# Bird's-eye view image acquisition from simulated scenes using geometric inverse perspective mapping

Daniels Jānis Justs, Rihards Novickis, Kaspars Ozols, Modris Greitāns
*Institute of Electronics and Computer Science*
Riga, Latvia
{daniels.justs,rihards.novickis,kaspars.ozols,modris_greitans}@edi.lv

***Abstract*—Technology advancement, major investments and mainstream interest have reignited the prospect of autonomous driving, nevertheless, the achievement of reliable perception is still an active research topic. This article examines a technique for simplifying path planning and control algorithms by reducing spacial dimensions to two, i.e. by acquiring a Bird's-Eye View (BEV) image. This article aspires to facilitate novel perception fusion approaches by specifying a geometric approach for inverse perspective mapping (IPM), construction of composite BEV of the vehicle, giving suggestions for implementation and proposing coherent, BEV-based approach for perception fusion. The proposed geometric IPM approach for BEV image acquisition is suited for simulated environments with limited knowledge of the camera's intrinsic parameters.**

***Keywords*—IPM, Inverse Perspective Mapping, Bird's-Eye View, BEV, Autonomous Driving, Perception Fusion**

## I. Introduction

The technology advancement, major investments and mainstream interest in upcoming autonomous driving applications have led to a major research interest in all aspects of autonomous driving, especially the aspect of reliable perception [1], [2]. Although AI-based algorithms bear the potential for a variety of applications [3] including autonomous driving, a robust overall perception model of an autonomous vehicle is still open for discussion.

One prospect of achieving improved safety of autonomous systems is fusing data of different kind of sensors and perception algorithms. Although, in mainstream media it has been famously stated that autonomous vehicles can manage autonomous functions without using LIDARs, it is still suggested that wide and global adaptation of autonomous vehicles still would need to compensate for shortcomings of different sensors [4].

We examine a technique for approaching path planning and control challenges by reducing the spacial dimensions to two, usually this approach is referred to as a *Bird's-Eye View* (BEV) or *Inverse Perspective Mapping* (IPM) of camera images.

Although IPM can be realised by a trivial matrix multiplication and even described with a look-up-table (LUT) for better performance, by performing the geometric analysis of camera image projection to two dimensions we can account for and compensate for the instabilities in suspension, accidental movement of the camera and to enable real-time data fusion with other kinds of sensors, e.g. LIDARs.

In this paper, we describe in detail the geometric reasoning of IPM and propose a novel, reduced in terms of known camera's intrinsic parameters, IPM approach, give recommendations on the implementation of the algorithm, show results of *Hardware-in-the-Loop* (HIL) setup and propose an approach for perception fusion which is based on the composition of multiple BEV images.

## II. Related work

In [5], a real-time automotive surround view camera system is presented to allow for a driver to see a BEV of the vehicle. Assuming the ground being a 2D flat surface, the authors used a calibration pattern placed on the ground to perform lens correction and to align a camera pair to each other, therefore they did not need to know exact camera's extrinsic parameters nor intrinsic parameters to acquire the BEV image.

An FPGA implementation for BEV image generation using high-level synthesis was presented in [6]. The authors proposed their own IP core to perform perspective transformation and generate a BEV image. They used polynomial approximation of the image transformation function, i.e., homography matrix, which was implemented in the FPGA. Homography matrix is used to warp the image perspective to a BEV, but it must be noted that the homography matrix coefficients must be found through a calibration process by placing a calibration image in the scene. Their proposed IP core's results were compared to a full software solution to evaluate the precision loss due to the use of fixed point number representation and the approximation of the transformation function.

In [7] a method for using multiple cameras to acquire a composed mosaic BEV is proposed. The author is combining a single BEV image of the *Region of Interest* (ROI), which is observed by multiple cameras from different perspectives. Most notably, the author is proposing to evaluate each camera's instantaneous sampling rate of the ROI to determine, which camera's perspective should be used in the final BEV image. One camera's BEV is selected as the reference frame while a mapping function is required to align images from other camera's to the reference. Homography transformation was used to transform the perspective image to a BEV image.

Adaptive IPM algorithm to acquire a BEV image was presented in [8], which used visual simultaneous localisation and mapping (SLAM) algorithm to correct for the distortion

caused by incorrect assumption of camera's extrinsic parameters. Their geometric IPM approach assumes that the intrinsic parameters of the camera are known as well as the extrinsic parameters as they are used in the acquired camera image transformation to a BEV image. Additionally, SLAM was utilised to infer from the camera images the change in its pitch angle relative to the ground as it changes due to the movement of the vehicle. By using the acquired pitch angle change in IPM, the BEV image was corrected from the distortions.

Stabilisation of IPM algorithm utilising vanishing point detection is presented in [9]. Pre-processing of the image is used to find the lane markings which are then used to find the point at which they seemingly converge, i.e., the vanishing point. With this knowledge it is possible to determine the extrinsic parameters - pitch and yaw - of the camera, therefore stabilisation and dynamic adaptation of the IPM is possible. Their IPM approach involves the use of a homography matrix to transform the acquired perspective image to a BEV image.

As it can be seen in the related work, the previous approaches are concerned with real-world image processing where it is possible to calibrate the used camera and acquire the intrinsic and extrinsic parameters. Most of the previous work utilises homography matrix to transform the acquired images to a BEV image, which is suitable when you can calculate the homography matrix coefficients by placing a calibration object on the ground plane and warp the camera image to the BEV perspective.

As in the simulated environments or during an actual exploitation of the vehicle where sensor placement can be altered by mechanical forces and suspension fluctuations, on-the-fly re-calibration of the cameras is not possible and geometrical model of IPM is favoured. Furthermore, BEV images from different cameras can be fused with other perception data to create a coherent view of vehicle's surroundings, which is further fed into the path planning and trajectory generation algorithms and brings a modular unified model, which can be improved over time while not impairing the control of the vehicle. This paper aims to present a more generic approach to perform IPM to acquire a BEV image of an area of interest by limiting the knowledge of the camera parameters only to its image resolution and *Field of View* (FOV),

## III. Proposed Approach

### A. Inverse Perspective Mapping

The general assumption for the acquisition of bird's-eye view (BEV) images using inverse-perspective mapping (IPM) is that the area of interest is assumed to be flat. This assumption is reasonable when it comes to performing IPM on an image of the road around the vehicle. In the proposed geometric IPM approach we assume that we have at least one of the camera's FOV angles available (as the other one can be inferred from, for example, the known one and image resolution) and its extrinsic parameters (the height and the relative angle to the scene). We also assume that the camera has no roll against the scene at this moment as this can be corrected independently. In Fig. 1 a 3D geometric model

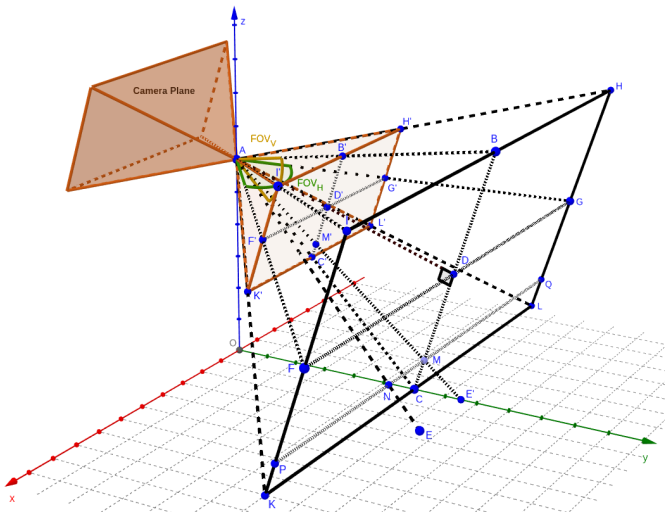


Fig. 1. 3D geometrical representation of inverse-perspective mapping to acquire a bird's-eye view from a camera (created using GeoGebra)

of how the camera is positioned and how it perceives the flat ground and the scene is shown to help explain the used transformation principles.

Another assumption is concerning the camera itself; the pinhole camera model is assumed with the pinhole as point A in Fig. 1 and the camera's sensor plane is marked with text. The image of the scene on the camera's focal plane is flipped, but we can look at the unflipped image of the scene or the camera's virtual image when we look at the rectangular pyramid $AI'H'L'K'$ [10]. The rectangle $I'H'L'K'$ is the virtual image of the camera, with which we get a reference plane for the actual image the camera is "seeing". We can use this reference plane to map the points of interest from the scene (BEV image points) to the actual image from the camera as the pixels of the image can be mapped directly to the reference plane, e.g., point $M'$ from point $E'$.

As it can be seen in the Fig. 1, the reference plane can be mapped to another rectangle $IHKL$. This rectangle is of the same scale as other objects in the scene, therefore it is essential for the transformation of the scene points into camera's image points. With the aforementioned rectangle we can view the rectangular pyramid $AIHLK$, which is similar to the pyramid $AI'H'L'K'$ and shares the same angles. This means that the rectangles $IHLK$ and $I'H'L'K'$ are just scaled versions of each other, therefore we can map the rectangle $IHLK$ directly to the input image and look for the input image pixels on it, e.g., the length of the segments $KL$ and $IH$ is mapped to the width of the input image in pixels and, similarly, the heights $KI$ and $LH$ are mapped to the height of the input image in pixels. Therefore, if we find the location for the searched points from the scene in the rectangle $IHKL$, we can calculate where it will be in the input image.

The flat ground plane of the BEV image is the xy-plane of the Fig. 1. In the figure, we can see that on the ground plane lies segment $KL$, which is a projection of the lowest border

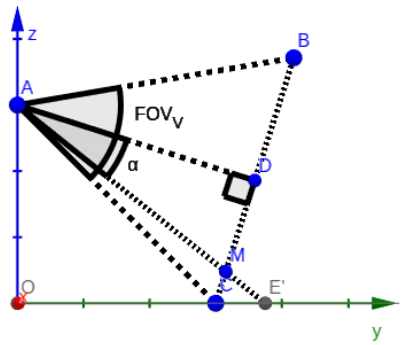


Fig. 2. Side view of Fig. 1 for calculation of row input image pixels from the longitudinal distance

of the input or camera's image plane $K'L'$, therefore segment $KL$ draws a border of the visible scene in the input image.

To start off, we look for the input image pixels from the perspective of the output image or BEV image. The BEV image can be viewed as a defined area of which we are interested with defined pixel resolution and defined metric dimensions of the area, therefore each pixel of the BEV image can be viewed as an elementary square with defined dimensions in the actual scene's ground plane.

Point $E$ is the point of interest (POI) in the BEV image, which is in the center of the elementary square, and we are looking for the input image pixel in which we can see this point. To find the input image pixel corresponding to the POI, we first split this task in two parts - finding the corresponding row and then the corresponding column of the input image.

To find the row of the input image pixel corresponding to the point $E$, we can look at the Fig. 2, which is the side view of Fig. 1. Point $E'$ is the projection of the point $E$ on the y-axis (as can be seen in Fig. 1). As it was previously established, the segments $KI$ and $LH$ can be mapped to the height of the input image as well as the segment $CB$, therefore in the Fig. 2 we are looking at cross-section, i.e., the triangle $ABC$ in order to find the input image's row corresponding to the BEV image point $E'$. The segment $AE'$ is the projection of the segment $AE$ on the triangle $ABC$ and the point $M$ is an intersection of the segments $AE'$ and $CB$. Therefore, to find the corresponding row, we need to find any distance related to the point $M$ and the segment $CB$.

First, we want to calculate the angle $\alpha$ and length $AD$ as they will aid us in calculating the distance $MD$ with which we can get the row of the input image. We assume that the distance $OE'$ is known or can be calculated from the knowledge of the POI distance in the scene and the camera's extrinsic parameters. For example, if the POI distance in y-direction is referenced from the lowest border of the mapped image plane $KL$, then the distance $CE'$ is known. As for the distance $OC$, we can calculate it by utilizing the known extrinsic parameters of the camera - we know the height at which it is positioned $OA$ and we know the angle $\angle OAD$ as it is the tilt or the pitch angle of the camera. To get the angle $OAC$:

$$\angle OAC = \angle OAD - \angle FOV_V/2 \tag{1}$$

where $\angle FOV_V$ is the vertical FOV angle of the camera. After that we calculate the length $OC$:

$$OC = OA \tan \angle OAC \tag{2}$$

Now that we have $OC$, we can find $\angle OAE'$ and then calculate $\angle \alpha$:

$$\angle \alpha = \angle OAD - \angle OAE' = \angle OAD - \arctan \frac{OE'}{OA} \tag{3}$$

To find the needed length of $MD$, we first need to find the lengths $AD$ and $AC$:

$$AC = \sqrt{OC^2 + OA^2} \tag{4}$$

$$AD = AC \cos \angle FOV_V/2 \tag{5}$$

$$MD = AD \tan \angle \alpha \tag{6}$$

As it was previously established, the segment $CB$ is proportional to the height of the image, therefore $CD$ is half the height. With this we can define a proportionality coefficient $k_v$:

$$CD = AC \sin(FOV_V/2) \tag{7}$$

$$k_v = \frac{img_H/2}{CD} \left[\frac{\text{px}}{\text{m}}\right] \tag{8}$$

where $img_H$ is the height of the image. By multiplying the calculated lengths on the segment $CB$ with $k_v$, we can get the row of the input image $r_i$:

$$r_i = k_v \cdot MD \tag{9}$$

where, as in this case, $MD$ is the distance from the middle of the image, therefore in this case we add $img_H/2$ to calculated $r_i$.

Now we have calculated the row of the input image $r_i$ in which we can see the POI of the BEV image. The calculated point rarely points to a pixel directly, therefore interpolation can be used to find the exact value. Most trivial method of interpolation would be to just use the whole part of the calculated row as this gives the index of the row in the input image. Usually, nearest-neighbour interpolation is assumed as the easiest interpolation method and is most used for a crude result as it rounds the calculated numbers to the nearest integer value.

Now we can find in a similar manner the column of the input image $c_i$. To find the column of the input image, let's find the length of segment $NM$, which will be proportional to the pixel offset from the image centre (Fig. 1). To put this into another perspective for easier observation, the reader can look at Fig. 3. Segment $PQ$, on which lies the segment $NM$, is of the same length as the segments $KL$ and $IH$, which, as was previously noted, are proportional to the width of the input image. First we'll consider the triangle $EAE'$ as it shares the angle $\angle EAE'$ with the triangle $NAM$, from which we can find the length of the segment $NM$.

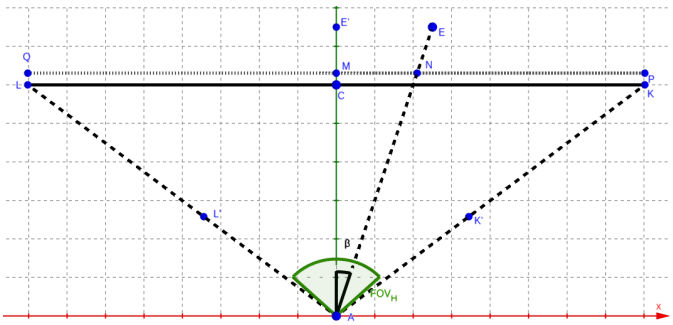

Fig. 3. Top view of Fig 1 for calculation of column image pixels from the lateral distance

First we need the length of $AE'$:

$$AE' = \sqrt{OE'^2 + OA^2} \tag{10}$$

As for the length $EE'$, it is already known as it is the distance from the y-axis or defined from the POI in the BEV image. The angle $\angle EAE'$ can be found with the following expression:

$$\angle EAE' = \arctan \frac{EE'}{AE'} \tag{11}$$

Now that we have the shared angle $\angle EAE'$ between of the two triangles $EAE'$ and $NAM$, we can calculate the segment lengths associated to the triangle $NAM$. Now we need to calculate at least one length from this triangle to further calculate the needed length of $NM$. From the previous calculations we can use the known angle $\angle\alpha$ and the length $MD$ to calculate $AM$, which will lead us to $NM$:

$$AM = \frac{MD}{\sin \angle\alpha} \tag{12}$$

$$NM = AM \tan \angle EAE' \tag{13}$$

Now we have to find the proportionality coefficient $k_h$ for the horizontal direction. As we can see from the top view in Fig. 3, the angle $\angle FOV_H$ is only known in the middle of the pyramid $AIHLK$ and it is not the same on the sides of the pyramid, therefore we need to find the length $AD$ as with it we can calculate $DF$, with which we can calculate the proportionality coefficient $k_h$:

$$AD = CD \cot \angle FOV_V \tag{14}$$

$$k_v = \frac{img_W/2}{DF} = \frac{img_W/2}{AD \tan(\angle FOV_H/2)} \tag{15}$$

Now we can find in which column from the middle is the pixel that we are looking for:

$$c_i = k_v \cdot NM \tag{16}$$

Just as before, the pixel we calculated might be between multiple pixels, therefore we need to either interpolate between the neighbouring values or use the value of the closest pixel.

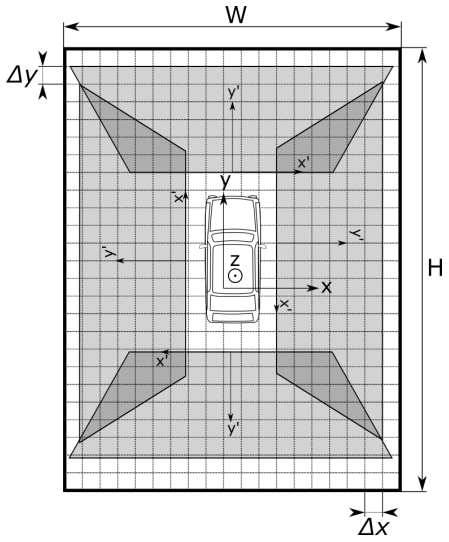


Fig. 4. Common coordinate frame for the fusion of the BEV images

### B. Common Coordinate Frame for a Unified Bird's-Eye View

As the IPM transformation is only performed concerning the camera, it has no relation to the relative location of the camera against the vehicle. To acquire a vehicle-centred BEV image, a common coordinate frame is considered for the BEV perception of the environment around the vehicle (see Fig. 4).

The common coordinate frame or the Region of Interest (ROI) is defined as vehicle-centred BEV image of the area around the vehicle, which is acquired by using multiple cameras and their individually acquired BEV images. The ROI has a defined metric area around the vehicle with width $W$ and height $H$. Additionally, this area is split into a grid with grid element dimensions $\Delta x$ and $\Delta y$ for the width (x-axis) and height (y-axis) respectively. The grid will correspond to the acquired vehicle-centred BEV image's pixel grid.

As can be seen in Fig. 4, the vehicle is in the centre of this proposed frame with four trapezoids around this vehicle, which indicate the visible field of an individual camera. The trapezoids have overlapping areas, therefore some form of image fusion is required.

When IPM was discussed in Section III-A, the acquired BEV image from a single camera was relative to the bottom border of the acquired perspective image, thus, every camera has it's own relative coordinate system $x'Oy'$ (Fig. 4), therefore we need to perform translation of coordinate axis to map individual camera's BEV images relative to the vehicle's coordinate system $xOy$.

The translation of the coordinate axis is performed by subtracting the location of the camera in the vehicle coordinate system and the distance $OC$ or the distance from the camera to the bottom line of the camera's BEV image from the absolute coordinates of the vehicle coordinate system:

$$v'_x = v_x - x_{\text{camera}} \tag{17}$$

$$v'_y = v_y - y_{\text{camera}} - OC_{\text{camera}} \tag{18}$$

As for the rotation, the yaw angle $\angle\theta$ of the camera is used to rotate the coordinate axes:

$$v''_x = v'_x \cos\angle\theta - v'_y \sin\angle\theta \tag{19}$$

$$v''_y = v'_x(-\sin\angle\theta) + v'_y \cos\angle\theta \tag{20}$$

This vehicle-centred approach provides room for additional sensor data to be fused together with the BEV images for improved perception of the environment surrounding the vehicle if the sensors in mind are calibrated against the vehicle itself as the common coordinate frame is relative to the vehicle's centre.

### C. Bird's-Eye View Based Fusion

One of the key suggestions of the BEV concept is to employ it as a baseline for the perception fusion. Firstly, camera images prior to IPM are processed by a *Convolutional Neural Network* (CNN) which performs semantic segmentation. This CNN categorises each pixel in a number of different classes, e.g. road, road marking, traffic sign, vehicle, motorcycle, pedestrian, etc.

At this point mapped and stitched semantic images produce a coherent BEV which already can be used for trajectory planning. Nevertheless, to improve on the performance characteristics of the vehicle, output of other perception modules, i.e. RADAR object detection, LIDAR object detection, traffic analysis, can be mapped on top of the produced BEV.

To find out more on the composition and implementation of a such autonomous driving concept, please refer to [11].

## IV. Implementation details

The proposed geometric IPM method is evaluated in CarLA (*Car Learning to Act*) - an open-source simulator based on Unreal Engine 4 for autonomous driving research [12], in which it is possible to setup a cameras with a defined acquired image resolution and its FOV.

The acquired data from the simulator are sent to NVIDIA Drive PX2 platform, where the developed algorithms for autonomous driving are deployed and are evaluated in a HiL manner. The PX2 processes the data and sends its vehicle commands back to the CarLA to execute them in the simulated environment.

To achieve higher processing performance of the proposed IPM method, it is performed beforehand, and the mapping of the pixels from the input image to the output or the BEV image is stored in a *Look-Up Table* (LUT). The LUTs addresses correspond to pixel indices of the output image while the LUTs values correspond to the indices of the input image, therefore, it is possible to iteratively fill the output image with the matching input image pixels without extensive calculations.

This physical setup is a part of the Car-in-the-Loop tests performed with the EDI Drive-by-Wire car [11]. To emulate the actual sensors used in the in the vehicle - Sekonix SF3325-100 - the CarLA camera sensor resolution is set to 1928x1208 (2.3M pixels) while horizontal FOV is set to 60°.

## V. Results

The proposed geometrical IPM approach is first evaluated with a single simulator camera image shown in Fig. 5. As it can be seen in the figure, the acquired BEV image (on the right of the Fig. 5) looks like a top-down view of the perspective view of the scene (on the left of the Fig. 5). The scene is from CarLA map `Town04`, the coordinates of the camera sensor - (384.5; -52.8), it is fixed on a test vehicle in the simulation environment with it's height above the ground 1.79 m, pitch -10°, yaw -90°.

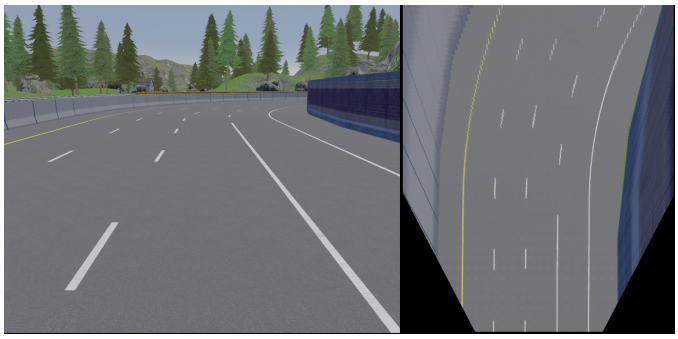

Fig. 5. BEV image (right) acquired using the proposed geometrical IPM approach on a CarLA simulator image (left)

As it can be seen in Fig. 5, the quality of acquired IPM image decreases as the distance from the camera increases. This is due to the perspective view of the camera for the objects further away from the camera. As points in the scene are further away, the individual pixel of the camera sensor corresponds to larger area in the scene, therefore the acquired BEV image exhibits jagged lines for road markings further away from the camera. The jaggedness can be reduced if interpolation is used as a processing step when performing IPM.


Fig. 6. BEV image (right) acquired using the proposed geometrical IPM approach on camera image taken on a EDI DbW Car test drive in Teika, Riga, Latvia (left)

To further evaluate the proposed IPM approach, in Fig. 6 the images taken on a test drive with EDI DbW Car ( [11]) are processed to acquire a BEV image of a real-world scene. As can be seen in the acquired BEV image, the approach can be

used on real-world images as well as the simulated ones. Both Fig. 5 and 6 show how the objects are obstructing the view of the flat ground, nevertheless, this does not obscure the directly feasible driving region. For more advanced manoeuvres, this obstruction can be taken into account by utilising other perception modules.

In standalone tests the LUT generation takes $545.0\,\text{ms}$ ($\sigma = 4.5\,\text{ms}$), while the application of the LUT for a single input-output image pair approximates to $3.1\,\text{ms}$ ($\sigma = 21\ \mu\text{s}$).

BEV image fusion with multiple simulated cameras is presented in Fig. 7. The simulated front and back cameras have $\text{FOV} = 60°$, but side cameras have $\text{FOV} = 120°$. The proposed common coordinate frame utilising a simple fusion algorithm of just filling in the BEV images from individual cameras would serve as the baseline for further perception fusion.

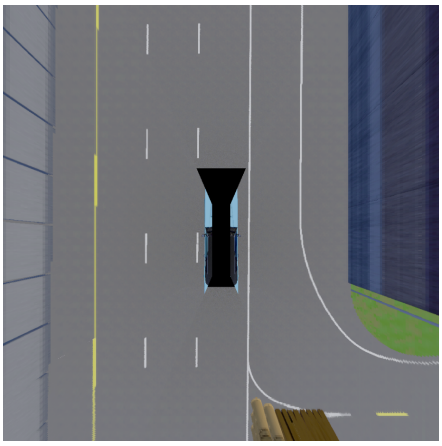

Fig. 7. BEV image fusion to a common image frame around the vehicle

## VI. Conclusion

We have presented a detailed derivation of an IPM algorithm based on geometry and pinhole camera model for the acquisition of BEV images. The proposed IPM approach, comparing to previous work, can be utilised in cases where the camera's intrinsic parameters are not fully known or their acquisition is restricted, e.g., in simulated environments.

By only knowing the camera's FOV and image resolution, assuming the pinhole camera model, it was presented that our IPM approach produces BEV from simulated camera images. Additionally, our approach was tested on the images acquired during a test run with EDI DbW Car [11], where the acquired image preprocessing was done in order to remove the barrel lens distortions before applying the IPM approach.

Furthermore, a BEV-based extendable perception fusion concept has been suggested, where multiple BEV images are fused into a coherent two-dimensional image to incorporate results of the semantic segmentation and other perception algorithms. Upon this concept it is proposed that data from other sensors could be fused in order to acquire more robust deduction of the surrounding environment and to reliably use the BEV image for path planning and control of the vehicle.

The algorithm has been implemented in an embedded automotive platform - NVIDIA Drive PX2 - where it has been tested in a simulated HiL setup and real-world environment. The allocation and the application of the LUT for the realisation of the IPM algorithm has been benchmarked in order to evaluate the usage of the proposed approach in a real-time system. While the usage of the LUT by itself has room for more stringent demands on the amount of data to be processed, the allocation of LUT itself is not suitable for real-time recalculation yet and has to be optimised.

The future work includes the enhancement of the BEV-based perception concept with a real-time data of the vehicle's angle against the ground plane and complemented with perception modalities from semantic segmentation, object detection and traffic analysis.

## Acknowledgment

This work is the result of activities within the "Programmable Systems for Intelligence in Automobiles" (PRYSTINE) project, which has received funding from ECSEL Joint Undertaking under grant agreement No. 783190 and from specific national programs and/or funding authorities.

## References

[1] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, Mar. 2020, conference Name: IEEE/CAA Journal of Automatica Sinica.

[2] N. Druml, G. Macher, M. Stolz, E. Armengaud, D. Watzenig, C. Steger, T. Herndl, A. Eckel, A. Ryabokon, A. Hoess, S. Kumar, G. Dimitrakopoulos, and H. Roedig, *PRYSTINE - PRogrammable sYSTems for INtelligence in automobilEs*, M. Novotny, N. Konofaos, and A. Skavhaug, Eds. Los Alamitos: Ieee Computer Soc, 2018, pages: 618-626 Publication Title: 2018 21st Euromicro Conference on Digital System Design (dsd 2018) WOS:000537466600091.

[3] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231216315533

[4] E. Commission, *Multi - Annual Strategic Plan ("MASP")*. European Commission, 2019.

[5] B. Zhang, V. Appia, I. Pekkucuksen, Y. Liu, A. U. Batur, P. Shastry, S. Liu, S. Sivasankaran, and K. Chitnis, "A Surround View Camera Solution for Embedded Systems," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2014, pp. 676–681.

[6] M. Bilal, "Resource-efficient FPGA implementation of perspective transformation for bird's eye view generation using high-level synthesis framework," *IET Circuits, Devices Systems*, vol. 13, no. 6, pp. 756–762, 2019.

[7] R. Laganiere, "Compositing a bird's eye view mosaic," *VI'2000*, pp. 382–387, 2000.

[8] J. Jeong and A. Kim, "Adaptive Inverse Perspective Mapping for lane map generation with SLAM," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. Xian, China: IEEE, Aug. 2016, pp. 38–41.

[9] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of Inverse Perspective Mapping Images based on Robust Vanishing Point Estimation," in *2007 IEEE Intelligent Vehicles Symposium*, Jun. 2007, pp. 315–320.

[10] K. Hata and S. Savarese, "CS231A Course Notes 1: Camera Models."

[11] R. Novickis, A. Levinskis, R. Kadikis, V. Fescenko, and K. Ozols, "Functional architecture for autonomous driving and its implementation," *Baltic Electronics Conference*, 2020.

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.