Fast object detection

Petar Ivanov
17.09.2012

Problem statement

Real-time object segmentation in stereo video

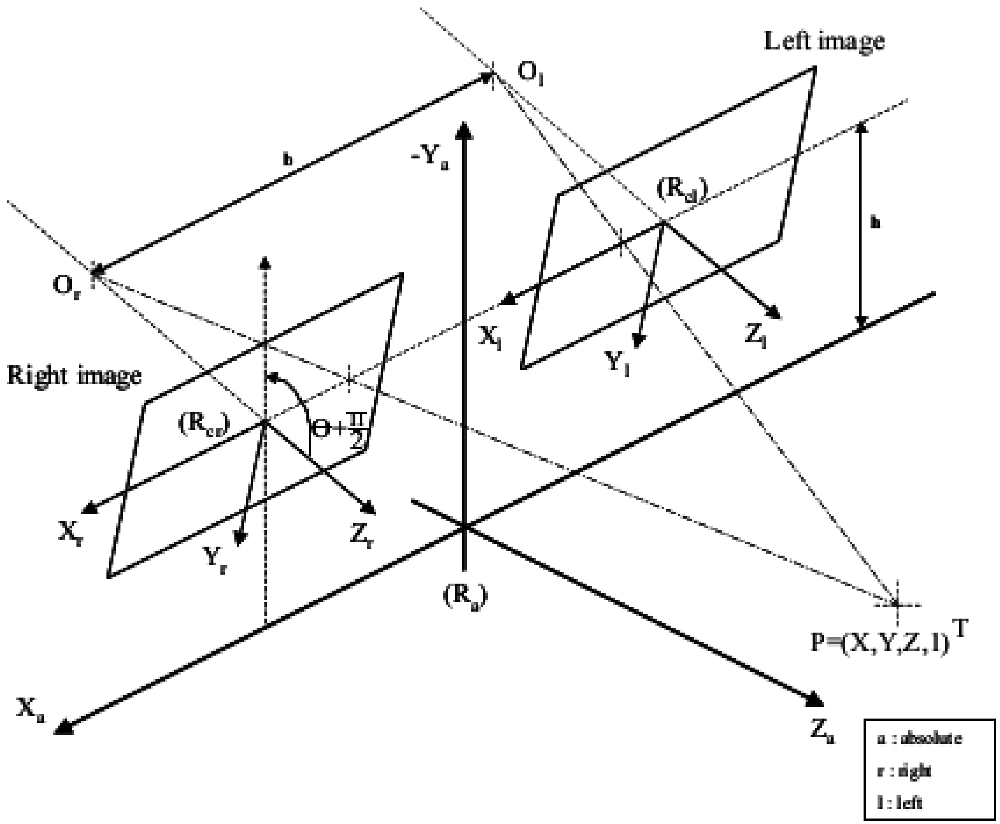
<u>Input</u>

Two moving video streams (left and right)
positioned on a car (so a road is seen)

<u>Output</u>

Sparse space segmentation
and dense time segmentation
on one of the streams (left)
of the nearest objects (cars/pedestrians)
in all directions

Assumptions

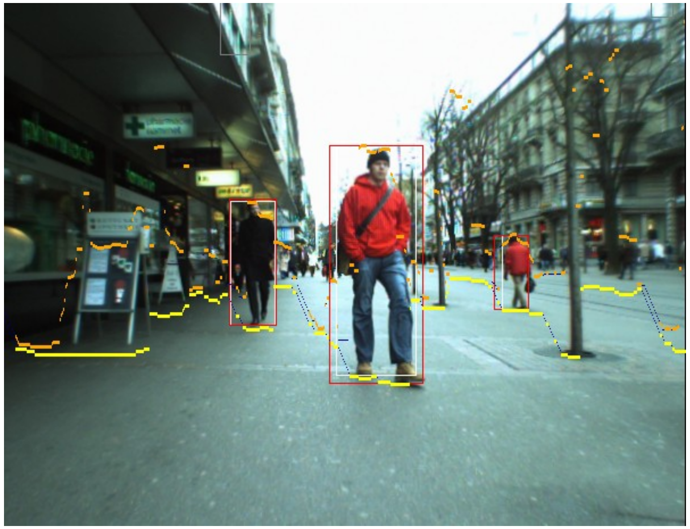


- Rectified cameras
- Calibrated cameras
- Horizontal cameras

R. Labayrade et al., “Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through ”V-disparity” Representation”, 2002

The naïve way

- Choose a random algorithm (from hundreds) for disparity estimation (disparity ~ 1/depth)
- If the road is flat then we already know its disparity for every row (the cameras are calibrated)
- Every part with different disparity is an object

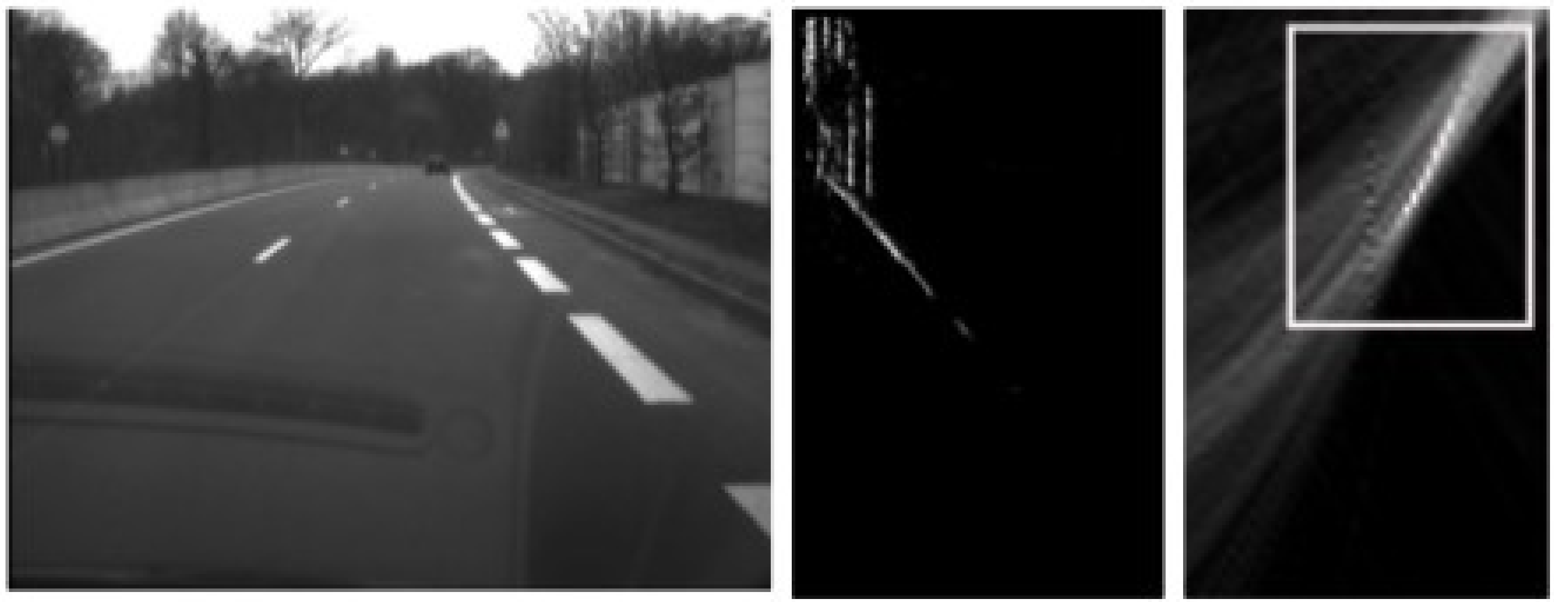


R. Benenson et al., “Stixels estimation without depth map computation”, 2011

But what if the road is not flat?

Raphael Labayrade et al.,
“Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through ”V-disparity” Representation”, 2002

Labayrade et al., “Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation”, 2002



Left image → disparity map (not shown) → V-disparity → Hough transform

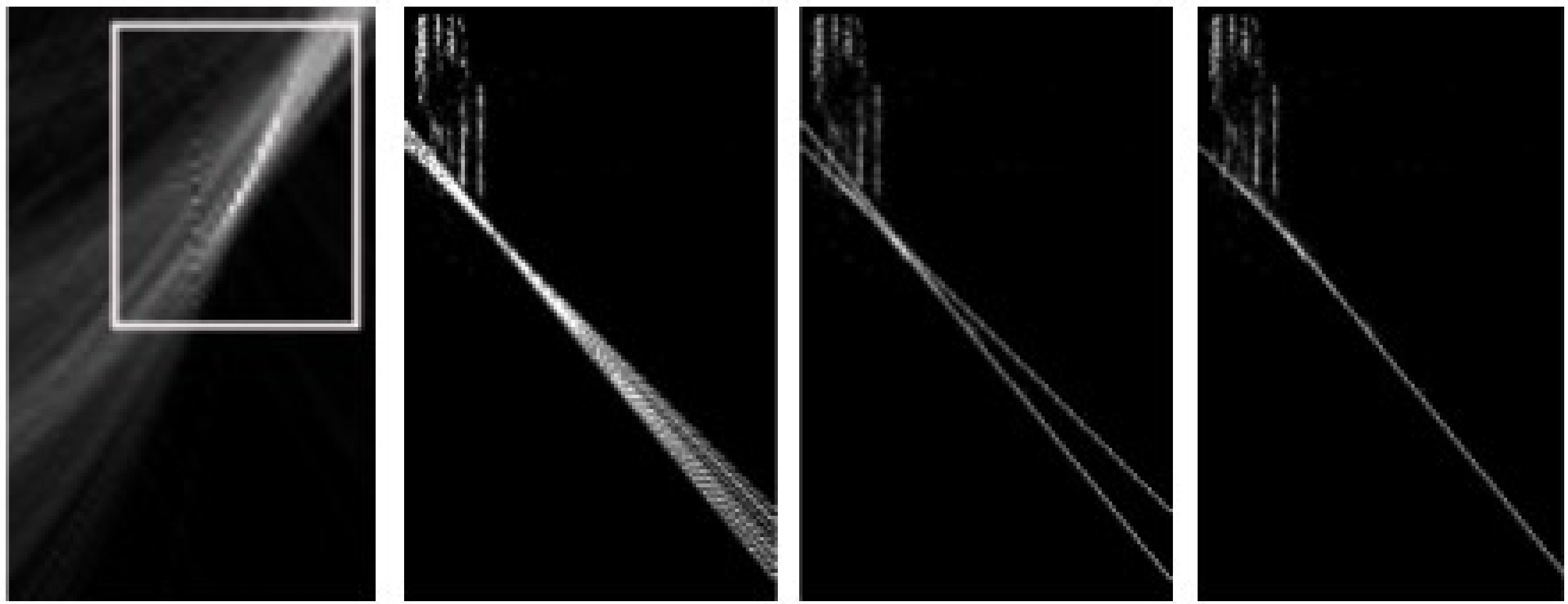
How real objects look like in V-disparity map:

- road → line (the steep determines if the road goes up or down)
- car → vertical line
- horizon → zero disparity
- trees, lines, buildings, etc. → diffuse areas

Where is the road?
Miss solution 2002
(> 300 citations)

Labayrade et al., “Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation”, 2002

Labayrade et al., “Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation”, 2002



Hough transform → k points ⇔ k lines → longitudinal profile

Choosing between <u>min</u> and <u>max</u> profile depends on whether the road is going upwards or downwards

Labayrade et al., “Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation”, 2002

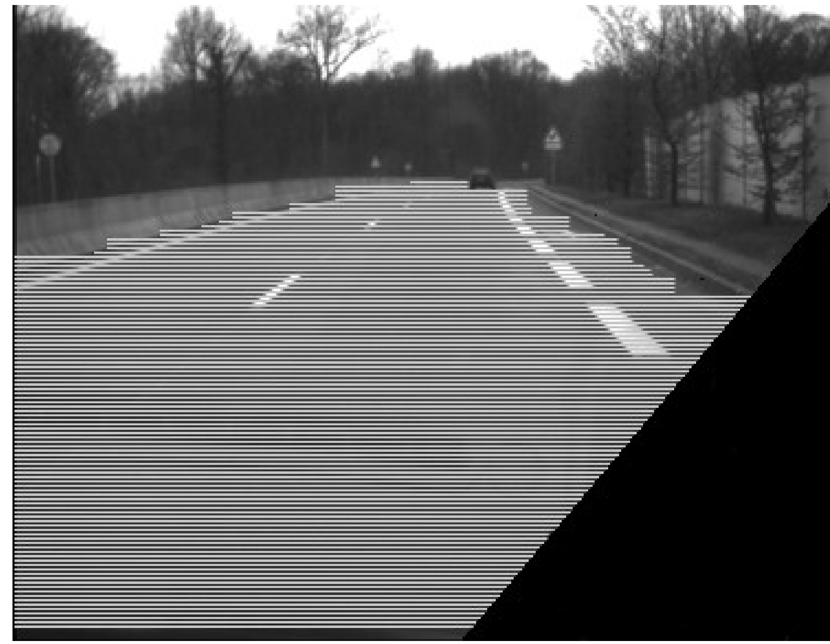
disparity map →

{ v-disparity & Hough transform }

→ longitudinal profile of the road

Then computing the obstacle areas

Labayrade et al., “Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation”, 2002



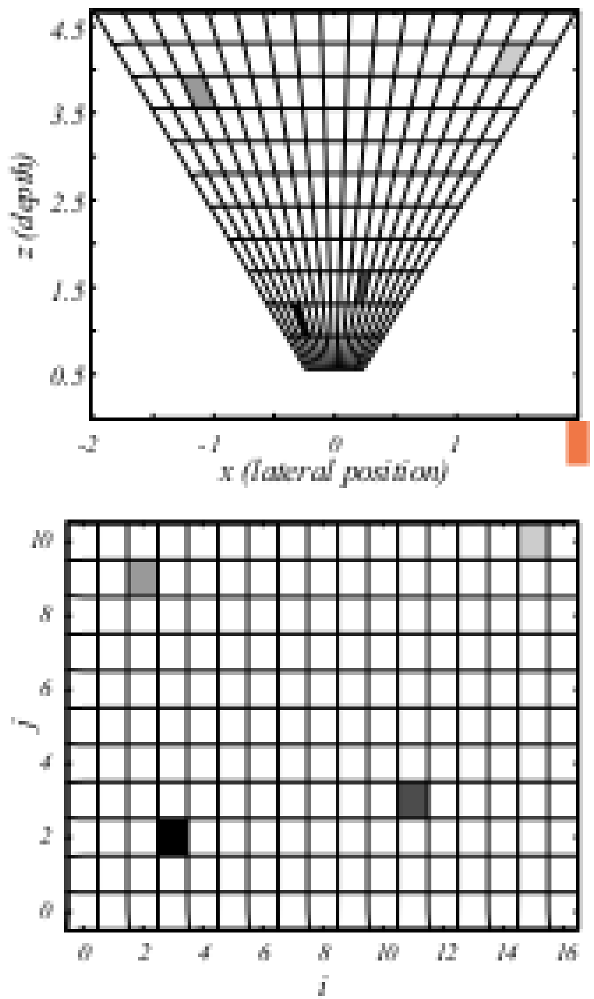
Using some bottom-up greedy the free space is constructed

Overall

Labayrade et al., “Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation”, 2002

- 25 Hz on 380×289 frames (in 2002) but a disparity map is needed (**slow**!)
- Greedy free space estimation (**a global optimization is possible**!)

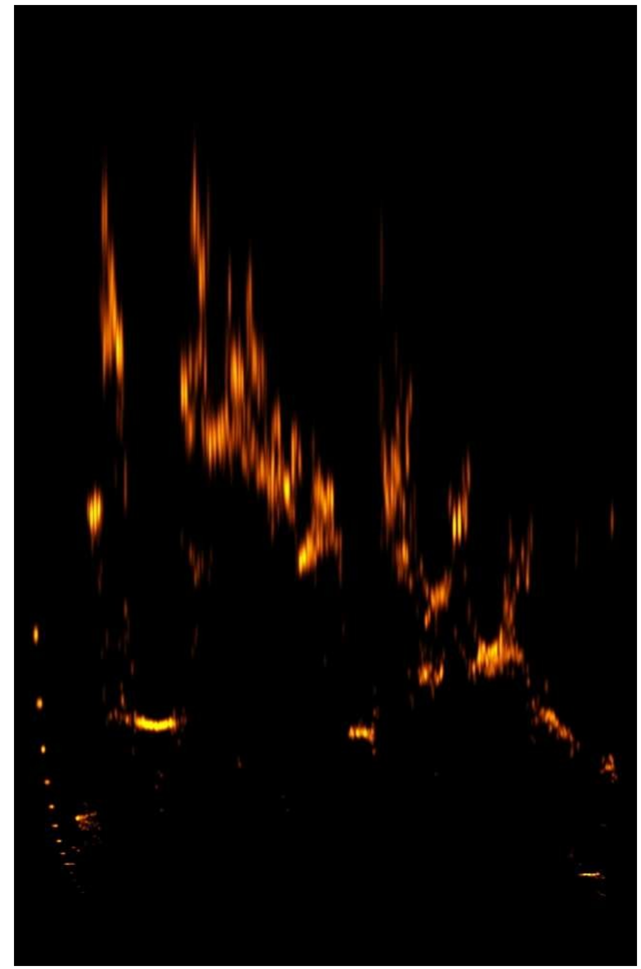
H. Badino et al., “Free space computation using stochastic occupancy grids and dynamic programming”, 2007



(c) Polar

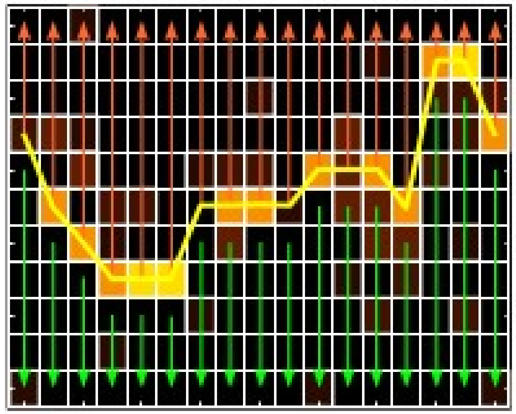
Occupancy grid is a 2D projection of the objects which shows the probability that a cell is occupied

Polar occupancy grid

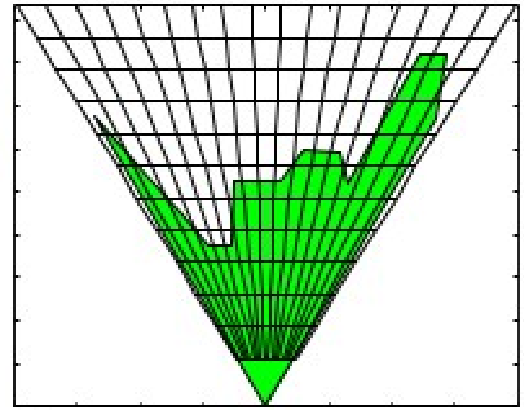


H. Badino et al., “Free space computation using stochastic occupancy grids and dynamic programming”, 2007

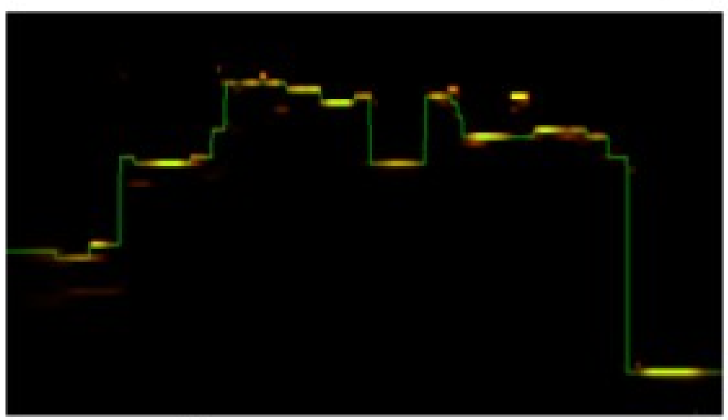
DP for estimating free space using the occupancy grid



(a) Polar Occupancy Grid.



(b) Corresponding free space in world coordinates.



(c) Segmentation result.

(d) Freespace.

H. Badino et al., “Free space computation using stochastic occupancy grids and dynamic programming”, 2007

Kubota et. al, “A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems”, 2007

Fig. 5. Image remapping scheme

65fps on VGA (640x480) on 3.6GHz

Kubota et. al, “A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems”, 2007

Algortihm

1) Estimate road plane parameters using v-disparity image
2) Calculate a summed disparity space image (DSI) over edges (light-invariant): DSI represents the matching score between the reference and the target images with respect to disparity and horizontal position. Matching score: SSD and SAD rely on constant luminance assumption so only edges are used (Canny detector)
3) Find the best path in the DSI which represent the boundary between the road and obstacles

Kubota et. al, “A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems”, 2007

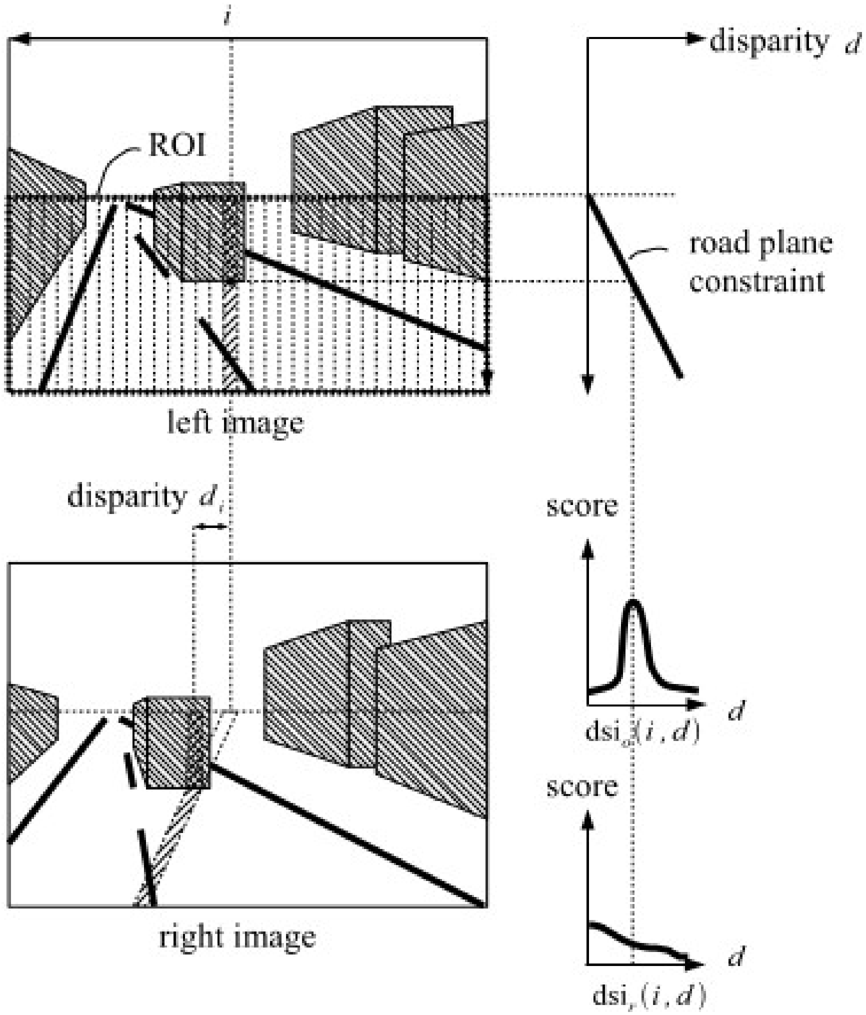


Fig. 8. Matching score calculation

Estimate road plane parameters using v-disparity image

Kubota et. al, “A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems”, 2007

Summed disparity space image (DSI) over edges (light-invariant)

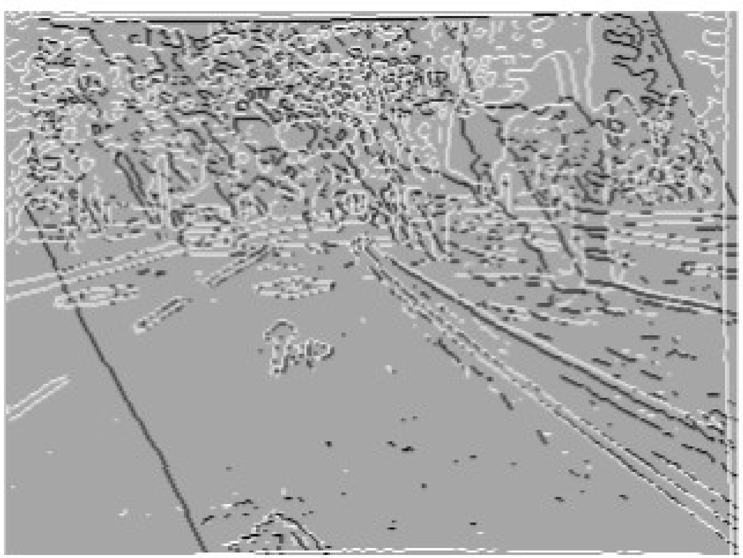


Fig. 7. Edges of a left image and the right image remapped with the estimated parameters

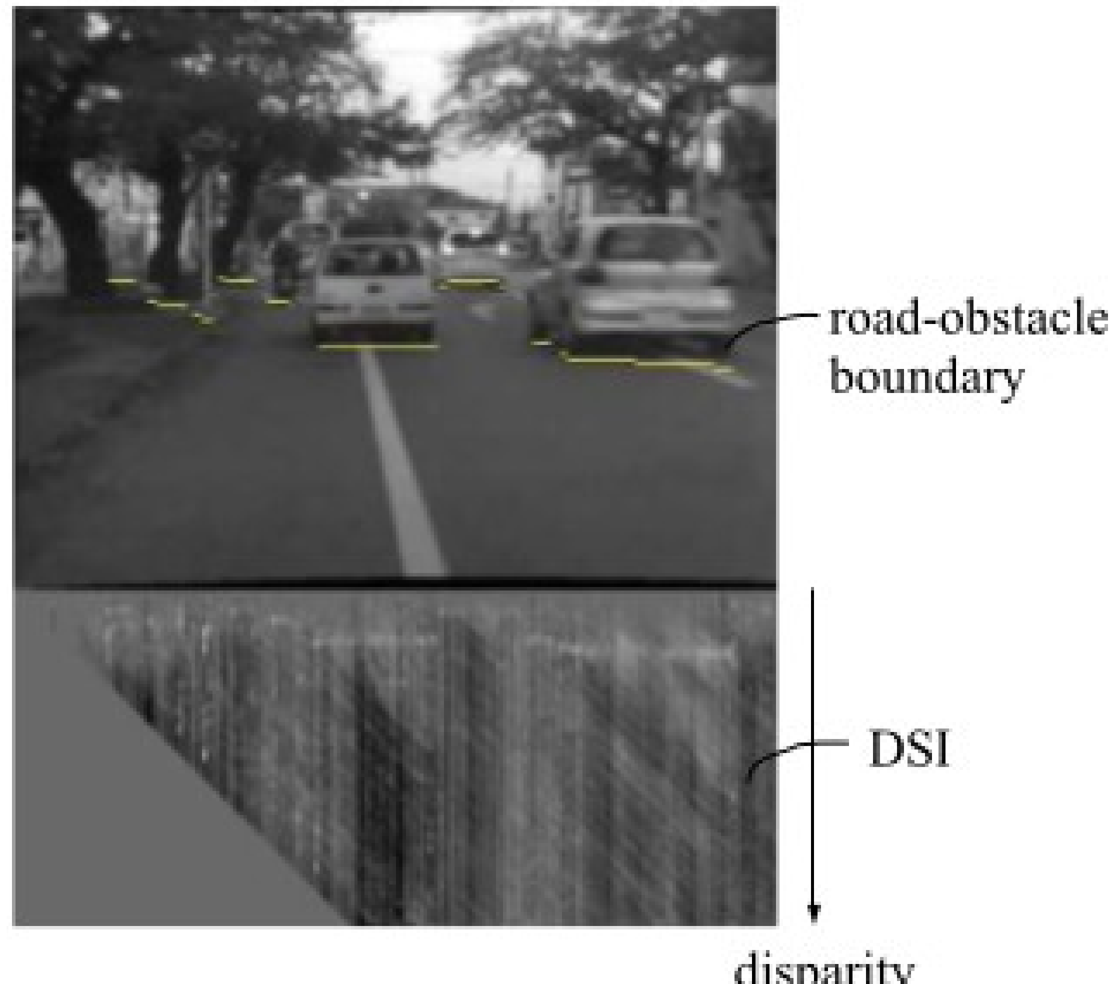


Fig. 10. DSI and road-obstacle boundary

Kubota et. al, “A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems”, 2007

The DSI is calculated by repeating the following procedure

for every i and j:

1) Calculate the y-coordinate of the hypothetical road-obstacle boundary from the disparity value j and (2).
2) Calculate the matching score DSI_r (i, j) for the road region (pixels below the boundary) of the i-th column with the correspondence given by (1).
3) Calculate the matching score DSI_o (i, j) for the obstacle region (pixels above the boundary) of the i th column where the correspondence is given by horizontal translation of j pixels.

Kubota et. al, “A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems”, 2007

Global Optimization Using Dynamic Programming

$$
\begin{aligned}
M_1(d_1) &= m_1(d_1), \\
M_i(d_i) &= m_i(d_i) \\
&+ \max_{d_{i-1}}\{M_{i-1}(d_{i-1}) - c_i(d_i, d_{i-1})\},
\end{aligned}
$$

$$
m_i(d_i) = \mathrm{dsi}_r(i, d_i) + \mathrm{dsi}_o(i, d_i)
$$

$$
c_i(d_i, d_{i-1}) = \begin{cases} \infty & for\ d_i < d_{i-1} - 1 \\ \mathrm{dsi}_o(i, d_i) & for\ d_i = d_{i-1} - 1 \\ 0 & for\ d_i > d_{i-1} - 1 \end{cases}
$$

no partial path
can have slope any
steeper than 45
degree toward
upward-left

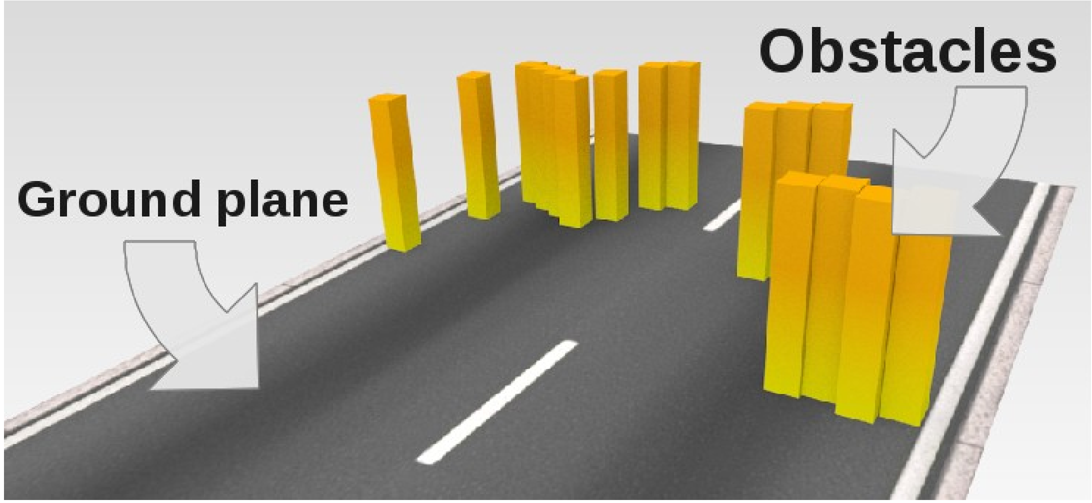
WARNING: 縦書き used (<u>not</u> 横書き !)

Kubota et. al, “A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems”, 2007

Multiresolution scheme: bring the max resolution by exponentially increasing steps:

- the disparity recomputation is limited to some interval around the previous best disparity
- the DSI images are not recomputed if they were high enough on the previous step

Hernan Badino et al.,"The Stixel World – A Compact Medium Level Representation of the 3D-World", 2009



Stixel ≈

Sticks above the ground in the image

- The stixel model consists of
  - ground plane
  - distance to the objects
  - objects height
- Free space estimation: background subtraction + DP from *
- Stixels height estimation: DP with member function M(u,v,d)

* Free space computation using stochastic occupancy grids and dynamic programming – 2007

Hernan Badino et al.,“The Stixel World – A Compact Medium Level Representation of the 3D-World”, 2009

Building the Stixel-World

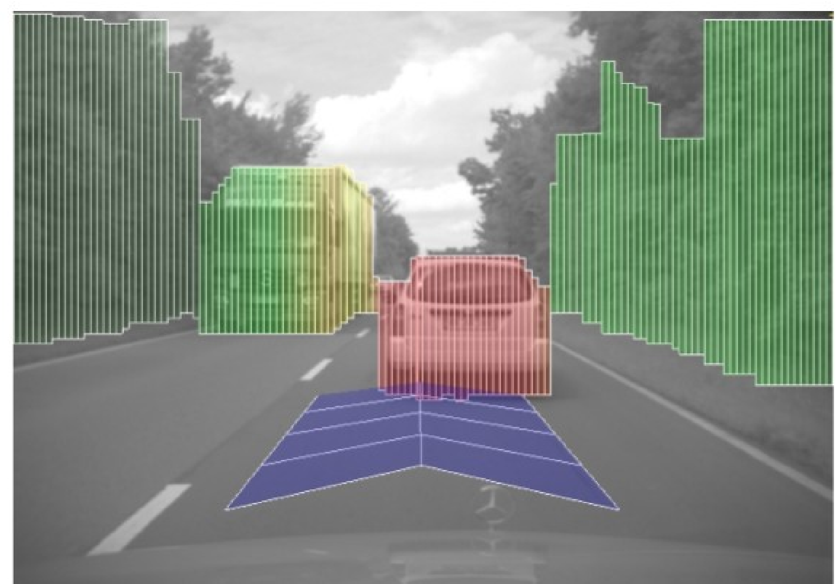
1) Dense stereo: Semi-Global Matching

2) Occupancy Grid

3) Free space estimation: background subtraction + DP

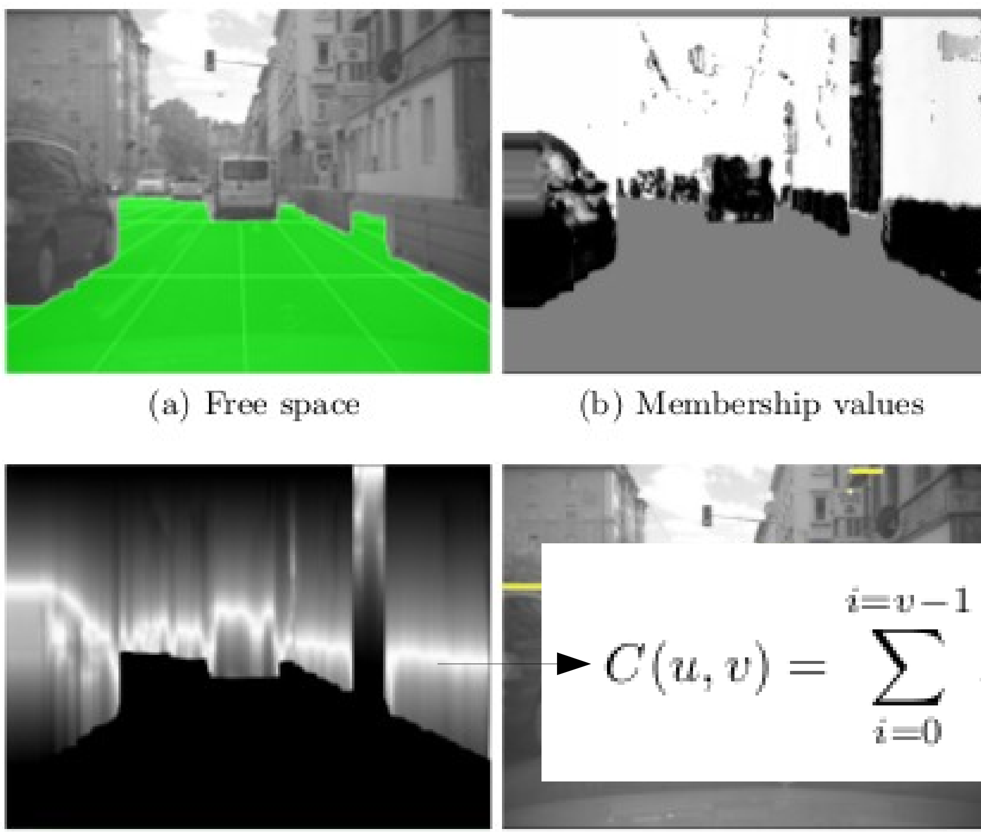
4) Height estimation: DP with member function M(u,v,d)

5) Stixel extraction



Hernan Badino et al.,“The Stixel World – A Compact Medium Level Representation of the 3D-World”, 2009

Height estimation



(a) Free space (b) Membership values

(c) Membership cost image (d) Height segmentation

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

min

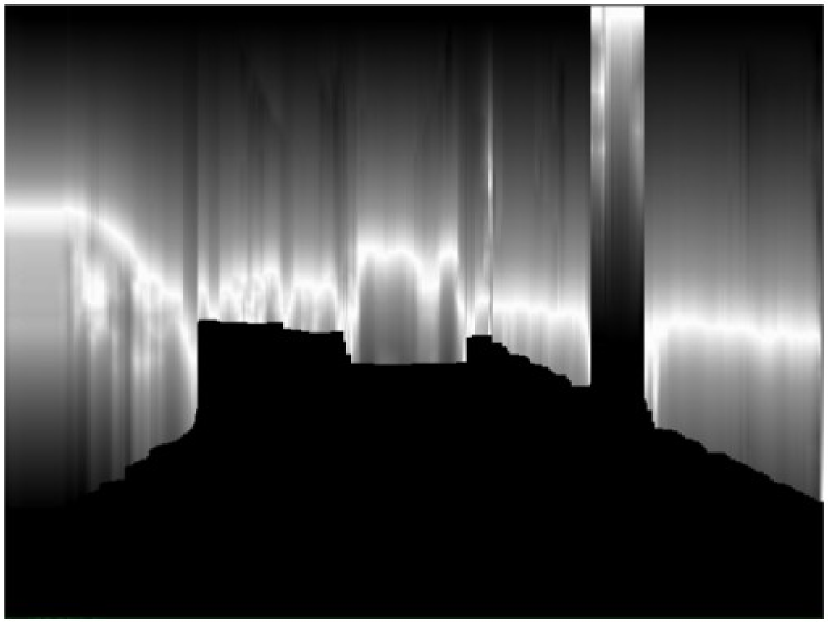
v

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

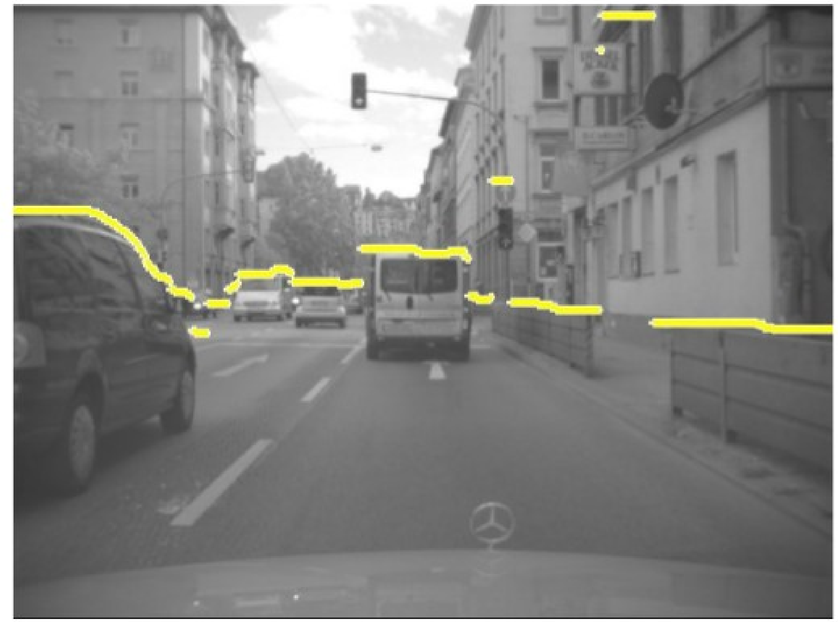
min

Fig. 3. Stixels computation: Fig. (a) shows the result obtained from free space computation with dynamic programming. The assigned membership values for the height segmentation are shown in Fig. (b), while the cost image is shown in Fig. (c) (the grey values are negatively scaled). Fig. (d) shows the resulting height segmentation.

Hernan Badino et al.,“The Stixel World – A Compact Medium Level Representation of the 3D-World”, 2009



(c) Membership cost image

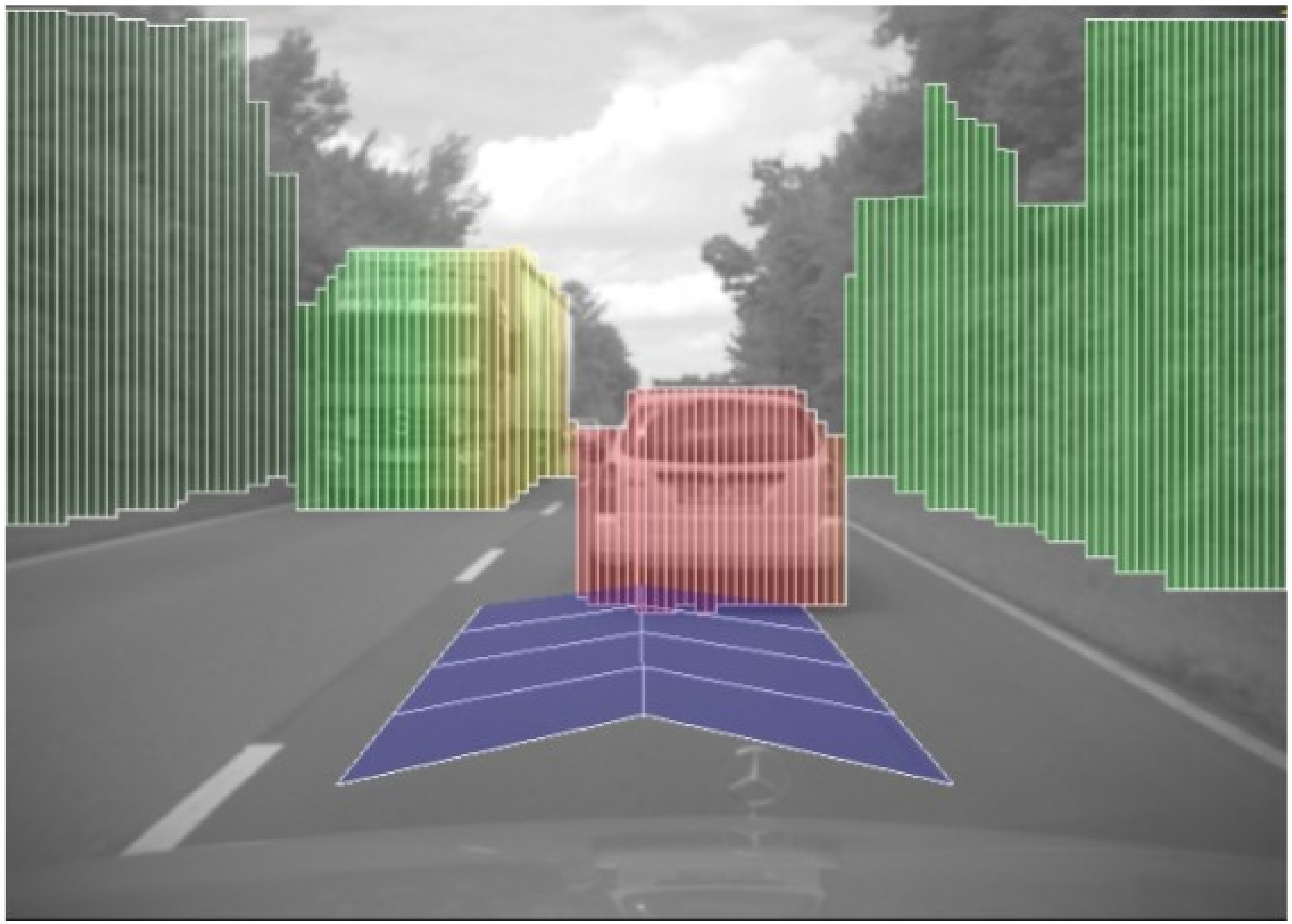


(d) Height segmentation

Final functional: $c_{u,v_0,v_1} = C(u, v_0) + S(u, v_0, v_1) \longrightarrow$ **min**

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

Hernan Badino et al.,“The Stixel World – A Compact Medium Level Representation of the 3D-World”, 2009

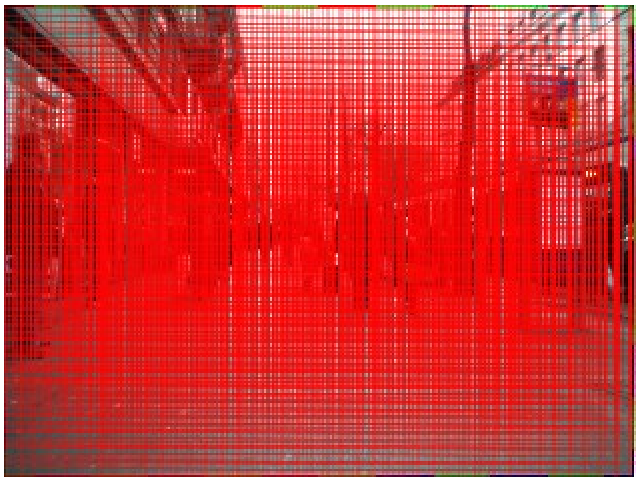


VGA 40fps on 3 GHz

Benenson et al., “Stixels estimation without depth map computation”, 2011

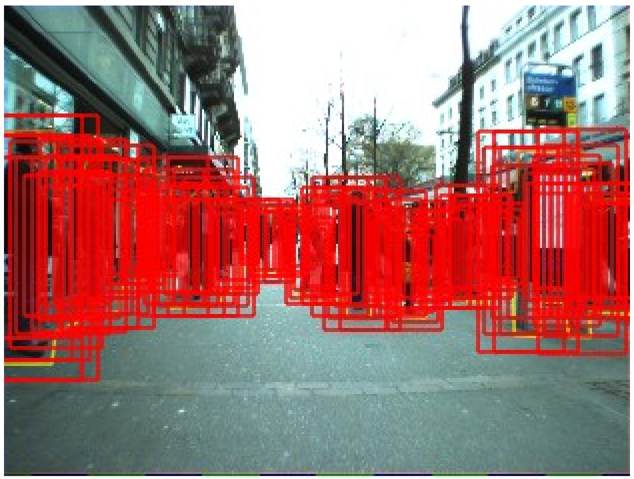
- Matching cost **c(u,v,d)** – SAD of a single pixel (u,v) on the left and (u+d,v) on the right image
- Ground plane estimation by V-disparity method but without computing disparity maps: directly project the costs along the horizontal axis (u-axis) so that each pixel is the summed cost of every pixel along the v-disparity unidimensional slice

Benenson et al., “Stixels estimation without depth map computation”, 2011



75000 windows

2500 windows

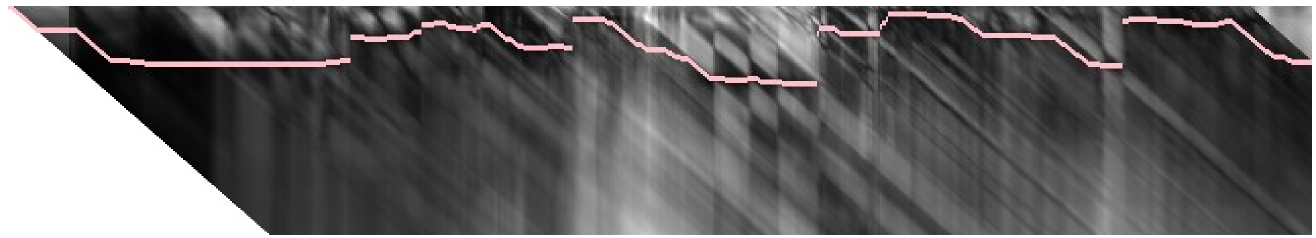


650 windows

- DP for stixels distance estimation
- DP for stixels height estimation
- Open source code in DOPPIA

Benenson et al., “Stixels estimation without depth map computation”, 2011

DP for stixels distance estimation



$$d_s^*(u) = \underset{d(u)}{\operatorname{argmin}} \sum_u c_s\left(u, d(u)\right) + \sum_{u_a, u_b} s_s\left(d(u_a), d(u_b)\right)$$

$$c_s(u,d) = c_o(u,d) + c_g(u,d)$$

$$c_o(u,d) = \sum_{v=v(\check{h}_o,d)}^{v(d)} c_m(u,v,d)$$

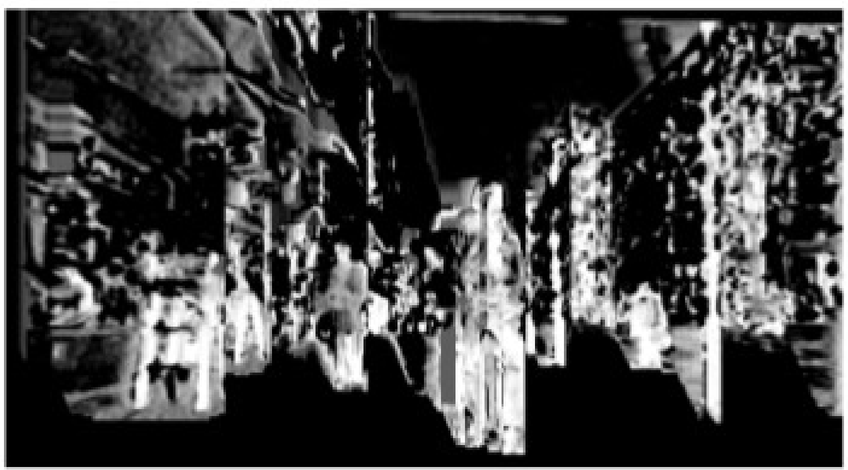
$$c_g(u,d) = \sum_{v=v(d)}^{|V|} c_m(u,v,f_{ground}(v))$$

$$s_s\left(d_a, d_b\right) = \begin{cases} \infty & \text{if} \quad d_a < d_b - 1 \\ c_o\left(u_a, d_a\right) & \text{if} \quad d_a = d_b - 1 \\ 0 & \text{if} \quad d_a > d_b - 1 \end{cases}$$

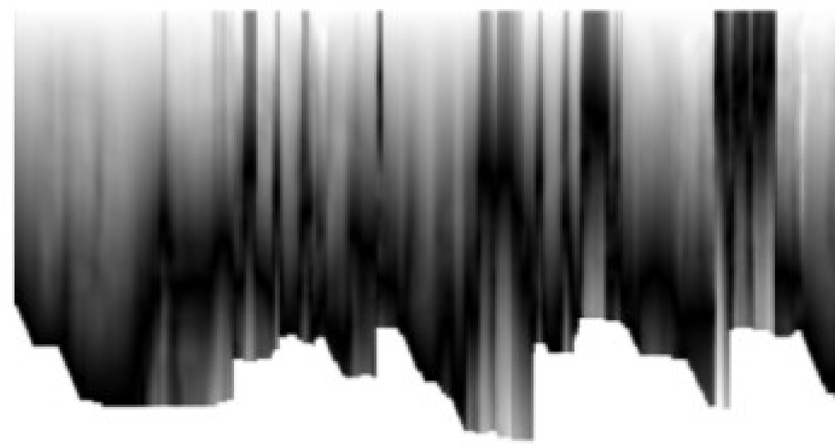
Similar to Kubota et. al, “A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems”, 2007

Benenson et al., “Stixels estimation without depth map computation”, 2011

DP for hight estimation



Membership value



Membership cost image

Similar to the Hernan Badino et al.,“The Stixel World – A Compact Medium Level Representation of the 3D-World”, 2009

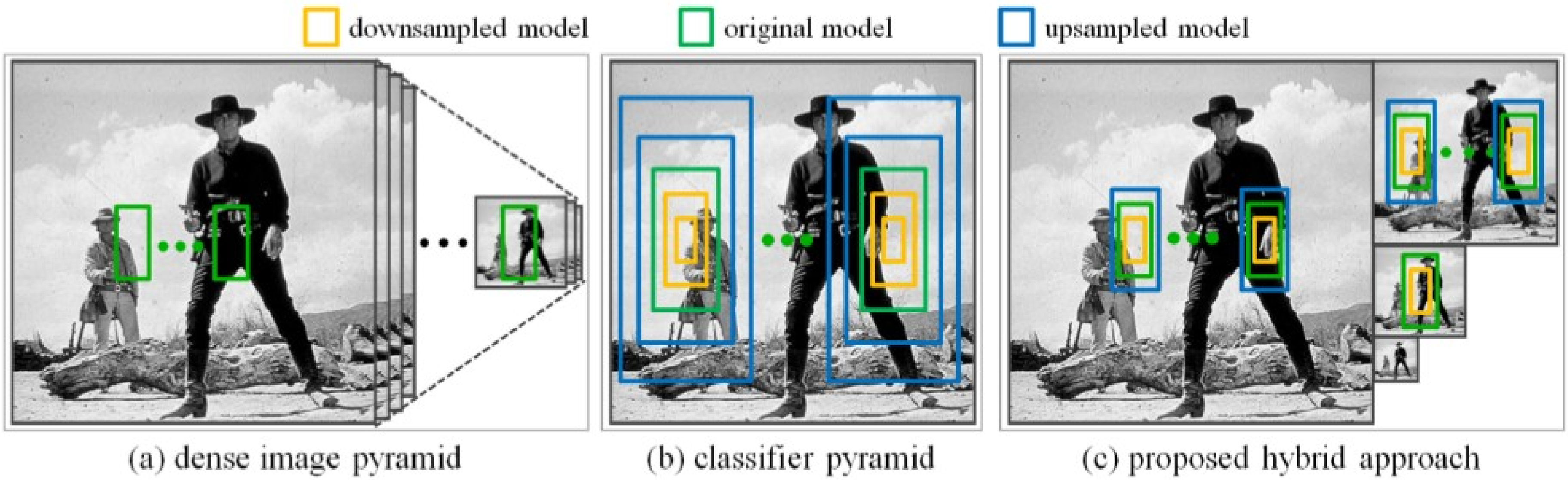
Benenson et al., “Stixels estimation without depth map computation”, 2011

~300Hz for ground plane estimation

~95Hz for ground plane + stixels distance estimation

~25Hz for ground plane + stixels distance + stixels height estimation

Dollár et al., “The Fastest Pedestrian Detector in the West”, 2010



- **<u>Not</u>** stereo
- Hybrid approach:
  – sparse image pyramid (instead of dense)
  – extrapolated features on different scales (instead of feature pyramid)

Dollár et al., “The Fastest Pedestrian Detector in the West”, 2010

- Theorem of Ruderman and Bialek that various statistics of natural images are independent of the scale at which the images were captured
- Exponential scaling law: E[f(I, s+s0) / f(I, s0)] = e^(-lambda*s)
- 0.25–0.5s on VGA images: 10 times faster than the sparse image pyramid approach

Dollár et al., “The Fastest Pedestrian Detector in the West”, 2010

- Theorem of Ruderman and Bialek: various statistics of natural images are independent of the scale at which the images were captured
- Exponential scaling law:

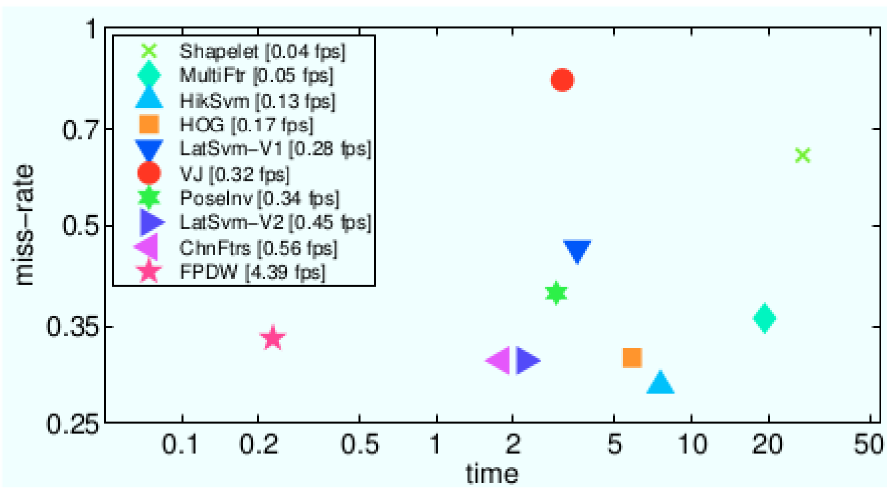
  $$\mathbf{f\ (I,\ s) \approx f\ (I,\ 0)\ e^{(-\lambda s)}}$$

  where

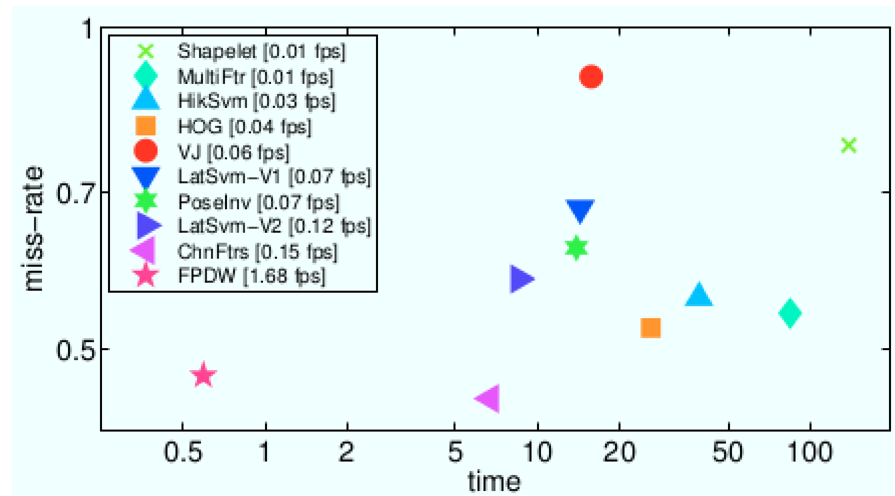
  - f(I, s) is a feature on the image I after downsampling by a factor of 2^s
  - λ is a parameter (about 1.099) estimated empirically on test data

Dollár et al., “The Fastest Pedestrian Detector in the West”, 2010

(a) Caltech Pedestrian Data (peds. ≥ 100 pixels)



(b) Caltech Pedestrian Data (peds. ≥ 50 pixels))



- 5fps on VGA
- Detection accuracy loss ~1–3%
- 0.25–0.5s on VGA images: 10 times faster than the sparse image pyramid approach

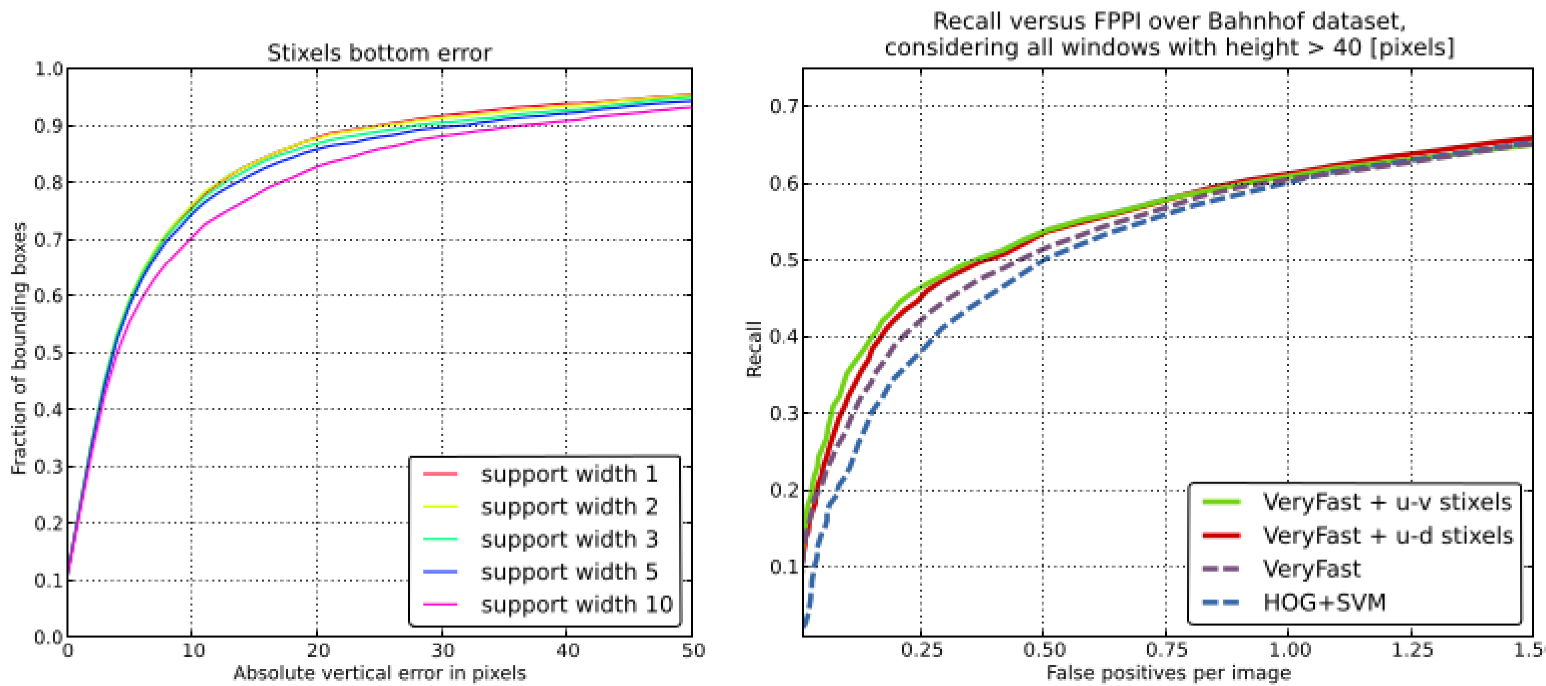
Benenson et al., “Pedestrian detection at 100 frames per second”, 2012

- Ground plane estimate + stixels at about 135 Hz on CPU (without depth map computation)
- Objects detection (GPU)
- Viola and Jones idea “scale the features not the images”
- improve on top of **ChnFtrs** (HOG+SVM) on GPU
- Speed-up by using the FPDW but with N/K classifiers
- Monocular: 50 fps
- 135 Hz from stereo-pair to stixels on CPU
- Speed-up by sparsing the possible stixel positions
- Open source code in DOPPIA

Benenson et al., “Fast stixel computation for fast pedestrian detection”, 2012

- stixel distance estimation problems examined
- wrong quantization: horizon objects are more fine grained than the near objects
- ignores horizontal gradient
- ground plane estimation domain changed from u-disparity to u-v-disparity
- complexity: DP – O(Q*B^2), where Q is the number of stixels and B is the number of row bands; lowering Q from 128 to 50 and and lowering B from 1 pixel stixels to 2 pixel stixels must drop the time by a factor 7
- Open source code in DOPPIA

Benenson et al., “Fast stixel computation for fast pedestrian detection”, 2012



(a) Varying stixel support width (stixel width 3 pixels, 25 row bands) (b) Detection quality of different methods

Figure 5: Stixel and detection quality results.

GPU-bound (165 Hz)