

Lightweight Solution to Generate Accurate Lanelet Maps [†]

Gergő Ignéczi ¹, Dávid Józsa ² and Mátyás Mesics ^{2,*}¹ Robert Bosch Kft., 1103 Budapest, Hungary; gergo.igneczi@ga.sze.hu² Zalaegerszeg Innovation Park, Széchenyi István University, 8900 Zalaegerszeg, Hungary; jozsa.david@ga.sze.hu

* Correspondence: mesics.matyas@ga.sze.hu

[†] Presented at the Sustainable Mobility and Transportation Symposium 2025, Győr, Hungary, 16–18 October 2025.

Abstract

As automated driving technologies become more mature, there is an increasing reliance on digital maps to support safe and efficient driving. Sensors like cameras and radars can be limited by occlusions, lighting conditions, or weather, and often fall short. High-definition (HD) maps offer excellent accuracy, but they are expensive to produce. These limitations make these techniques impractical for large-scale deployment. What makes our approach particularly attractive is its hardware simplicity: the entire process requires only a precise GNSS receiver and a commonly available lane detection camera, eliminating the need for expensive sensors like LiDAR or complex multi-vehicle fleets. We rigorously evaluated our method in a highway environment, where a vehicle equipped with our generated maps successfully executed autonomous lane following and adapted its speed based on detected speed limit signs. The positional deviation of the resulting maps was consistently under 5 cm.

Keywords: automatic map creation; lightweight static maps; lanelet map generation

1. Introduction

High-definition (HD) maps have become a cornerstone of automated driving technologies, particularly for vehicles operating at SAE Level 3 and above [1]. These maps offer a detailed, structured view of a driving environment that goes well beyond what onboard sensors can perceive in real time [2]. As vehicles are expected to perform increasingly complex driving tasks—such as hands-free highway cruising or active lane-keeping under varied conditions—map data becomes an unavoidable part of the systems that provides essential support. Among the various map formats available today, lanelet maps are widely adopted in academic and industrial applications due to their modular structure and strong compatibility with routing, planning, and regulatory feature layers [3,4]. They have been used in a range of domains, from simulation and behavior planning to real-world deployment [5–7]. Additionally, there has been meaningful progress in adjacent areas such as map validation and automated lane modeling, and several open-source or partially open-source pipelines have emerged in recent years [8,9]. These developments have helped push the field forward, offering a foundation for map-based functionality in both prototyping and production environments. However, despite these advances, many of the current solutions are still constrained by several critical shortcomings. Most notably, many pipelines are not optimized for lightweight deployment. They rely heavily on expensive hardware like LiDAR or multi-sensor fusion setups, which limits scalability and affordability [7].



Academic Editors: Szabolcs Kocsis-Szürke, Dániel Csikor and Boglárka Eisinger Balassa

Published: 13 November 2025

Citation: Ignéczi, G.; Józsa, D.; Mesics, M. Lightweight Solution to Generate Accurate Lanelet Maps. *Eng. Proc.* **2025**, *113*, 68. <https://doi.org/10.3390/engproc2025113068>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Furthermore, some approaches underperform in terms of the accuracy required for precise vehicle control. As a result, there is a growing need for solutions that strike a better balance between precision, cost, and ease of integration. In this research, we aim to address these gaps by developing a lightweight yet accurate toolchain for generating lanelet-format HD maps using minimal hardware. The main contributions of our work are as follows:

- C1 In contrast to existing, complex solutions, our proposed method relies only on a high-precision GNSS receiver and a lane detection camera configuration that is already present in many modern vehicles. This makes the integration and usage of our method easy.
- C2 Our generated maps only contain minimal information for vehicle control (i.e., lanes and traffic rules); therefore their real-time usage is possible, while their lane position precision is remarkable, further enhancing the applicability of this solution.

The process is divided into three main phases. In the pre-processing stage, we use MATLAB (2024b) to manually annotate reference lane lines from recorded data. This step helps anchor the rest of the pipeline to a reliable baseline. In the refinement phase, these references are further improved by aligning multiple measurements, increasing geometric accuracy across the map. Finally, the map generation step compiles the processed data into lanelet-format files using the official Python (3.10) API provided by the Lanelet2 library [8]. The usability of our generated maps has been demonstrated in a highway lane following a scenario with a real vehicle. This serves as a proof-of-concept for the broader application of our approach and highlights its potential for scalable and real-world deployment.

2. Materials and Methods

2.1. Data Collection

The first and arguably most critical step in the HD map generation pipeline is the collection of high-quality raw data. This step underpins the entire mapping process by establishing the geometric and spatial basis for the maps. Our approach to data collection was intentionally designed to balance practicality with technical precision: we used minimal sensor hardware that is readily available in production-level systems, while still achieving the accuracy required for advanced automated driving applications. The collected data served as the foundation for generating lightweight yet reliable lanelet-format maps. Our data acquisition efforts were focused on highways and rural roads, as these road types present structured but sufficiently variable environments. Highways offer well-defined lanes, consistent signage, and no pedestrian interaction—making them ideal for the initial development and validation of autonomous functionalities. The sensor setup employed during the data collection phase consisted of just two components: a high-precision GNSS receiver and a camera-based lane detection system. The GNSS unit is a Genesys ADMA Gen3 device [10], which was responsible for tracking the global position of the vehicle throughout each run. To ensure a positional accuracy adequate for HD map creation, we used a differential GNSS configuration capable of centimeter-level precision. The GNSS data was collected at a high rate and synchronized with camera readings to maintain spatial and temporal coherence. To maintain precise localization throughout the whole measurement, we used a Kalman filter to fuse vehicle odometry with the live GNSS position. For detecting lane boundaries, we used the Bosch MPC 2.5 camera [11], a production-grade forward-facing sensor designed for ADAS applications. This camera can detect both left and right lane markings in a variety of road conditions, and outputs structured lane line data in real time. This edge device is also capable of enhancing its measurements using the vehicle's odometry. It represents a realistic choice for real-world deployment, as similar sensors are already integrated into many modern passenger vehicles. The output of the camera includes the detected lane edges given by their position, orientation, and curvature

in the vehicle coordinate frame (1), shown in Figure 1. The GNSS system measures the pose of the vehicle in the global UTM frame (2), also illustrated in Figure 1. It should be noted that the lane edge information (1) is measured separately for the left and right edges of the lane, but they are supposed to be parallel, therefore their orientation and curvature are assumed to be the same. This is a production concept of the used camera and its corresponding software component.

$$l^{\text{left,right}}(t) = \begin{bmatrix} d_y^{\text{left,right}} & \theta & \kappa \end{bmatrix}^T \quad (1)$$

$$\rho(t) = \begin{bmatrix} X & Y & \Psi \end{bmatrix}^T \quad (2)$$

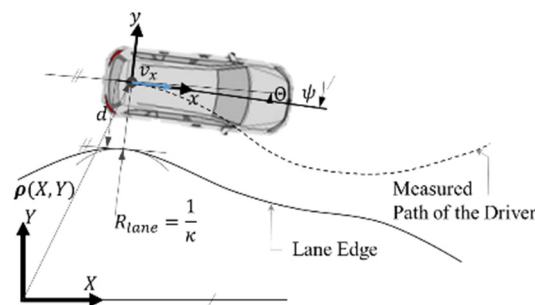


Figure 1. The relevant quantities measured by the video camera system of the vehicle, especially the lane edge position, orientation, and curvature.

During each data collection session, sensor readings were logged and stored in CSV format, which included synchronized entries for time, detected lane edge geometries $l^{\text{left,right}}(t)$, and global pose $\rho(t)$. This format enabled the easy inspection and preliminary filtering of the data. However, for more structured analysis and processing, we converted the raw CSV data into MATLAB ‘.mat’ files, which served as the working format throughout the pre-processing phase. The development data was collected at ZalaZONE test track, Hungary, in the motorway section, illustrated in Figure 2. The motorway segment is a built highway environment, with a long straight section and one curved section, with a total length of approximately 1.5 km.

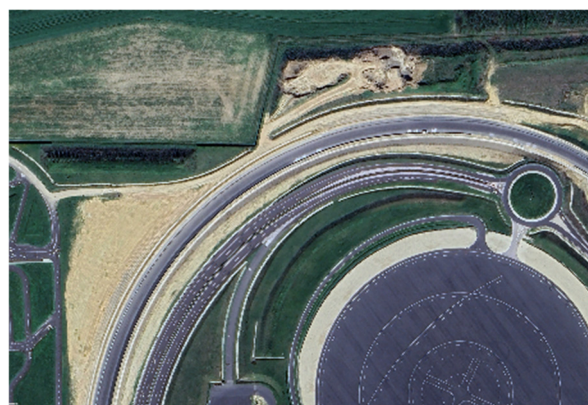


Figure 2. Example of the labeled road section. ZalaZONE test track, highway segment.

2.2. Pre-Processing Phase

Each file contains two primary datasets: a time series of the vehicle pose, and a time series of the lane geometry information for the left and right lane edges. In order to apply physical calculations on the lane markings, the curve parameters of (1) are transformed to lane points. This is possible as the position of each measured lane point is implicitly given

in (1), as the lane distance from the vehicle $d_y^{\text{left,right}}$ equally qualifies as the y coordinate of the given point in the vehicle coordinate frame $x - y$. Supposing a sample measuring time of T_s , (1) and (2) are discretized via measuring their samples. Thus, at sample time kT_s , where $k \in \mathbb{N}$, the lane edge point position, given in the vehicle coordinate frame, can be calculated (3).

$$p_k^{\text{left,right}}(kT_s) \equiv p_k^{\text{left,right}} = \begin{bmatrix} 0 & d_{y,k}^{\text{left,right}} & \theta_k \end{bmatrix}^T = \begin{bmatrix} 0 & y_k^{\text{left,right}} & \theta_k \end{bmatrix}^T \quad (3)$$

Based on (2–3), the global lane edge point position can be calculated (4).

$$P_k^{\text{left,right}} = p_k^{\text{left,right}} T_k + \rho_k \quad (4)$$

where $P_k^{\text{left,right}}$ is the global lane edge point pose at the k^{th} sample, and T_k is the rotational matrix (5).

$$T_k = \begin{bmatrix} \cos(\Psi_k) & \sin(\Psi_k) & 0 \\ -\sin(\Psi_k) & \cos(\Psi_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

The pre-processing stage requires the manual annotation of the measurement metadata, namely the road and lane identification in the global map. Following manual annotation, we perform a refinement phase in which data from multiple passes over the same road segments are integrated. This allows for the statistical filtering and averaging of repeated measurements, effectively smoothing out noise and improving the overall positional stability of the detected lane features. Given the vectors formulated from (4), the points of the same lanes are concatenated (6).

$$\begin{aligned} P^{\text{IID}-i} &= \text{col}(P_k^{\text{IID}-i}) \\ P^{\text{IID}} &= \text{col}(P^{\text{IID}-i}) \end{aligned} \quad (6)$$

Then, the lane edge line is divided into smaller sections, where a transformation is possible, after which the local longitudinal coordinate is strictly monotonically increasing (7).

$$P_j^{\text{IID}} = P^{\text{IID}}[(j-1)N : jN, :] \quad (7)$$

where $j, N \in \mathbb{N}$, and N are hyper parameters which define the length of the snippets. Choosing a low value of N guarantees the strictly monotonically increasing coordinates, but regression may fail or become unstable. Too high a value of N may result in non-monotonically increasing longitudinal coordinates but regression would be stable. The snippet points are transformed to a quasi-local coordinate frame, providing the strictly monotonically increasing coordinates (8).

$$\begin{aligned} \tilde{P}_j^{\text{IID}} &= (P_j^{\text{IID}} - P_j^{\text{IID}}[0, 1 : 2]) \tilde{T}_j^{\text{IID}} \\ \tilde{T}_j^{\text{IID}} &= \begin{bmatrix} \cos(P_j^{\text{IID}}[0, 3]) & -\sin(P_j^{\text{IID}}[0, 3]) & 0 \\ \sin(P_j^{\text{IID}}[0, 3]) & \cos(P_j^{\text{IID}}[0, 3]) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (8)$$

Using the quasi-local path points (8), polyline regression is applied, fitting a polynomial curve onto the snippet points (9).

$$y_j^{IID} = c_{j0}^{IID} + c_{j1}^{IID} x_j^{IID} + c_{j2}^{IID} (x_j^{IID})^2 + \dots + c_{jn}^{IID} (x_j^{IID})^n$$

$$c_j^{IID} = g(P_j^{IID}, n)$$
(9)

where $[x_j^{IID} y_j^{IID}]$ are the fitted polyline points of snippet j , and n is the order of the polynomial.

2.3. Post-Processing Phase

The result of the pre-processing is a robust set of lane edge points that can serve as reliable inputs for map construction. The idea is to collect all available detections within a spatial bounding box, typically sized at approximately 100 m in length and 20 m in width and compute a refined lane line by fitting a polyline on the collected data points, as introduced in Section 2. When repeated over the entire route, this method allows us to generate a comprehensive and continuous map of lane boundaries with a high positional accuracy. This refinement loop is repeated until all mappable segments are processed, ensuring consistent quality across the entire dataset. The output of this phase is a structured set of cleaned lane line points, which can now serve as the input for the HD map generation phase. To produce a usable map in a standard, machine-readable format, we utilize the Lanelet2 Python API [2], a widely adopted open-source library for HD map creation and manipulation.

3. Results

The results are generated using self-developed MATLAB scripts that accumulate and average lane line data from repeated passes over the same location. Our pipeline converts the processed lane line data into Lanelet2 map files by defining individual lanelet primitives, which include left and right lane boundaries, regulatory elements (e.g., speed limits), and lane centerlines, as illustrated in Figure 3.

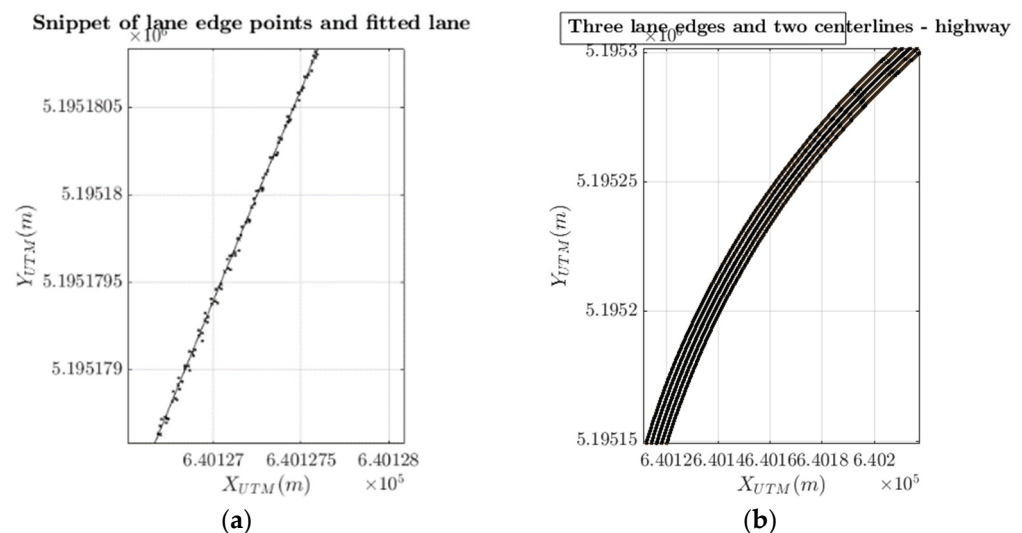


Figure 3. (a) Highlighted snippet of one lane edge, illustrating the data points from various measurements and the regression result, (b) two lanes of the highway from the development data, including the centerlines of the lanes.

The absolute fit error between the fitted polylines and the measured lane points is calculated (10) for all measured points for the centerlines of the two lanes shown in Figure 3b. Using the error vectors, statistics are calculated, namely the mean absolute fit error and the standard deviation of the error [12,13]. The results are shown in Figure 4. The

mean absolute fit error is below 3 cm for both centerlines, as well as the standard deviation. This means that 95% of the measured lane points are within 2 ± 5 cm proximity of the fitted centerline. This result proves that the fitted centerline can be used for vehicle control purposes, as most modern controllers can provide a tracking accuracy up to ± 20 cm.

$$e_j^{IID} = \left| \tilde{P}_j^{IID}[:, 2] - y_j^{IID} \right| \quad (10)$$

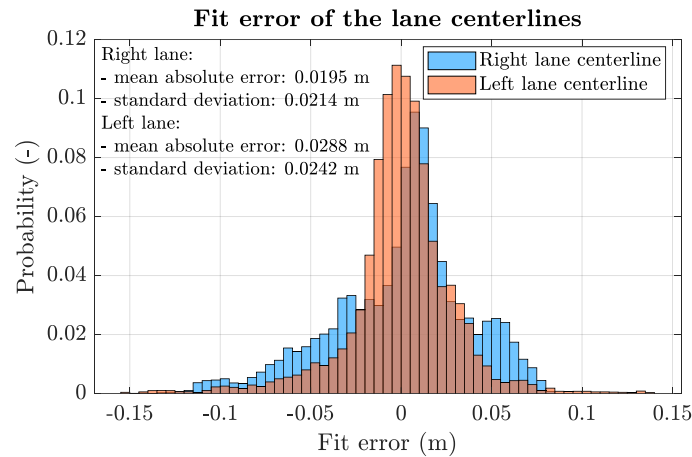


Figure 4. Statistical results of the centerline fit error for the two lanes in the development highway dataset.

The generated map can be directly used in downstream modules such as motion planning or trajectory tracking. The resulting map is then integrated into our autonomous driving framework and used to replace traditional, real-time lane detection systems in test environments. By using the generated Lanelet2 maps [14], we can guide the vehicle along centerlines with improved stability and foresight, overcoming the inherent limitations of real-time perception methods that may fail under suboptimal conditions. The map-based control significantly improves lane-keeping performance, particularly in scenarios involving curves, occluded markings, or inconsistent road paint. The maps used in the autonomous chain are illustrated in Figure 5.

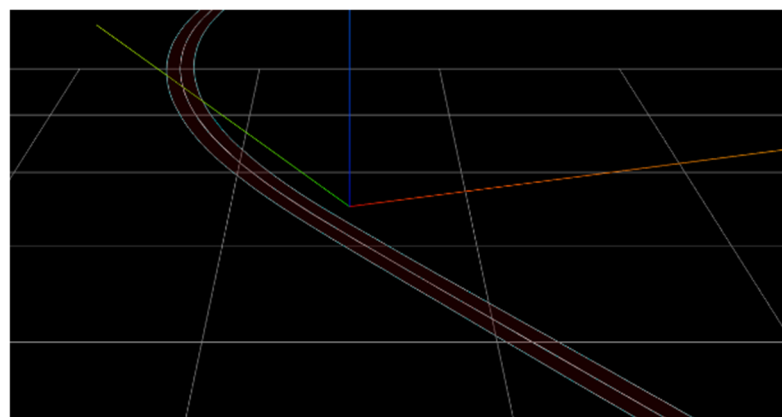


Figure 5. Generated lanelet, from the lanelet viewer, which is directly used in the autonomous driving system.

4. Conclusions

In our work, we represented a way of generating Lanelet2 maps and refining them over time. Throughout the development we have acquired measurement data from different routes and environments. With future developments in mind, we will try to unify the

generation pipeline into a Python-only solution. Also, in some environments, mainly in rural and country roads with harsh turns, the pipeline could not generate usable maps, and this is definitely something we would like to fix in future developments. These automatically generated maps can be used in ADAS features such as in Active Lane Assist, SCC (Smart Cruise Control), and later also in Level 3 functions and beyond. In comparison to traditional HD maps, we believe our solution could be scaled easily and provide a more up-to-date map compared with the use of HD mapping equipment.

Author Contributions: Conceptualization, G.I.; methodology, G.I.; investigation, M.M.; resources, D.J.; data curation, D.J.; writing—original draft preparation, G.I., M.M. and D.J.; writing—review and editing, G.I. and M.M.; supervision, M.M.; project administration, G.I.; funding acquisition, G.I. All authors have read and agreed to the published version of the manuscript.

Funding: The publication was created in the framework of the Széchenyi István University's VHFO/416/2023-EM_SZERZ project entitled "Preparation of digital and self-driving environmental infrastructure developments and related research to reduce carbon emissions and environmental impact" (Green Traffic Cloud).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data are available within the paper.

Conflicts of Interest: Robert Bosch Kft has no commercial conflict of interest. The authors declare no conflicts of interest.

References

1. J2399_201409; Adaptive Cruise Control (ACC) Operating Characteristics and User Interface. SAE International: Warrendale, PA, USA, 2014.
2. Gamal, E.; Raphaël, F.; Scott, H.; Stefan, S. High-Definition Maps: Comprehensive Survey, Challenges, and Future Perspectives. *IEEE Open J. Intell. Transp. Syst.* **2023**, *4*, 527–550. [CrossRef]
3. Naumann, A.; Hertlein, F.; Grim, D.; Zipf, M.; Thoma, S.; Rettinger, A.; Lavdim, H.; Luettin, J.; Schmid, S.; Caesar, H. Lanelet2 for nuScenes: Enabling Spatial Semantic Relationships and Diverse Map-Based Anchor Paths. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Vancouver, BC, Canada, 18–22 June 2023.
4. Poggenhans, F.; Pauls, J.-H.; Janosovits, J.; Orf, S.; Naumann, M.; Kuhnt, F.; Mayr, M. Lanelet2: A High-Definition Map Framework for the Future of Automated Driving. In Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018.
5. Fabian, I.; Richard, F.; Frank, B.; Christoph, S. Generation of Training Data from HD Maps in the Lanelet2 Framework. *arXiv* **2024**, arXiv:2407.17409. [CrossRef]
6. Matteo, B.; Paolo, C.; Simone, M. Semantic Interpretation of Raw Survey Vehicle Sensory Data for Lane-Level HD Map Generation. *Robot. Auton. Syst.* **2024**, *172*, 104513.
7. Mengmeng, Y.; Jiang, K.; Benny, W.; Tuopu, W. Review and Challenge: High Definition Map Technology for Intelligent Connected Vehicle. *Fundam. Res.* **2024**, *in press*.
8. Lagahit, M.; Matosouka, M. gpkg2lanelet v1.0: A Python-Based Conversion Tool That Converts HD Map Vector Primitives from Geopackage Format to Lanelet2 Format. In Proceedings of the International Symposium on Applied Geoinformatics (ISAG2021), Online, 2–3 December 2021.
9. Doer, C.; Hanzler, M.; Messner, H.; Tromer, G.F. HD Map Generation from Vehicle Fleet Data for Highly Automated Driving on Highways. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 21–24 June 2020.
10. Genesys Sensors and Navigation Solution. User Manual ADMA-G, ADMA-Speed, ADMA-Slim. In *ADMA Manual*; GeneSys Elektronik GmbH: Offenburg, Germany, 2019. Available online: <https://genesys-offenburg.de/support/guides-and-manuals/adma/adma-manual/> (accessed on 6 August 2025).
11. Robert Bosch GmbH. Chassis Systems Control Second Generation Multi Purpose Camera (MPC2). 2013. Available online: <https://www.bosch-mobility.com/en/solutions/camera/multi-purpose-camera/> (accessed on 6 August 2025).
12. Bender, P.; Ziegler, J.; Stiller, C. Lanelets: Efficient Map Representation for Autonomous Driving. In Proceedings of the IEEE Intelligent Vehicles Symposium, Ypsilanti, MI, USA, 8–11 June 2014.

13. Harald, S.; Eder, S.; Andrew, H.; Riccardo, B. A Commute in Data: The comma2k19 Dataset. *arXiv* **2018**, arXiv:1812.05752. [[CrossRef](#)]
14. Poggenhans, F.; Janosovits, J. Pathfinding and Routing for Automated Driving in the Lanelet2 Map Framework. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–7.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.