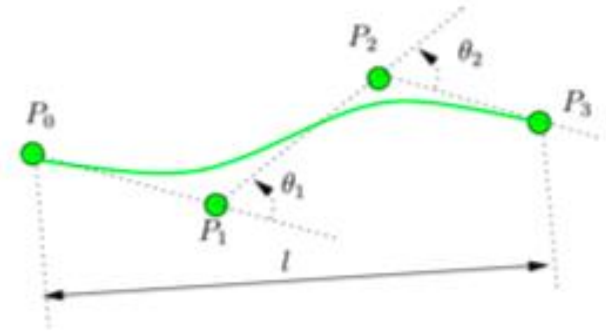


Fig. 7. Spline score computation.

```
Algorithm 1 RANSAC Spline Fitting
for i = 1 to numIterations do
    points=getRandomSample()
    spline=fitSpline(points)
    score=computeSplineScore(spline)
    if score > bestScore then
        bestSpline = spline
    end if
end for
```

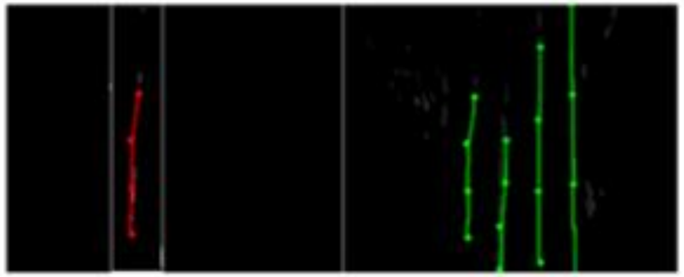


Fig. 8. RANSAC Spline fitting. Left: one of four windows of interest (white) obtained from previous step with detected spline (red). Right: the resulting splines (green) from this step

Fig. 10. Post-processing splines. Left: splines before post-processing in blue. Right: splines after post-processing in green. They appear longer and localized on the lanes.

https://arxiv.org/abs/1411.7113

Summary

Today’s lesson

→ Road and Lane detection

→ **Free space estimation**

→ Optical Flow and Scene Flow

Free Space Estimation

Problem definition and approach

→ Given depth map per frame (from stereo reconstruction), provide freespace in Bird's Eye View (BEW)

→ Useful for collision avoidance and path planning (local path planning)

→ Approach:

- Integrating disparity maps temporally (using Visual Odometry)
- Convert depth measurements into BEW occupancy map

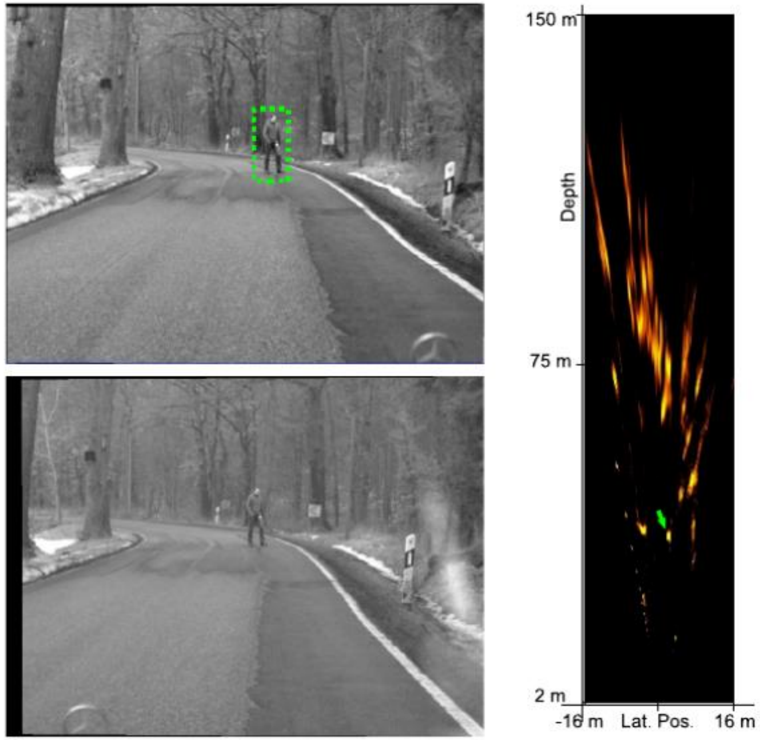
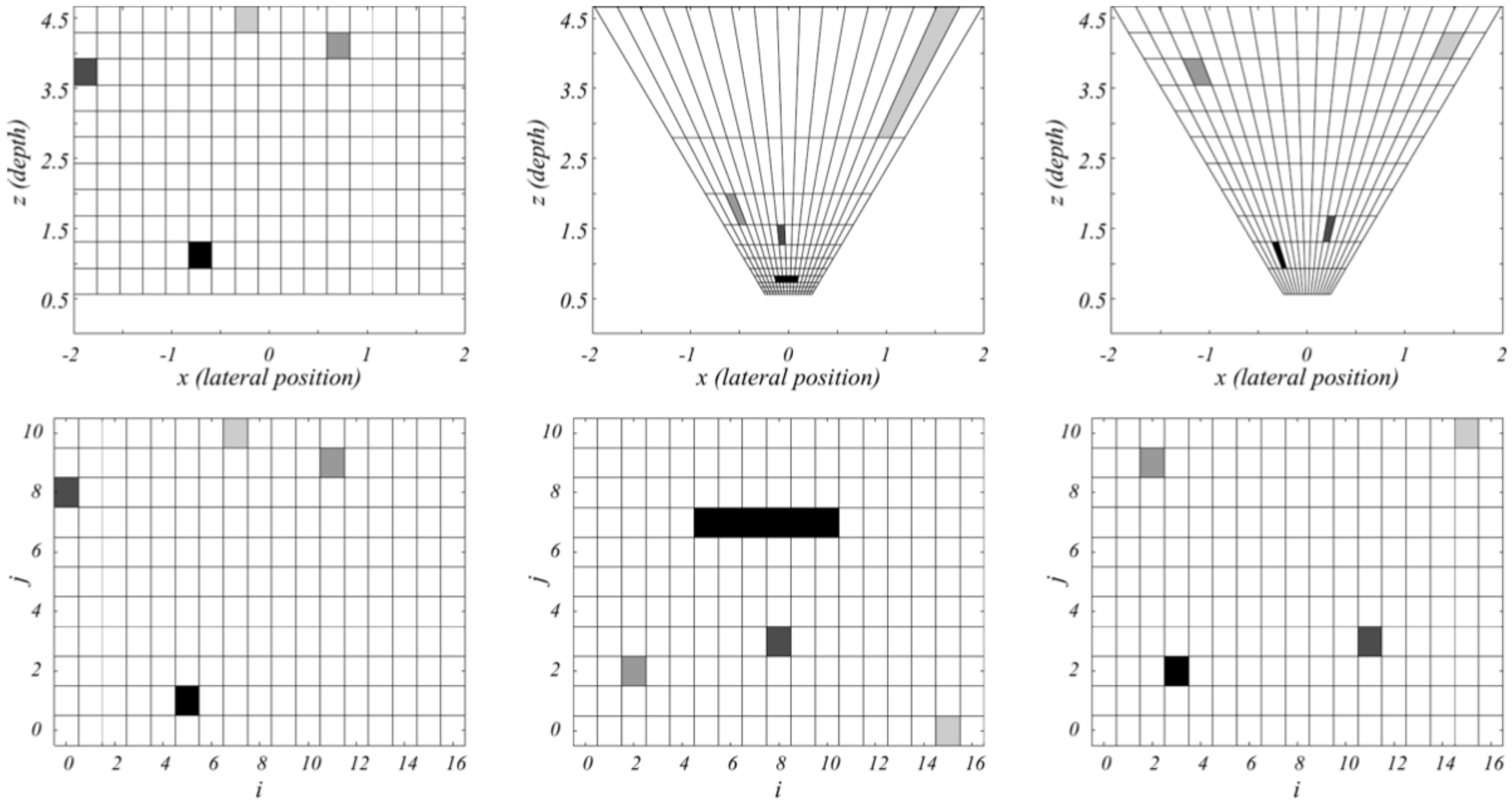


Image Cartesian

Free Space Estimation

Translation from cartesian positions to column/disparity maps



(a) Cartesian (b) Column/Disparity (c) Polar

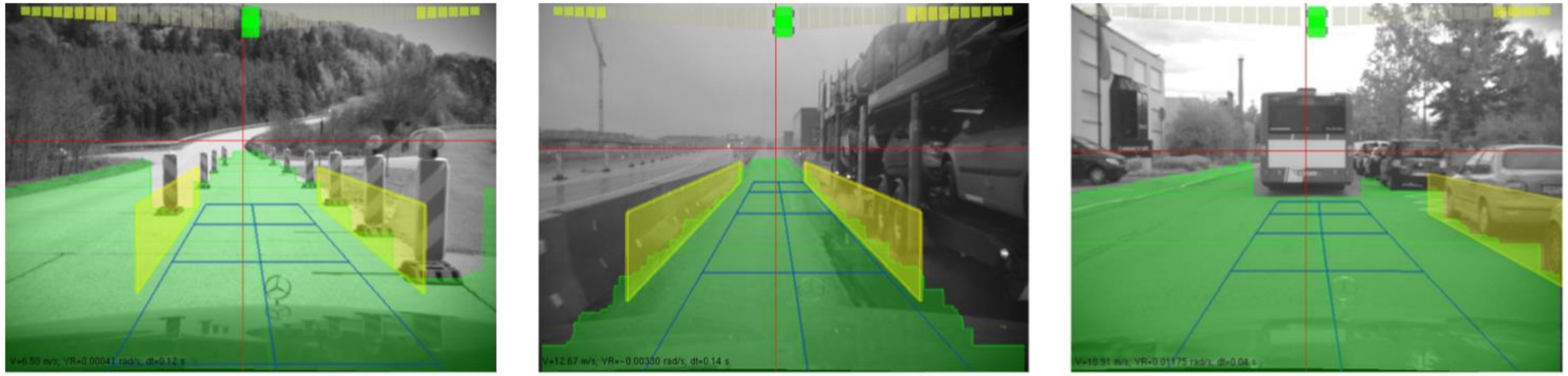
Free Space Estimation

Depth-to-occupancy conversion

→ Estimate the occupancy likelihood at a location (x,z) in the occupancy map
  - In a polar representation, where x=column, and z=depth

→ We use a kernel estimator that accumulates evidence of nearby observations
  - Optimization solved via dynamic programming

→ Height of obstacles is unknown, we only estimate freespace



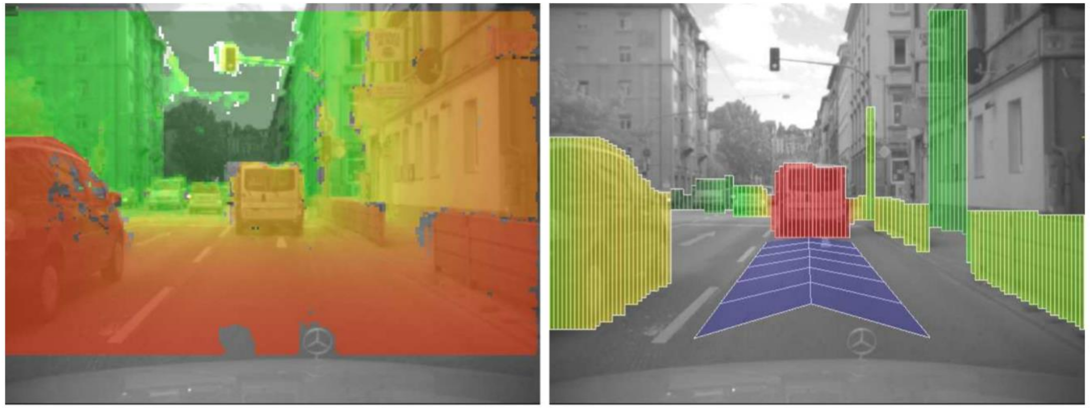
(a) Highway. (b) Freeway. (c) Downtown.

http://vision.jhu.edu/iccv2007-wdv/WDV07-badino.pdf

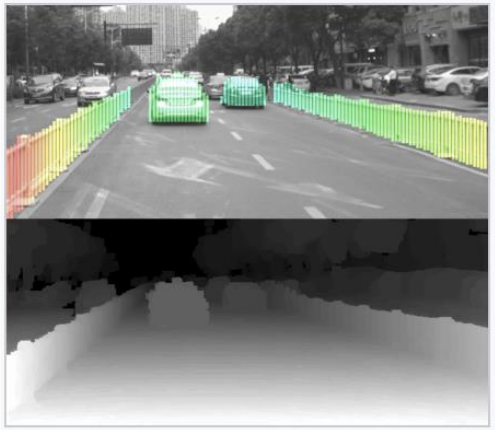
Moving to the Stixel world

Stixel: superpixel representation

→ Stixel” a stik that approximates an obstacle as a vertical line in the scene



(a) Dense disparity image (SGM result) (b) Stixel representation



Top: Grayscale input image with stixels superimposed to it, with colour denoting depth (from red denoting closer, to blue denoting farther). Bottom: Dense disparity map, with brighter intensity denoting higher values of disparity (lower depth), darker intensity denoting lower values of disparity (higher depth), and black denoting invalid disparity.

https://link.springer.com/chapter/10.1007/978-3-642-03798-6_6

Summary

Today’s lesson

→ Road and Lane detection

→ Free space estimation

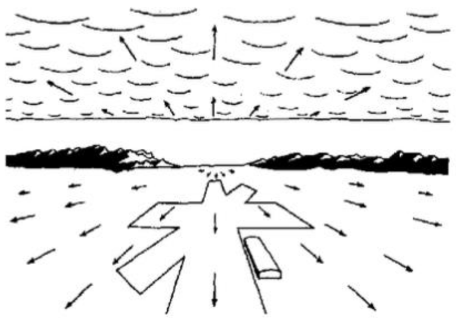
→ **Optical Flow and Scene Flow**

Optical Flow

Apparent motion of objects in 2D

→ Optical flow is the apparent motion of objects in a scene caused by the relative motion between observer and scene

→ Knowing past/current motion allows to make predictions on flow
- Practical use: predict the position of a vehicle 1sec into the future
- Careful! Predictions are made in image space, not in the 3D space!
- 2D motion can either be due to observer moving, or to object moving

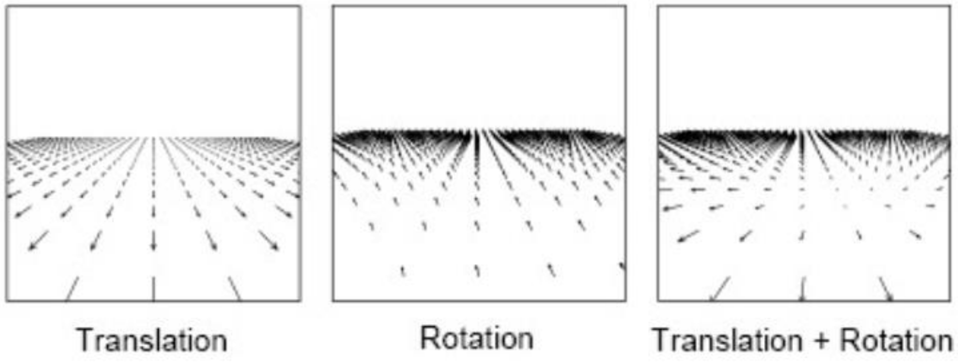


Optical Flow

And aperture problem

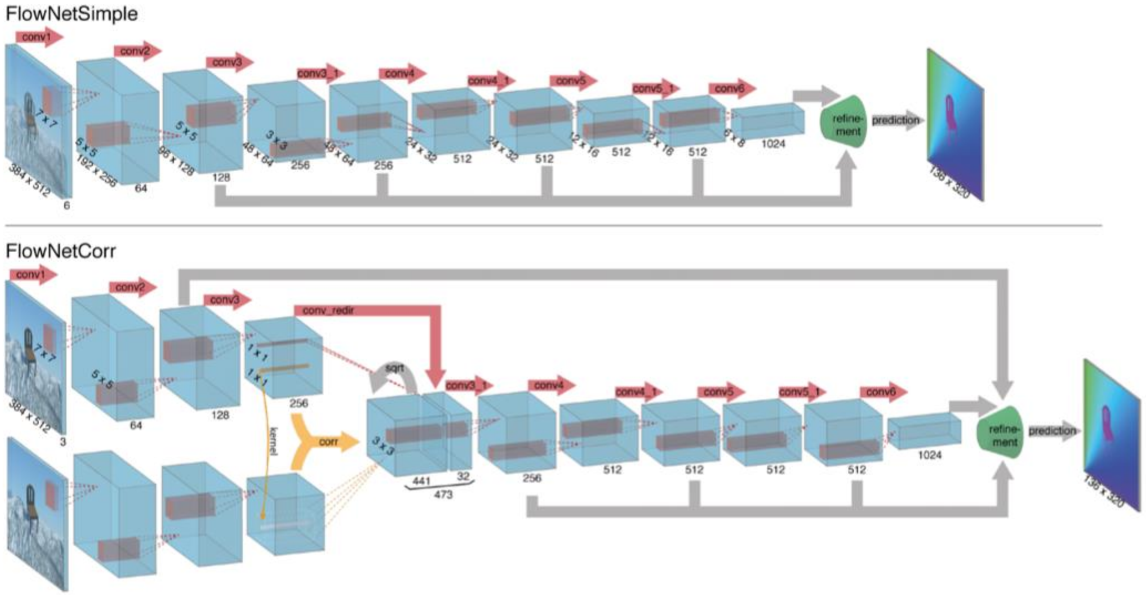
→ The optical flow tells us information on:
- 3D structure
- Motion of object
- Motion of observer

→ A single observation
(even coming from two cameras)
is not enough to determine optical flow



Deep Learning for Optical Flow Estimation

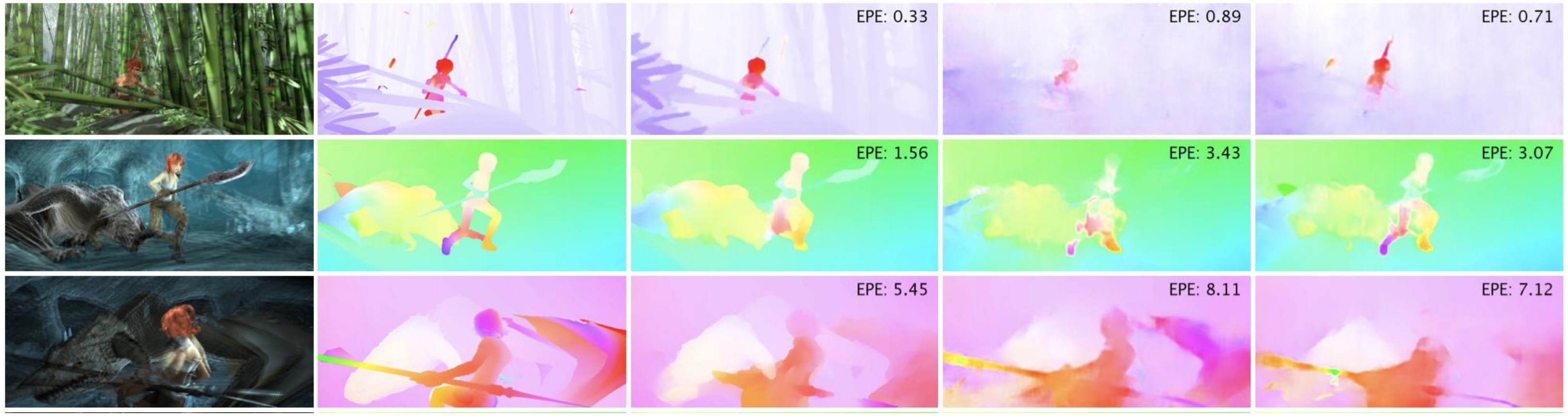
FlowNet



→ Predicting Flow with CNNs

→ Two networks:
- FlowNetSimple
- FlowNetCorr

→ KITTI dataset



https://arxiv.org/pdf/1504.06852.pdf

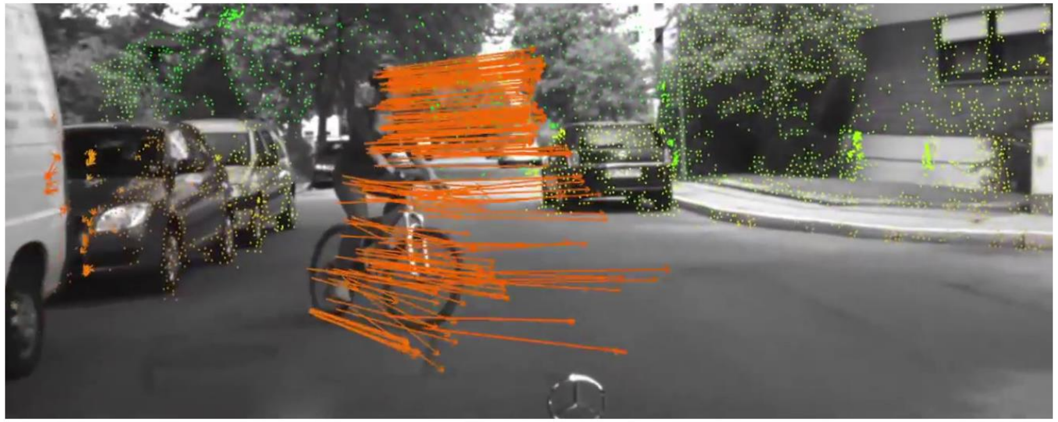
Scene Flow Estimation

Optical Flow Estimation in 3D (not in 2D)

→ In practice Optical Flow (2D motion estimation) is not enough

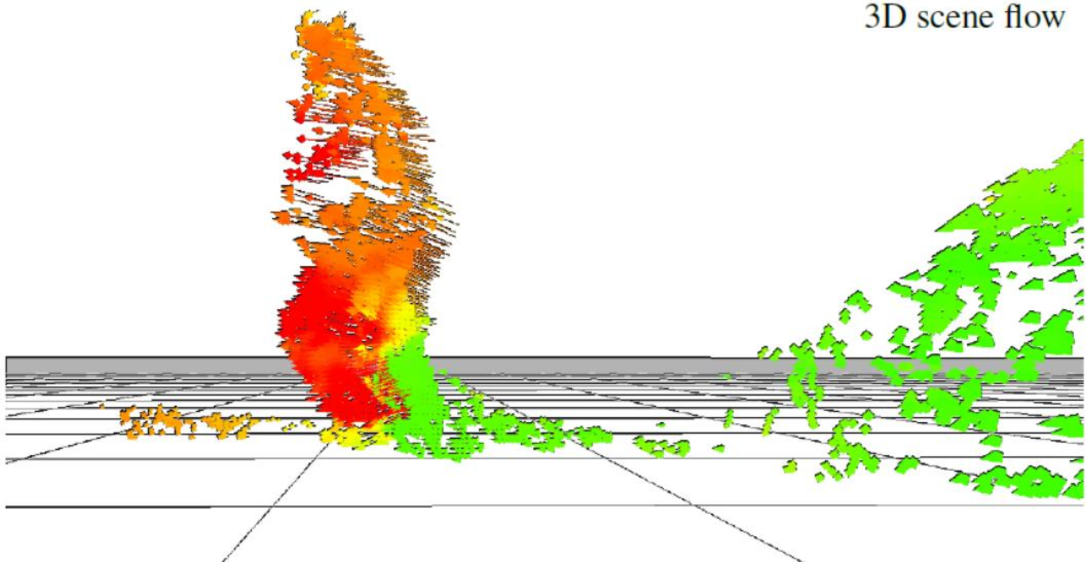
→ We need motion in 3D → Scene Flow

→ “Scene flow is a dense 3D vector field defined for point of every surface in the scene”



left image at time t

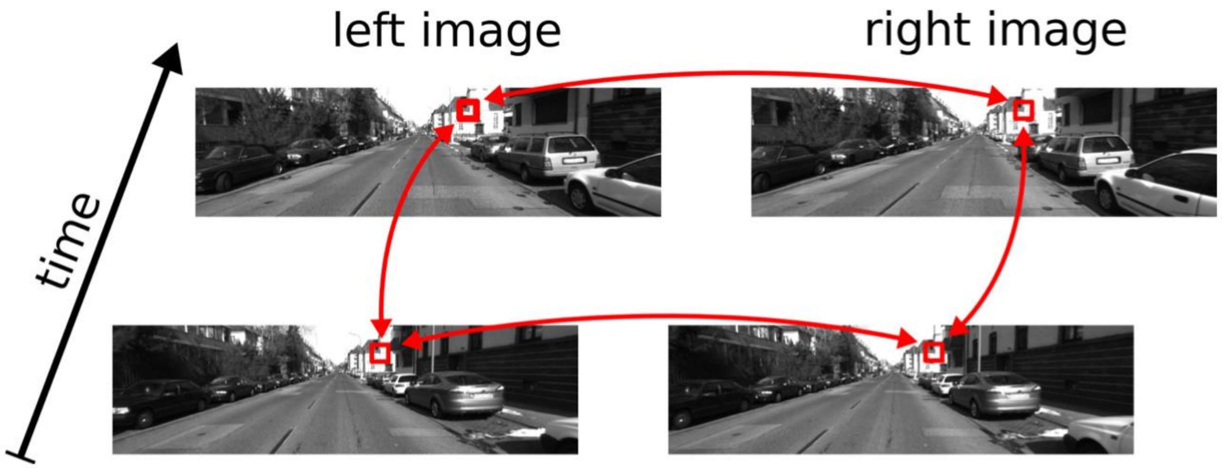
left image at time $t+1$



3D Scene Flow

Scene flow is the instantaneous 3D motion of evey point

→ Optical flow is the 2D projection of the scene flow into an image
→ To estimate scene flow, we require at least 4 images (or 2 images with depth)
→ Combining the problem of stereo and optical flow



3D Scene Flow estimation

Using rigid body concepts

→ A recurring idea: exploiting the fact that scene is composed of rigidly moving objects

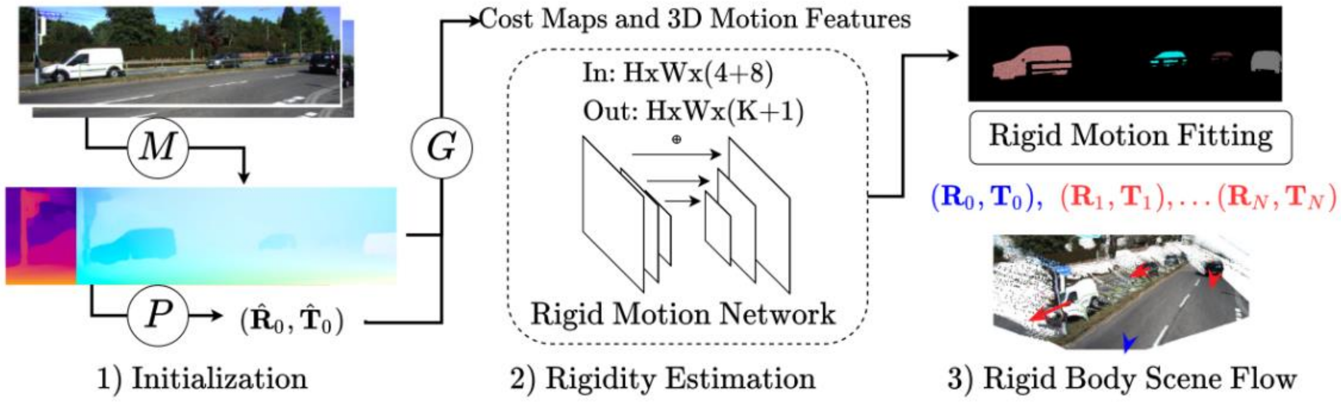


Figure 4: We detect and estimate rigid motions in three steps: First, depth and optical flow are computed using off-the-shelf networks (M) and camera motion is estimated by epipolar geometry (P) given two frames. Then, rigidity cost maps and rectified scene flow are computed (G) and fed into a two-stream network that produces the segmentation masks of a rigid background and an arbitrary number of rigidly moving instances. Finally, we fit rigid transformations for the background and each rigid instance to update their depth and 3D scene flow.

→ Ranked 1st on KITTI Scene Flow Benchmark
http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php

https://arxiv.org/abs/2101.03694

HEIG VD

REDS
Institut
Reconfigurable
and Embedded
Digital Systems

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale