Multi-sensorial Environment Perception in Urban Environment

ROBOVIS 2020 – Keynote talk

Csaba Benedek
*Machine Perception Research Laboratory, Institute for Computer Science and Control (SZTAKI), Hungary*

benedek.csaba@sztaki.hu
Research background

- Institute for Computer Science and Control (SZTAKI),
  - most prestigious IT research institute of Hungary, which bridges the gap between academic research and industrial expectations.

- SZTAKI Machine Perception Research Laboratory (MPLab), founded in 2006
  - Goal: interpretation and organization of information coming from distributed multimodal sensors

- Research Group on Geo-information Computing (GeoComp) @ MPLab headed by Csaba Benedek
  - Research objectives: large scale spatial data management and filtering, automated 3D scene understanding used in various applications
  - Applied methodologies: Computer vision, geometry based and probabilistic modeling, machine learning applications
**LIDAR technology**

**LIDAR = Light Detection and Ranging**

- surveying method that measures distance to a target by illuminating it with laser light and measuring the reflected light with a sensor (≈ laser radar)
- rapidly provides large amount of accurate range data from 100+ meter distances

**LIDAR platforms:**

- Aerial LIDAR
- Ground based mobile mapping system
- Terrestrial static laser scanner
- Real time mobile LIDAR sensors
Instant environment perception from a mobile platform with a new generation geospatial database background

Project of the Hungarian National Research, Development and Innovation Fund (2016-2021)

- Car-mounted multisensory platform
- High resolution localization map
- Accurate environment perception and modeling in smart cities based on multiple sensors

Challenges:
- Different sensor types and different dimension of the data
- Synchronization issues
- Accurate Lidar - Lidar and Lidar - Camera calibration is essential
Environment modeling: static and mobile laser scanning data

Output: colored point cloud
Pros and issues

i3D data

- Quite sparse, less accurate, only coarse geo-coordinates, strongly inhomogeneous point clouds (typical ring patterns) ❌
- Instant data: situation analysis is possible, moving street objects can be observed and analyzed ✓
- Online processing required ❌
- Region classification, object separation and recognition is very challenging due to the particular point cloud characteristics ❌
- Compact point cloud frames (max 60-100k points) – easy management and data transfer ✓

MLS data

- Very dense, very accurate, geo-refered, quite homogeneous point clouds (linear density characteristics wrt. sensor dist), ✓
- Only static environment model ❌
- Offline processing allowed ✓
- Ghosts of moving objects ❌
- Parking cars, standing pedestrians, temporary street furniture, vegetation change ❌
- Huge data: computational issues, memory issues, data transfer issues ❌
Phantom object effects by mobile laser scanning

B. Nagy, and Cs. Benedek: "3D CNN Based Phantom Object Removing from Mobile Laser Scanning Data", International Joint Conference on Neural Networks (IJCNN), pp. 4429-4435, Anchorage, Alaska, USA, 14-19 May, 2017

RIEGL VMX-450
Phantom object effects by mobile laser scanning

• 3D point cloud representation: detected phantoms can be removed
Phantom region detection

• Challenges of phantom region detection
  – Usually sparser than static objects, but...
  – MLS clouds have considerably inhomogeneous point density due
    – occlusions
    – different distances of the surface points from the scanner’s trajectory
    – varying speed of the scanner platform (e.g. accelerating and breaking)
    – different laser adsorption/reflection properties of the different surface materials
  – Highly diverse phantom regions
    – varying speed of moving objects

• Proposed solution: deep neural networks – 3D semantic segmentation task
  – Apart from phantom detection let us distinguish different classes
Goal: semantic point cloud segmentation with 9 classes

- pedestrian
- phantom
- parking vehicle
- tram/bus
- column/tree trunk
- vegetation
- street furniture
- road
- building facade
Existing point cloud classification solutions

• *Global* approaches
  – information from the complete 3D scene for classification of the individual voxels - main challenge: time and memory requirements
  • OctNet (expensive training data annotation)

• Sliding volume based techniques
  – move a 3D box over the scene, using locally available information for the classification of each point cloud segment.
    – Vote3Deep (fixed object size)
    – OG-CNN (purely local features)
    – multi-view technique (projection + 2D CNN models)
    – Latest: PointNet++ (indoors), SGPN (limited scene size), SPLATNet3D (discrete grid)

• Proposed solution: sliding volume based technique, but using both local and global features in data representation
Data representation

• Sparse voxel structure for the input point cloud, with a fine resolution (used 0.1m voxel side length)

• Two feature channels:
  – point *density*: number of included points,
  – *mean elevation*, average of the point height values

• Training volume:
  – \( K \times K \times K \) voxel neighborhood (used \( K = 23 \)),
  – unit of training and recognition in our network
  – Central voxel is classified with considering the whole training volume
Training data representation and dataset generation

Vehicle
Phantom
Wall
Vegetation
Pedestrian
3D CNN

3D CNN architecture
Benchmarking issues

• Existing Mobile laser scanning (MLS) point cloud datasets - relatively small annotated segments
  – Oakland (1.6M points),
  – Paris-rue-Madame (20M points)
  – IQmulus & TerraMobilita (12M labeled points)

• Proposed dataset: **SZTAKI-CityMLS**
  – 327 Million annotated points
  – various urban scenes, including main roads with both heavy and solid traffic, public squares, parks, and sidewalk regions, various types of cars, trams and buses,
  – several pedestrians and diverse vegetation.

URL: [http://mplab.sztaki.hu/geocomp/SZTAKI-CityMLS-DB.html](http://mplab.sztaki.hu/geocomp/SZTAKI-CityMLS-DB.html)
Data characteristic comparison

Paris-rue-Madame  TerraMobilita  SZTAKI-CityMLS
Semantic segmentation of MLS point clouds

- **Offline** utilization: mapping, road state estimation

Evaluation

(a) Original point cloud (Riegl VMX-450)
(b) Ground Truth annotation
(c) OG-CNN labeling
(d) Multi-view projection based labeling
(e) PointNet++ labeling
(f) Proposed C²-CNN labeling

Legend:
- phantom
- tram/bus
- pedestrian
- car
- vegetation
- column
- st. furn.
- ground
- facade
Comparison to reference methods
## Comparison to reference methods

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pr</td>
<td>Rc</td>
<td>F-r</td>
<td>Pr</td>
<td>Rc</td>
<td>F-r</td>
</tr>
<tr>
<td>Phantom</td>
<td>85.3</td>
<td>34.7</td>
<td>49.3</td>
<td>76.5</td>
<td>45.3</td>
<td>56.9</td>
</tr>
<tr>
<td>Pedestrian</td>
<td>61.2</td>
<td>82.4</td>
<td>70.2</td>
<td>57.2</td>
<td>66.8</td>
<td>61.6</td>
</tr>
<tr>
<td>Car</td>
<td>56.4</td>
<td>89.5</td>
<td>69.2</td>
<td>60.2</td>
<td>73.3</td>
<td>66.1</td>
</tr>
<tr>
<td>Vegetation</td>
<td>72.4</td>
<td>83.4</td>
<td>77.5</td>
<td>71.7</td>
<td>78.4</td>
<td>74.9</td>
</tr>
<tr>
<td>Column</td>
<td>88.6</td>
<td>74.3</td>
<td>80.8</td>
<td>83.4</td>
<td>76.8</td>
<td>80.0</td>
</tr>
<tr>
<td>Tram/Bus</td>
<td>91.4</td>
<td>81.6</td>
<td>86.2</td>
<td>85.7</td>
<td>83.2</td>
<td>84.4</td>
</tr>
<tr>
<td>Furniture</td>
<td>72.1</td>
<td>82.4</td>
<td>76.9</td>
<td>57.2</td>
<td>89.3</td>
<td>69.7</td>
</tr>
<tr>
<td>Overall</td>
<td>76.9</td>
<td>74.2</td>
<td>75.5</td>
<td>72.5</td>
<td>73.4</td>
<td>72.9</td>
</tr>
</tbody>
</table>

Note: Voxel level Precision (Pr), Recall (Rc) and F-rates (F-r) are given in percent (overall values weighted with class significance)
Cleaning of terrestrial laser scanning data
Lidar-camera platform

Rotating multi-beam Lidar data acquisition
Ground-obstacle separation for i3D data flow

- 2D grid based locally adaptive terrain modeling
  - expect inhomogeneous RMB Lidar point clouds with typically non-planar ground.

Point cloud classification:
- **Road**
- **Wall/lamp post**
- **Street objects**

Object separation by a two-level grid model

- Object detection

- Separation of close objects with a two-layer hierarchical grid model
Object detection and tracking with Lidar sensors

A. Börcs, B. Nagy and Cs. Benedek: "Fast 3-D Urban Object Detection on Streaming Point Clouds", *Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving at ECCV, Lecture Notes in Computer Science*, Zurich, Switzerland, September 6-12 2014
Working with low-resolution Lidars

Point cloud classification

Object tracking in point clouds

2 x Velodyne VLP-16

A. Börcs, B. Nagy and Cs. Benedek: "Fast 3-D Urban Object Detection on Streaming Point Clouds", Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving at ECCV, Lecture Notes in Computer Science, Zurich, Switzerland, September 6-12 2014
Object classification with Lidar sensors

Lidar based object detection

Sources of errors:
• Similar appearance of point clouds corresponding to different objects
• Merged object blobs

Proposed solution:
• Map based correction

MLS map based vehicle localization and change detection

City Management and Maintenance Map by Mobile Laser Scanning (MLS)

Intelligent Vehicles (IV): Instant 3D (i3D) data
Change detection workflow

1. Ground/obstacles separation
2. Crossmodal point cloud registration
3. Crossmodal change detection

i3D data (RMB Lidar)

MLS data
Result of ground-obstacle separation for sensors
Fővám tér, Budapest

Velodyne HDL64E (i3D)

Velodyne VLP16 (i3D)

Riegl VMX450 (MLS)

Step 1: Ground-obstacle separation
MLS point cloud based semantic map generation

- Point cloud acquired by a Mobile Laser Scanning (MLS) systems
- Handle large point sets (ca. 50 million points)
- Offline data processing
  - Semantic segmentation
  - Filtering
- Reduced point sets (ca. 1 million points)
MLS point cloud based semantic map generation

- Point cloud acquired by a Mobile Laser Scanning (MLS) systems
- Handle large point sets (ca. 50 million points)
- Offline data processing
  - Semantic segmentation
  - Filtering
- Reduced point sets (ca. 1 million points)
MLS point cloud based semantic map generation

- Point cloud acquired by a Mobile Laser Scanning (MLS) systems
- Handle large point sets (ca. 50 million points)
- Offline data processing
  - Semantic segmentation
  - Filtering
- Reduced point sets (ca. 1 million points)
MLS based background point cloud map
After obstacle-ground separation and moving object removal

Step 1: Ground-obstacle separation
**Step 2: Proposed point cloud registration workflow**

**Input:** obstacle clouds after ground removal

- **2.1: Landmark object extraction**
  - Advanced Euclidean clustering

- **2.2: Object based coarse alignment**
  - Hough transform based matching of the extracted object centers

- **2.3: Point level registration refinement**
  - Classical ICP or NDT variant

**Output:** registered point clouds

---

Step 2.1: Landmark object extraction

- RMB Lidar point cloud:
  - two layer hierarchical grid model for object separation (see earlier)
  - detected moving objects may be removed (accuracy is not critical)

- MLS point cloud
  - Starting from semantic segmentation map (keeping columns, street furniture elements)
  - Euclidean clustering

Step 2: Crossmodal point cloud registration
Step 2.2: Object based coarse alignment

- Motivation: fingerprint minutiae matching
  - Generalized Hough transform to estimate the optimal match between the two sets of object centers extracted from the input clouds

```plaintext
procedure Alignment(F1, F2, T)
    C1 <- ObjectDetect(F1)
    C2 <- ObjectDetect(F2)
    Initialize 4D accumulator array A
    for all o1 in C1 do
        for all o2 in C2 do
            for a in [0, 359] do
                o1' <- Rot(a) * o1
                (dx, dy, dz) <- o2 - o1'
                A[dx,dy,dz,a] <- A[dx,dy,dz,a] + 1
            end for
        end for
    end for
    a, dx, dy, dz <- FindMax(A)
    F1, T <- TransformCloud(F1, a, dx, dy, dz)
end procedure
```
Coarse alignment result

- Algorithm handles large initial rotation error
- Point clouds to be matched may have significantly different resolution and density characteristics
- Point clouds may be recorded at different times: several non-consistent object samples
Step 2.3: Point level registration refinement

- Normal Distributions Transform (NDT) or Iterative Closest Point (ICP) applied on the obstacle cloud
Registration results for high definition inputs

Initial

Registered
Registration results between VLP16 and HDL64 inputs

- Algorithm can be also used to auto-calibrate different sensors on the same platform
- For even sparser point clouds (e.g. 8 or 4 beam Lidars) support of IMU is recommended
Point cloud registration results
Deák tér, Budapest

Step 2: Crossmodal point cloud registration
**Step 3: Crossmodal change detection**

i3D vs. MLS point cloud matching

Change detection workflow:

**Input:** registered i3D and MLS obstacle clouds

1. **3.1: Range image creation**
2. **3.2: MRF based change detection in the range image domain**
3. **3.3: Label backprojection to point cloud**

**Output:** separation of dynamic and static regions of the i3D obstacle cloud
Step 3.1 Range image formation

• Velodyne range image:
  – Projection on a cylinder around the main axis of the RMB Lidar sensor
  – Interpolation of missing data of the range image due to
    • inhomogeneous point clouds
    • quantization errors of the rotation angle

• MLS range image:
  – Generated from the 3D MLS point cloud with ray tracing
    • available global position and orientation of the RMB Lidar after registration
    • simulated rays emitted into the MLS cloud from the moving platform’s center position
      with the same vertical and horizontal resolution as the RMB Lidar scanner
3.1 Range image formation

Raw Velodyne (i3D) range image

Filtered & interpolated Velodyne (i3D) range image

MLS range image
3.2 Change detection in the range image domain

- Filtered & interpolated Velodyne (i3D) range image
- MLS based range image from the Velodyne’s position
- MRF based change mask in the range image domain

- Markov Random Field (MRF)
- Energy minimization (potts model)

\[ E = \sum_{s \in S} V_D(\delta_{i3D}^s | l_s) + \sum_{s \in S} \sum_{r \in N_s} \beta \cdot 1\{l_s \neq l_r\} \]

**Data term:** consistent range values in the input images

**Prior term:** smooth change mask
3.2 Label backprojection

Filtered & interpolated Velodyne (i3D) range image

MRF based change mask in the range image domain

Back projection of the 2D change mask to the 3D Velodyne (i3D) point cloud
Change detection results in a tram stop

Reference background MLS point cloud

Classified instant 3D Lidar point cloud

Dynamic  Static  Ground
Comparison to a voxel based technique in a sidewalk area

Voxel based (VOX)  Proposed MRF-range image based
Velodyne - MLS data registration
Synthetized view for geo-referred moving object detection

Low level change map for correcting object detection errors

So far: object detection based on a **single onboard sensor**
Why camera-Lidar fusion is necessary?

- Low resolution: close objects (pedestrians) may be merged into the same blob

- *false pedestrian detections due to mirroring effects*
Target-based semi-manual reference method

Automated Lidar-camera calibration
Working without calibration object

Structure from Motion (SfM)

SFM point cloud

Camera images

Structure from Motion (SfM)

Reprojection calculation during the SfM method
Proposed method

SfM point cloud

$T_1(t_n)$

$t_n$

timestamp of the image

Lidar point cloud

$T_R = T_2 T_3$

$t_1$

$t_2$

$t_3$

$\cdots$
Object based registration

procedure Alignment(F1, F2, T)
C1 <- ObjectDetect(F1)
C2 <- ObjectDetect(F2)
Initialize 4D accumulator array A
for all o1 in C1 do
    for all o2 in C2 do
        for a in [0, 359] do
            o1' <- Rot(a) * o1
            (dx, dy, dz) <- o2 - o1'
            A[dx,dy,dz,a] <- A[dx,dy,dz,a] + 1
        end for
    end for
end for
a, dx, dy, dz <- FindMax(A)
F1, T <- TransformCloud(F1, a, dx, dy, dz)
end procedure
Using object detection for false alarm reduction

- Vehicle and pedestrian regions detected and excluded from the object based alignment step
- Camera data: 2D object detection by Mask R-CNN, then reprojection of results to SFM point cloud
- Lidar data: PointPillars based 3D object detection
Effect of dynamic object removal on object based registration
Curve based registration refinement

- Limitations of rigid body transform estimation
  - SFM pipeline artifacts - *deformations* due to *invalid depth calculation*
  - Issues with Lidar point clouds: *shape distortion* due to *vehicle motion*
- Control point based non-linear registration refinement
  - NURBS curves fit on the extracted object centers
Curve based registration refinement

Rigid transform based registration

Result of non-linear refinement
Some further results

Camera-LiDAR fusion result 1

Steps:
1. Camera-LiDAR self calibration
2. Detection and tracking LiDAR point clouds
   • Fully geometry based object detection
   • Kalman-filter-based motion model for tracking
3. Projection of the object point clouds to the camera frames
Thank you for your attention

• Contact: Csaba Benedek, benedek.csaba@sztaki.hu
• Contributors: Balázs Nagy, Örkény Zováthi
• Website:
  – Institute for Computer Science and Control, Machine Perception Research Laboratory
• Funding:
  – Hungarian National Research, Development and Innovation Fund, under grants NKFIA K-120233 and 2018-2.1.3-EUREKA-2018-00032;
  – Michelberger Master Award of the Hungarian Academy of Engineering