Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 190 (2021) 695–700

Procedia Computer Science

www.elsevier.com/locate/procedia

2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society

Alternative Inverse Perspective Mapping Homography Matrix Computation for ADAS Systems Using Camera Intrinsic and Extrinsic Calibration Parameters

Andrejs Rudzitis[a,*], Margarita A. Zaeva[a]

[a]*National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe shosse, 31, Moscow 115409, Russian Federation*

Abstract

This paper presents a simple novel IPM matrix computation method, which does not require any additional solutions to calculate homography matrices. This method only requires determining the distance from camera to projecting surface. It's simplicity allows execution even on very small energy-efficient embedded device processors without use of complex libraries such as OpenCV, that cannot be deployed there. Furthermore, it has strictly analytical solution, which allows avoidance of false optimums that are probable in heuristic methods for optimization. We provide formulas, that are easy transferable to any device and can be calculated using standard math libraries.

© 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society

Keywords: IPM; Homography; Bird eye view; calibration; projection

1. Introduction

Advanced Driver Assistance Systems (ADAS) are getting more popular each year. They are a necessity for most modern vehicles. With help of these systems, dangerous road accidents can be greatly reduced. Currently there are two tasks in driver assistance system that require a "birds-eye-view" from the camera. These tasks are surround view ([1], [2]) and lane departure warning [3]. With their help driver can more fully assess environment around the vehicle and receive notifications in case of breaking the law.

* Corresponding author. Tel.: +7-495-788-5699.
E-mail address: andreyrud@inbox.lv

1877-0509 © 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society
10.1016/j.procs.2021.06.081

2. Existing solutions

Most common method for ”birds-eye-view” image construction are perspective transformation matrices. When working with a two-dimensional space four reference points are selected and after that, their position on a transformed image is specified. Based on this information system of linear equations is built. Its solution provides the necessary coefficients to build a transformation matrix. These systems can be solved with the help of neural networks [4], as well as least squares method [5]. Paul Heckbert proposed analytical solution for this system by solving it’s particular case in general and then acquiring general solution by combining particular cases [6]. This method has one caveat: when building a ”birds-eye-view” three-dimensional space is required, whereas 3×3 matrix provides only 2-dimensional transformations. This is why LDWS mainly use 3×4 matrices [5]. The main idea is coordinate transformation and their further projection on an image surface ([7] and [8]). Main cons of this approach is relatively more demanding computations and lack of hardware solutions ([9] and [10]).

3. Proposed method

The main idea is to combine these two approaches. For efficiency 3×3 matrices will be used, and for construction parameters for 4×3 matrices will be used. Perspective transformations using matrices are described in (1) and (2).

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} * \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, x = \frac{x'}{z'}, y = \frac{y'}{z'} \tag{1}$$

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix}, x = \frac{x'}{z'}, y = \frac{y'}{z'} \tag{2}$$

In first case for a three-dimensional space one cannot calculate coefficients based on came intrinsic and extrinsic parameters, whereas in second it is possible. Matrix R is and extended rotation matrix along the axes with shifts (3).

$$R = \left[R_\alpha \; R_\beta \; R_\gamma | t \right] \tag{3}$$

To reduce one dimension Z projection is replaced with a constant $-h$, that symbolizes a distance from camera to the ground (4).

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ -h \\ 1 \end{pmatrix} = \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ 1 \end{pmatrix} \tag{4}$$

If we write it in a system, we get (5).

$$\begin{cases} X_c = r_{11} \cdot X_w + r_{12} \cdot Y_w - r_{13} \cdot h + t_x = a \cdot X_w + b \cdot Y_w + c \\ Y_c = r_{21} \cdot X_w + r_{22} \cdot Y_w - r_{23} \cdot h + t_y = d \cdot X_w + e \cdot Y_w + f \\ Z_c = r_{31} \cdot X_w + r_{32} \cdot Y_w - r_{33} \cdot h + t_z = g \cdot X_w + h \cdot Y_w + i \end{cases} \tag{5}$$

From it can be attained that:

$$\begin{cases} a = r_{11} \\ b = r_{12} \\ c = -r_{13} \cdot h + t_x \\ d = r_{21} \\ e = r_{22} \\ f = -r_{23} \cdot h + t_y \\ g = r_{31} \\ h = r_{32} \\ i = -r_{33} \cdot h + t_z \end{cases}$$

This concludes, that with fixed projection $Z = -h$ and 3×3 matrix can be acquired from an extended matrix R. To get parameter values for the resulting matrix we make an assumption, that camera yaw angle is relatively small and can be neglected for the case of LDW. So rotation matrix consists only of pitch and roll. Since ”birds-eye-view” suggests that camera is looking straight down, we provide additional 90−degree pitch rotation (6).

$$\begin{aligned} R_\beta \cdot R_\alpha \cdot R_{\alpha(90^\circ)} &= \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\frac{\pi}{2} & -\sin\frac{\pi}{2} \\ 0 & \sin\frac{\pi}{2} & \cos\frac{\pi}{2} \end{pmatrix} = \\ &= \begin{pmatrix} \cos\beta & \sin\beta \cdot \sin\alpha & \sin\beta \cdot \cos\alpha \\ \sin\beta & -\cos\beta \cdot \sin\alpha & -\cos\beta \cdot \cos\alpha \\ 0 & \cos\alpha & -\sin\alpha \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \end{aligned} \tag{6}$$

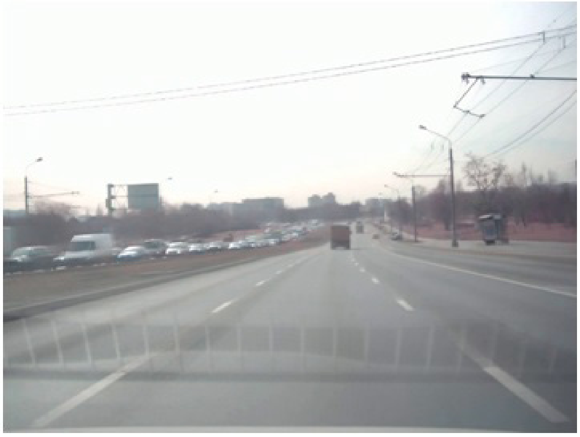

Fig. 1. Original image from a calibrated camera.

By replacing r_{ij} with new values we get:

$$\begin{cases} a = r_{11} = \cos\beta \\ b = r_{12} = \sin\beta\cdot\sin\alpha \\ c = -r_{13}\cdot h + t_x = -\sin\beta\cdot\sin\alpha\cdot h + t_x \\ d = r_{21} = \sin\beta \\ e = r_{22} = -\cos\beta\cdot\sin\alpha \\ f = -r_{23}\cdot h + t_y = \cos\beta\cdot\cos\alpha\cdot h + t_y \\ g = r_{31} = 0 \\ h = r_{32} = \cos\alpha \\ i = -r_{33}\cdot h + t_z = \sin\alpha\cdot h + t_z \end{cases}$$

To acquire the resulting matrix one must multiply focal distance matrix with a transformation matrix (7).

$$\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}\cdot\begin{pmatrix} \cos\beta & \sin\beta\cdot\sin\alpha & -\sin\beta\cdot\sin\alpha\cdot h + t_x \\ \sin\beta & -\cos\beta\cdot\sin\alpha & \cos\beta\cdot\cos\alpha\cdot h + t_y \\ 0 & \cos\alpha & \sin\alpha\cdot h + t_z \end{pmatrix} =$$

$$= \begin{pmatrix} f_x\cdot\cos\beta & f_x\cdot\sin\beta\cdot\sin\alpha + c_x\cdot\cos\alpha & f_x(-\sin\beta\cdot\sin\alpha\cdot h + t_x) + c_x\cdot(\sin\alpha\cdot h + t_z) \\ f_y\cdot\sin\beta & f_y\cdot(-\cos\beta)\cdot\sin\alpha + c_y\cdot\cos\alpha & f_y(\cos\beta\cdot\cos\alpha\cdot h + t_y) + c_y\cdot(\sin\alpha\cdot h + t_z) \\ 0 & \cos\alpha & \sin\alpha\cdot h + t_z \end{pmatrix} \quad (7)$$

This is a resulting perspective transformation matrix acquired wholly from cameras extrinsic and intrinsic parameters, which can be applied in hardware solutions such as Still IMage COProcessor (SIMCOP) on TDA platforms. The transformation was tested on a video from a calibrated camera (Fig. 1). With camera intrinsic and extrinsic parameters matrix was built and with help of OpenCV warp perspective function transformed video was constructed.

Camera parameter were:
f_x = 1633.3333333333335 focal length (in pixels);
$f_y = f_x$;
c_x = 640 - digital matrix horizontal center (in pixels);
c_y = 480 - digital matrix vertical center (in pixels);
pitch = −2.026380478113274 (degrees);
yaw = 2.2671202323992437 (degrees);

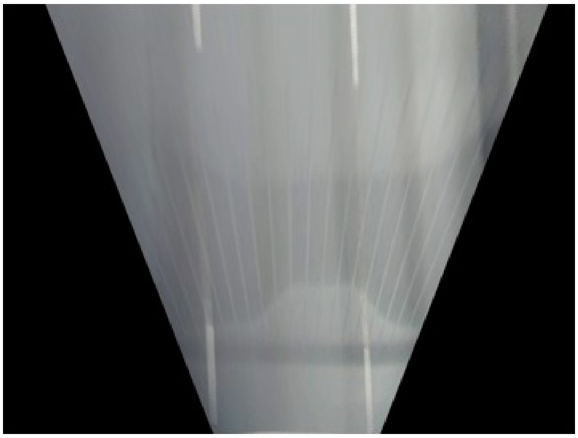

Fig. 2. Transformed imaged based on a 3 × 3 matrix.

roll = 0.5719324365448999 (degrees);
h = 1.1962422471370726 (meters);
t_x = 0.1595051172328819 (meters, shift along *X* axis);
t_y = 0 (shift along *Y* axis);
t_z = 0 (shift along *Z* axis);

Resulting image provided expected results (Fig. 2).

Conclusions

In this paper, we provided a formula for construction of a two-dimensional perspective transformation based on a three-dimensional but with a fixed *Z* projection. This method allows calculating transformation even on very small energy-efficient devices and furthermore the resulting matrix can be utilized with hardware solutions for perspective image transformations. Due to lack of extra surface projection step in a three-dimensional case the speed of an image construction increased by a multiple of two.

Acknowledgements

This work was supported by Competitiveness Growth Program of the Federal Autonomous Educational Institution of Higher Professional Education National Research Nuclear University MEPhI (Moscow Engineering Physics Institute).

References

[1] Liu Yu-Chih, Lin Kai-Ying and Chen Yong-Sheng. (2008) "Birds-Eye View Vision System for Vehicle Surrounding Monitoring." **4931** 207–218, doi:10.1007/978-3-540-78157-8_16.
[2] Sato Takaaki, Alessandro Moro, A. Sugahara, T. Tasaki, A. Yamashita and H. Asama. (2013) "Spatio-temporal bird's-eye view images using multiple fish-eye cameras." *Proceedings of the 2013 IEEE/SICE International Symposium on System Integration* Kobe, Japan: 753–758, doi:10.1109/SII.2013.6776674.
[3] A. V. Vinuchandran and R. Shanmughasundaram. (2017) "A real-time lane departure warning and vehicle detection system using monoscopic camera," *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, Kerala, India, **2018-January**: 1565–1569, doi:10.1109/ICICICT1.2017.8342803.
[4] Bruls Tom, Porav Horia, Kunze Lars and Newman Paul. (2018) "The Right (Angled) Perspective: Improving the Understanding of Road Scenes using Boosted Inverse Perspective Mapping." *IEEE Intelligent Vehicles Symposium (IV)* Paris, France, 302–309, doi:10.1109/IVS.2019.8814056.

[5] Kriegman David. (2007) "Computer Vision I." *CSE 252A, Winter 2007 Homography Estimation*.

[6] Heckbert Paul. (1995) "Projective Mappings for Image Warping."

[7] Jeong Jinyong and Kim Ayoung. (2016) "Adaptive Inverse Perspective Mapping for lane map generation with SLAM." *13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*: 38–41, urldoi: 10.1109/URAI.2016.7734016.

[8] Oliveira Miguel, Santos Vitor and Sappa Angel. (2014) "Multimodal Inverse Perspective Mapping." *Information Fusion* urldoi:10.1016/j.inffus.2014.09.003.

[9] Rybina Galina and Fontalina Elena. (2019) "Automated Acquisition, Representation and Processing of Temporal Knowledge in Dynamic Intelligent Systems." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, doi:10.1007/978-3-030-33274-7_9.

[10] Zamanov Ainur, Timofeev Yuriy, Alexeev Maxim and Mazurenko Grigoriy. (2018) "Building a three-dimensional picture of space based on the video of the system of stereo pairs." *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, Moscow and St. Petersburg, Russia, 397–399, doi:10.1109/EIConRus.2018.8317116