A Dynamic MRF Model for Foreground Detection on Range Data Sequences of Rotating Multi-Beam Lidar

Csaba Benedek, Dömötör Molnár and Tamás Szirányi

Distributed Events Analysis Research Laboratory
Computer and Automation Research Institute (MTA SZTAKI)
Budapest Hungary
contact email: csaba.benedek@sztaki.mta.hu

Workshop on Depth Image Analysis 2012, Tsukuba City,

Content

Introduction

Problem formulation and data mapping

Point cloud classification

Evaluation and applications

MTA SZTAKI

Content

Introduction

Problem formulation and data mapping

Point cloud classification

Evaluation and applications

Introduction

- Foreground detection from a static viewpoint:
  - separating regions moving objects in measurement sequences of a sensor installed in a fixed position
- Applications of foreground detection in visual surveillance
  - people or vehicle detection and tracking
  - activity analysis
  - biometric identification
- Difficulties with optical video sequences

Low illumination

Occlusion (1 or 2 ?)

Various object appearances

Introduction

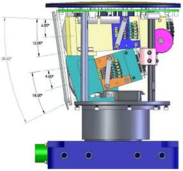
- Range cameras instead of conventional video sources
  - Direct geometric information, independent of outside illumination
  - Avoiding artifacts of stereo vision

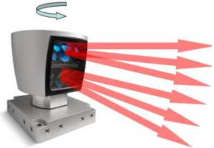
- Time-of-Light (ToF) cameras
  - depth image sequences over a regular 2D pixel lattice
  - established image processing approaches (such as MRFs)
  - limited Field of View (FoV)

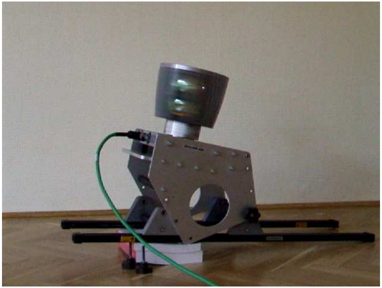
- Rotating multi-beam Lidar systems (RMB-Lidar)
  - 360° FoV of the scene
  - artifacts of rotating sensor: angle shift between time frames, fluctuation of rotation speed

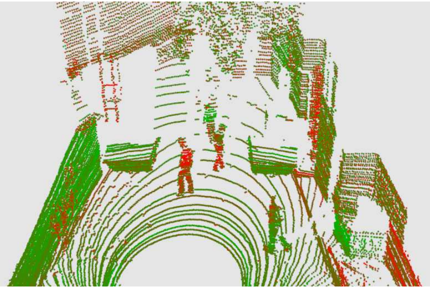
Velodyne HDL-64 High Speed RBM System

- Specification
  - 64 laser and sensor
  - 120m distance
  - $<$ 2cm accuracy
  - $>$ 1.333M point/sec



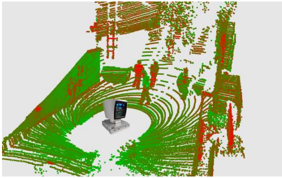


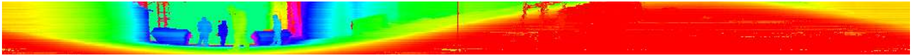




Range image formation of a RMB Lidar

- Point cloud mapping into a cylinder shaped range image
  - cylinder axis: axis of the rotation
  - vertical resolution: number of sensors
  - horizontal resolution: rot. speed dependent

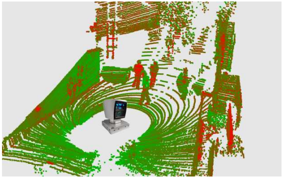


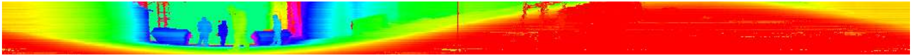


- Problems
  - Ambiguous pixel-surface mapping:
    - different objects at a given pixel in the consecutive time steps
  - Multi-modal distributions for the background-range values
    - aggregated errors in case of dense background motion (e.g. moving vegetation)
  - Non-linear calibration to obtain Euclidean coordinates from the measurements (distance, pitch and angle)
    - inhomogeneous density of the projected points

Range image formation of a RMB Lidar

- Point cloud mapping into a cylinder shaped range image
  - cylinder axis: axis of the rotation
  - vertical resolution: number of sensors
  - horizontal resolution: rot. speed dependent

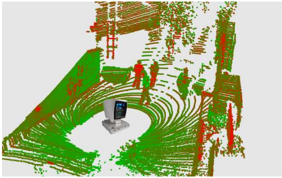


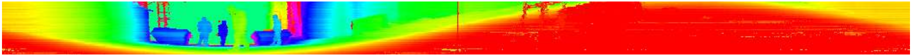


- Problems
  - Ambiguous pixel-surface mapping:
    - different objects at a given pixel in the consecutive time steps
  - Multi-modal distributions for the background-range values
    - aggregated errors in case of dense background motion (e.g. moving vegetation)
  - Non-linear calibration to obtain Euclidean coordinates from the measurements (distance, pitch and angle)
    - inhomogeneous density of the projected points

Range image formation of a RMB Lidar

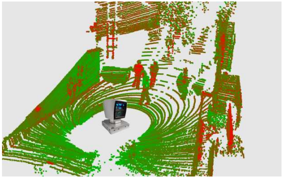
- Point cloud mapping into a cylinder shaped range image
  - cylinder axis: axis of the rotation
  - vertical resolution: number of sensors
  - horizontal resolution: rot. speed dependent



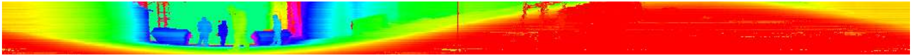


- Problems
  - Ambiguous pixel-surface mapping:
    - different objects at a given pixel in the consecutive time steps
  - Multi-modal distributions for the background-range values
    - aggregated errors in case of dense background motion (e.g. moving vegetation)
  - Non-linear calibration to obtain Euclidean coordinates from the measurements (distance, pitch and angle)
    - inhomogeneous density of the projected points

Range image formation of a RMB Lidar

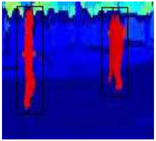


- Point cloud mapping into a cylinder shaped range image
  - cylinder axis: axis of the rotation
  - vertical resolution: number of sensors
  - horizontal resolution: rot. speed dependent



- Problems
  - Ambiguous pixel-surface mapping:
    - different objects at a given pixel in the consecutive time steps
  - Multi-modal distributions for the background-range values
    - aggregated errors in case of dense background motion (e.g. moving vegetation)
  - Non-linear calibration to obtain Euclidean coordinates from the measurements (distance, pitch and angle)
    - inhomogeneous density of the projected points

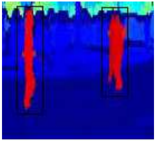
Related work on RBM-Lidar sensors



- Kalyan,2010, IEEE SMC: direct extraction of the foreground objects from the range image by mean-shift segmentation
  - moving and static objects may be merged into the same blob
- Foreground detection in the spatial 3D domain
  - only bounding boxes $\rightarrow$ insufficient for activity recognition (e.g. skeleton fitting)
  - MRF techniques based on 3D spatial point neighborhoods $\rightarrow$ low accuracy for small neighborhoods, high computational complexity for large ones

- Proposed model: a hybrid approach
  - MRF filtering in the 2D range image domain
  - 3D point classification to handle 2D ambiguities
  - spatial foreground model to eliminate background motion

Related work on RBM-Lidar sensors



- Kalyan,2010, IEEE SMC: direct extraction of the foreground objects from the range image by mean-shift segmentation
  - moving and static objects may be merged into the same blob
- Foreground detection in the spatial 3D domain
  - only bounding boxes $\rightarrow$ insufficient for activity recognition (e.g. skeleton fitting)
  - MRF techniques based on 3D spatial point neighborhoods $\rightarrow$ low accuracy for small neighborhoods, high computational complexity for large ones

- Proposed model: a hybrid approach
  - MRF filtering in the 2D range image domain
  - 3D point classification to handle 2D ambiguities
  - spatial foreground model to eliminate background motion

Content

Introduction

Problem formulation and data mapping

Point cloud classification

Evaluation and applications

Problem definition and notations

- Pointcloud at time t: $\mathcal{L}^t = \{p_1^t, \ldots, p_{l^t}^t\}$, $l^t = R \cdot c^t$
  - R number of vertically aligned sensors,
  - c^t: number of point columns at t
- Point attributes for $p \in \mathcal{L}^t$:
  - sensor distance $d(p) \in [0, D_{\max}]$, pitch index $\hat{\vartheta}(p) \in \{1, \ldots, R\}$ and yaw angle $\varphi(p) \in [0, 360°]$
- Point labeling: $\omega(p) \in \{\mathrm{fg}, \mathrm{bg}\}$
- Range image formation:
  - Cylinder projection using a $R \times S_W$ sized 2D pixel lattice S. $s = [y_s, x_s]$: given pixel in S
  - $\mathcal{P} : \mathcal{L}^t \to S$ point mapping operator:

$$s \stackrel{\text{def}}{=} \mathcal{P}(p) \text{ iff } y_s = \hat{\vartheta}(p),\ x_s = \mathrm{round}\left(\varphi(p) \cdot \frac{S_W}{360°}\right)$$

MTA SZTAKI

Problem definition and notations

- Pointcloud at time t: $\mathcal{L}^t = \{p_1^t, \ldots, p_{l^t}^t\}$, $l^t = R \cdot c^t$
  - R number of vertically aligned sensors,
  - c^t: number of point columns at t
- Point attributes for $p \in \mathcal{L}^t$:
  - sensor distance $d(p) \in [0, D_{\max}]$, pitch index $\hat{\vartheta}(p) \in \{1, \ldots, R\}$ and yaw angle $\varphi(p) \in [0, 360°]$
- Point labeling: $\omega(p) \in \{\mathrm{fg}, \mathrm{bg}\}$
- Range image formation:
  - Cylinder projection using a $R \times S_W$ sized 2D pixel lattice S. $s = [y_s, x_s]$: given pixel in S
  - $\mathcal{P} : \mathcal{L}^t \to S$ point mapping operator:

$$s \stackrel{\text{def}}{=} \mathcal{P}(p) \text{ iff } y_s = \hat{\vartheta}(p),\ x_s = \mathrm{round}\left(\varphi(p) \cdot \frac{S_W}{360°}\right)$$

Problem definition and notations

- Pointcloud at time t: $\mathcal{L}^t = \{p^t_1, \ldots, p^t_{l^t}\}$, $l^t = R \cdot c^t$
  - R number of vertically aligned sensors,
  - c^t: number of point columns at t
- Point attributes for $p \in \mathcal{L}^t$:
  - sensor distance $d(p) \in [0, D_{\max}]$, pitch index $\hat{\vartheta}(p) \in \{1, \ldots, R\}$ and yaw angle $\varphi(p) \in [0, 360°]$
- Point labeling: $\omega(p) \in \{\mathrm{fg}, \mathrm{bg}\}$
- Range image formation:
  - Cylinder projection using a $R \times S_W$ sized 2D pixel lattice S. $s = [y_s, x_s]$: given pixel in S
  - $\mathcal{P} : \mathcal{L}^t \to S$ point mapping operator:

$$s \stackrel{\text{def}}{=} \mathcal{P}(p) \text{ iff } y_s = \hat{\vartheta}(p),\ x_s = \mathrm{round}\left(\varphi(p) \cdot \frac{S_W}{360°}\right)$$

Problem definition and notations

- Pointcloud at time t: $\mathcal{L}^t = \{p_1^t, \ldots, p_{l^t}^t\}$, $l^t = R \cdot c^t$
  - R number of vertically aligned sensors,
  - c^t: number of point columns at t
- Point attributes for $p \in \mathcal{L}^t$:
  - sensor distance $d(p) \in [0, D_{\max}]$, pitch index $\hat{\vartheta}(p) \in \{1, \ldots, R\}$ and yaw angle $\varphi(p) \in [0, 360°]$
- Point labeling: $\omega(p) \in \{\mathrm{fg}, \mathrm{bg}\}$
- Range image formation:
  - Cylinder projection using a $R \times S_W$ sized 2D pixel lattice S. $s = [y_s, x_s]$: given pixel in S
  - $\mathcal{P} : \mathcal{L}^t \to S$ point mapping operator:

$$s \stackrel{\text{def}}{=} \mathcal{P}(p) \text{ iff } y_s = \hat{\vartheta}(p),\ x_s = \mathrm{round}\left(\varphi(p) \cdot \frac{S_W}{360°}\right)$$

Content

Introduction

Problem formulation and data mapping

Point cloud classification

Evaluation and applications

Background model

- $\forall s \in S$: Mixture of Gaussians approximation of the $d(s)$ range history
  - fixed K number of components (here $K = 5$)
  - background: k_s largest weighted components $\sum_{i=1}^{k_s} w_s^i > T_{\mathrm{bg}}$
- $f_{\mathrm{bg}}(s)$: background fitness term of pixel s

$$f_{\mathrm{bg}}(s) = \sum_{i=1}^{k_s} w_s^i \cdot \eta\left(d(s), \mu_s^i, \sigma_s^i\right).$$

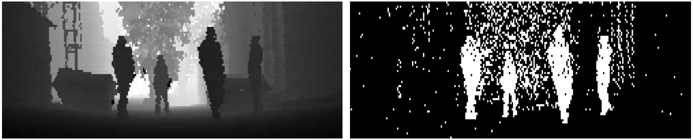
- Noisy result - errors in textured or dynamic background

Background model

- $\forall s \in S$: Mixture of Gaussians approximation of the $d(s)$ range history
  - fixed K number of components (here $K = 5$)
  - background: k_s largest weighted components $\sum_{i=1}^{k_s} w_s^i > T_{\mathrm{bg}}$
- $f_{\mathrm{bg}}(s)$: background fitness term of pixel s

$$f_{\mathrm{bg}}(s) = \sum_{i=1}^{k_s} w_s^i \cdot \eta\left(d(s), \mu_s^i, \sigma_s^i\right).$$

- Noisy result - errors in textured or dynamic background



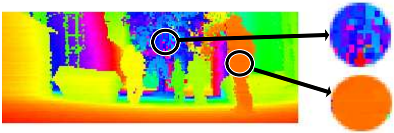
Background model

- $\forall s \in S$: Mixture of Gaussians approximation of the $d(s)$ range history
  - fixed K number of components (here $K = 5$)
  - background: k_s largest weighted components $\sum_{i=1}^{k_s} w_s^i > T_{\text{bg}}$
- $f_{\text{bg}}(s)$: background fitness term of pixel s

$$f_{\text{bg}}(s) = \sum_{i=1}^{k_s} w_s^i \cdot \eta\left(d(s), \mu_s^i, \sigma_s^i\right).$$

- Noisy result - errors in textured or dynamic background

Foreground model



Local range values in *motion-regions*

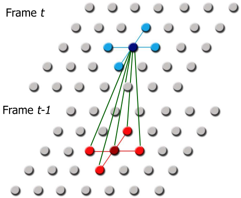
- Foreground class: non-parametric kernel density model
  - in the neighborhood of foreground pixels, we should find foreground pixels with similar range values

$$f_{\mathrm{fg}}(s) = \sum_{r \in N_s} \left(1 - \zeta(f_{\mathrm{bg}}(r), \tau_{\mathrm{fg}}, m_\star)\right) \cdot k\left(\frac{d_s^t - d_r^t}{h}\right)$$

  - h: kernel bandwidth, $\zeta : \mathbb{R} \to [0,1]$ sigmoid function

Dynamic MRF Model

- 2-D pixel lattice $\rightarrow$ graph: $S = \{s\}$
- Nodes: image points (s is a pixel)
- Edges: interactions $\rightarrow$ cliques
  - intra-frame edges: spatial smoothness
  - inter-frame edges: temporal smoothness



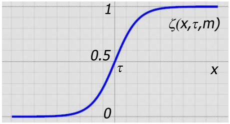
- MRF energy function

$$E = \sum_{s \in S} \underbrace{V_D(d_s^t | \omega_s^t)}_{\text{Dataterm}} + \sum_{s \in S} \underbrace{\sum_{r \in N_s} \alpha \cdot \mathbf{1}\{\omega_s^t \neq \omega_r^{t-1}\}}_{\text{temporal smoothness}} + \sum_{s \in S} \underbrace{\sum_{r \in N_s} \beta \cdot \mathbf{1}\{\omega_s^t \neq \omega_r^t\}}_{\text{spatial smoothness}},$$

- Energy optimization
  - Graph cut based method (real time)

Dynamic MRF Model: data terms

- $\zeta(x, \tau, m)$ sigmoid function: *soft thresholding*
  - τ: soft threshold, m: steepness



- The data terms are derived from the data energies by sigmoid mapping:

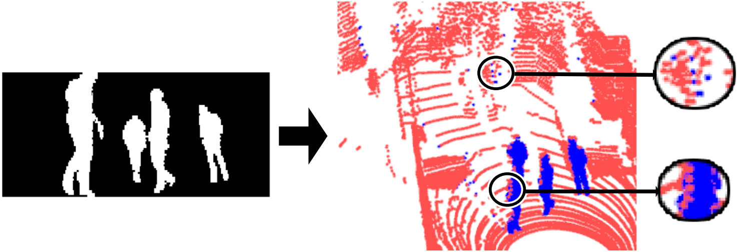
$$V_D(d_s^t | \omega_s^t = \mathrm{bg}) = \zeta(-\log(f_{\mathrm{bg}}^t(s)), \tau_{\mathrm{bg}}, m_{\mathrm{bg}})$$

$$V_D(d_s^t | \omega_s^t = \mathrm{fg}) = \begin{cases} 1 & \text{if } d_s^t > \max_{\{i=1\ldots k_s\}} \mu_s^{i,t} + \epsilon \\ \zeta(-\log(f_{\mathrm{fg}}^t(s)), \tau_{\mathrm{fg}}, m_{\mathrm{fg}}) & \text{otherwise.} \end{cases}$$

  - Setting sigmoid parameters $\tau_{\mathrm{fg}}, \tau_{\mathrm{bg}}, m_{\mathrm{fg}}, m_{\mathrm{bg}}$: Maximum Likelihood learning, based on training samples

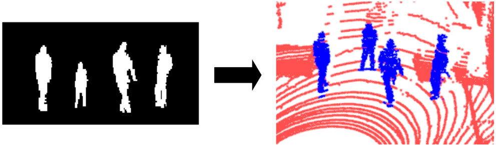
Label backprojection

- Point cloud labeling based on the segmented range image
  - Problems due to angle quantization for the discrete pixel lattice
  - Misclassified points near *object* edges and,'shadow' edges



Final point cloud classification

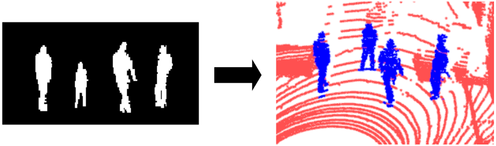
- Classification of the point of the cloud based on the segmented range image
  - $\omega(p)$: point cloud label
  - ω_S: range image label of pixel corresponding to point p
  - *handling* the ambiguous point (p) - pixel (s) assignments



- $\omega(p) = \mathrm{fg}$, iff one of the following two conditions holds:
  - $\omega_S = \mathrm{fg}$ and distance of p *matches* to the background range image value in s
  - $\omega_S = \mathrm{bg}$ and we find a neighbor r of pixel s, where $\omega_r = \mathrm{fg}$ and the distance of p *matches* to the background range image value in r
- $\omega(p) = \mathrm{bg}$: otherwise.

Final point cloud classification

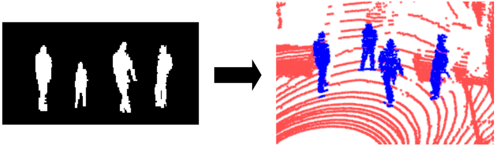
- Classification of the point of the cloud based on the segmented range image
  - $\omega(p)$: point cloud label
  - ω_s: range image label of pixel corresponding to point p
  - *handling* the ambiguous point (p) - pixel (s) assignments



- $\omega(p) = \mathrm{fg}$, iff one of the following two conditions holds:
  - $\omega_s = \mathrm{fg}$ and distance of p *matches* to the background range image value in s
  - $\omega_s = \mathrm{bg}$ and we find a neighbor r of pixel s, where $\omega_r = \mathrm{fg}$ and the distance of p *matches* to the background range image value in r
- $\omega(p) = \mathrm{bg}$: otherwise.

Final point cloud classification

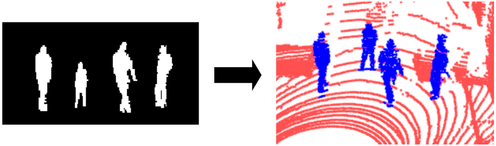
- Classification of the point of the cloud based on the segmented range image
  - $\omega(p)$: point cloud label
  - ω_s: range image label of pixel corresponding to point p
  - *handling* the ambiguous point (p) - pixel (s) assignments



- $\omega(p) = \mathrm{fg}$, iff one of the following two conditions holds:
  - $\omega_s = \mathrm{fg}$ and distance of p *matches* to the background range image value in s
  - $\omega_s = \mathrm{bg}$ and we find a neighbor r of pixel s, where $\omega_r = \mathrm{fg}$ and the distance of p *matches* to the background range image value in r
- $\omega(p) = \mathrm{bg}$: otherwise.

Final point cloud classification

- Classification of the point of the cloud based on the segmented range image
  - $\omega(p)$: point cloud label
  - ω_s: range image label of pixel corresponding to point p
  - *handling* the ambiguous point (p) - pixel (s) assignments



- $\omega(p) = \mathrm{fg}$, iff one of the following two conditions holds:
  - $\omega_s = \mathrm{fg}$ and distance of p *matches* to the background range image value in s
  - $\omega_s = \mathrm{bg}$ and we find a neighbor r of pixel s, where $\omega_r = \mathrm{fg}$ and the distance of p *matches* to the background range image value in r
- $\omega(p) = \mathrm{bg}$: otherwise.

Content

Introduction

Problem formulation and data mapping

Point cloud classification

Evaluation and applications

Test datasets

- Two LIDAR sequences: *Courtyard* (video surveillance) and *Traffic* (traffic monitoring)
  - Sensor: Velodyne HDL 64E S2 camera, $R = 64$ beams
  - *Courtyard*: 2500 frames, four pedestrians, 20 Hz recording
  - *Traffic*: 160 frames, >20 objects (cars), 5 Hz recording
- Reference techniques:
  - *Basic MoG* on the range image
  - *uniMRF*: uniform foreground model for range image segmentation in the DMRF framework.
  - *3D-MRF* MRF model in the 3D point cloud space
- Quantitative analysis:
  - 3D point cloud annotation tool - manual Ground Truth (GT) generation
  - Point level F-measure of foreground detection

Test datasets

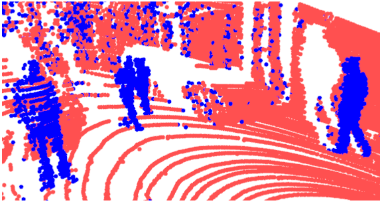
- Two LIDAR sequences: *Courtyard* (video surveillance) and *Traffic* (traffic monitoring)
  - Sensor: Velodyne HDL 64E S2 camera, $R = 64$ beams
  - *Courtyard*: 2500 frames, four pedestrians, 20 Hz recording
  - *Traffic*: 160 frames, >20 objects (cars), 5 Hz recording
- Reference techniques:
  - *Basic MoG* on the range image
  - *uniMRF*: uniform foreground model for range image segmentation in the DMRF framework.
  - *3D-MRF* MRF model in the 3D point cloud space
- Quantitative analysis:
  - 3D point cloud annotation tool - manual Ground Truth (GT) generation
  - Point level F-measure of foreground detection

Test datasets

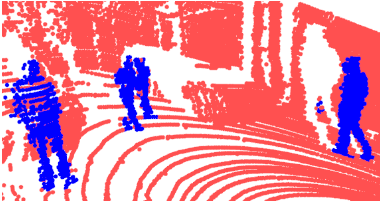
- Two LIDAR sequences: *Courtyard* (video surveillance) and *Traffic* (traffic monitoring)
  - Sensor: Velodyne HDL 64E S2 camera, $R = 64$ beams
  - *Courtyard*: 2500 frames, four pedestrians, 20 Hz recording
  - *Traffic*: 160 frames, >20 objects (cars), 5 Hz recording
- Reference techniques:
  - *Basic MoG* on the range image
  - *uniMRF*: uniform foreground model for range image segmentation in the DMRF framework.
  - *3D-MRF* MRF model in the 3D point cloud space
- Quantitative analysis:
  - 3D point cloud annotation tool - manual Ground Truth (GT) generation
  - Point level F-measure of foreground detection

Qualitative results

Courtyard scenario

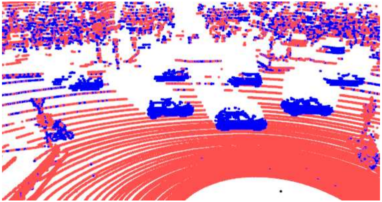


Basic MoG

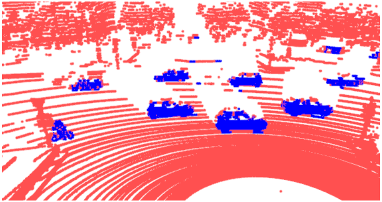


Proposed DMRF

Traffic scenario

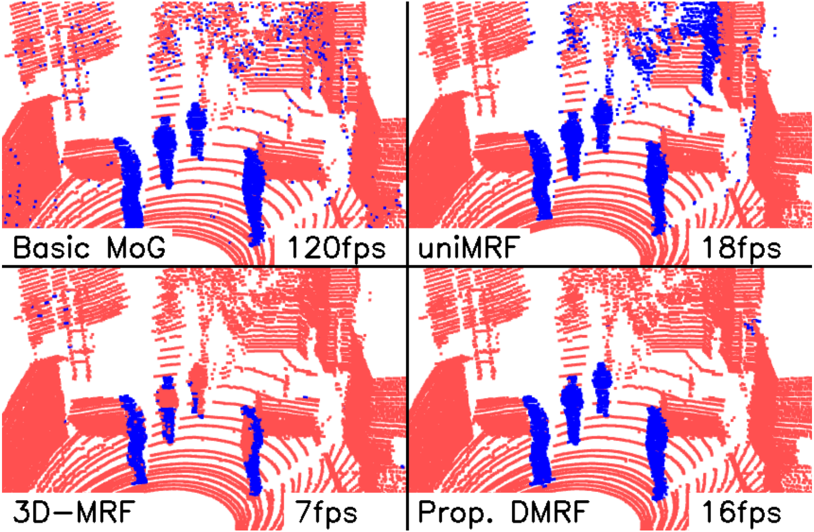


Basic MoG



Proposed DMRF

Qualitative results

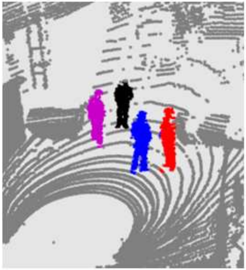
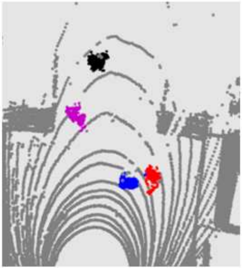
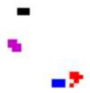


Quantitative evaluation

	Sequence	Prop.	MoG	uniMRF	3D-MRF	**DMRF**
Det. rate	*Courtyard*	4 obj/fr.	55.7	81.0	88.1	95.1
	Traffic	20 obj/fr.	70.4	68.3	76.2	74.0
Speed (fps)	*Courtyard*	65Kpt/fr	120 fps	18 fps	7 fps	16 fps
	Traffic	260Kpt/fr	120 fps	18 fps	2 fps	16 fps

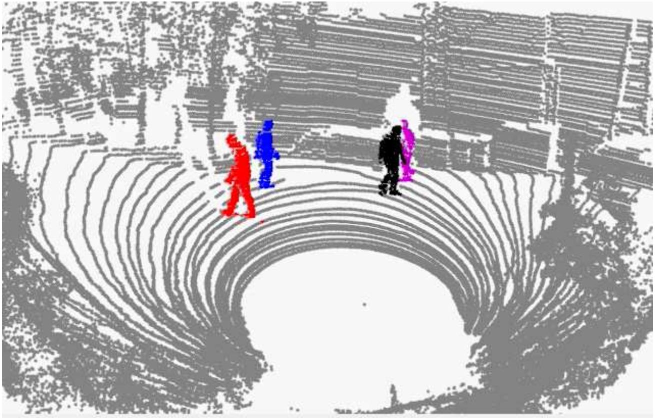
Application: multiple pedestrian detection & tracking

- Object detection: ground projection of foreground points + blob detection
- Tracking: based on Kalman filter and Hungarian matching algorithm

Application: multiple pedestrian detection & tracking

Online demo available at our laboratory



Application: towards dynamic scene reconstruction