The Design of Vehicle Surround View Monitor

Antonia Juskova and Ondrej Kovac

Technical University of Kosice, Letna 9, 04200 Kosice, Slovakia
e-mail addresses: ondrej.kovac@tuke.sk, antonia.juskova@student.tuke.sk

***Abstract* – In this paper, the technology and design of a driving assistant called the Surround View Monitor (SVM) are discussed. The procedures used to create a unified, bird's-eye view are presented, which involve stitching images captured by strategically placed cameras on the vehicle. The proposed algorithm was successfully implemented on a car model and involves various stages, including camera calibration, photometric and perspective correction, scale planar transformation, and final image stitching. The paper should be considered student work and it is positioned as the foundation for the authors' future scientific work in the field of automated parking systems.**

I. INTRODUCTION

Sensing and communication between the environment and the vehicle are key challenges in the automotive industry [1]. The constantly increasing number of vehicles in traffic cause more emphasis on traffic safety and driving comfort [2]. Today's vehicles consist of newer and more sophisticated driving assistants referred to as Advanced Drivers Assistance Systems (ADAS). ADAS use advanced knowledge about the surrounding environment obtained by radars, lidars, cameras, or map databases [1]. Adaptive cruise control, driver drowsiness detection, and park assists are among many driver assistance systems which are frequently used in current vehicles. Advanced camera systems, such as SVM, are considered ADAS as well [2]. The surround view systems market was valued at 3.8 billion and probably reach 9.9 billion USD by 2027 [3]. This shows that development in this area of the automotive industry is economically interesting. SVM systems offer a 360° overview of the entire surrounding of the vehicle from a bird's eye perspective [2]. The first SVM was introduced by Nissan car division Infinity and was presented by the name Around View monitor [4]. Later, other manufacturers also started developing this technology. For example, Toyota introduced this system as Bird's View Camera or Chevrolet as Surround View [4]. SVM uses four wide-angle cameras strategically placed on the vehicle [2]. These cameras catch the entire surrounding of the vehicle. The control unit performs all the operations needed to transform images and stitches them into a unified, bird's eye view image.

II. CAMERA CALIBRATION

For capturing as much of the vehicle's surrounding as possible 160° wide-angle cameras are used. Wide-angle or fish-eye cameras are very popular and widely used types of cameras in robotic and computer vision [5]. These cameras provide wide fields of view, which allows for mapping a greater field of the 3D visual scene. Those wide-angle camera lenses produce significant visual distortion [6]. Approximation by a standard camera model may not be sufficient for those cameras. Using this model for wide-angle cameras may cause significant information loss in calibrated images. The lenses of these cameras are designed to capture a semi-spherical field of the 3D area. This view cannot be projected into the image plane by perspective correction. There are several methods of calibrating omnidirectional cameras. The crucial part of calibration is choosing a suitable camera model [5]. Most models of camera projection are estimated by the known geometry of captured objects or patterns, such as straight lines, angles, or planes**Chyba! Nenalezen zdroj odkazů.**. The projection of any camera can be approximated by an omnidirectional camera model [7]. The omnidirectional camera model is defined by two planes: the sensor plane and the image plane. The point from the 3D visual scene is projected into the sensor plane as a point $\boldsymbol{u}''$ defined by coordinates u'' and v''. The relationship between the 3D world and sensor plane can be mathematically described by imaging function $\boldsymbol{g}$. This function can have various forms and depends on lens type or construction. Projection between point $\boldsymbol{X}$ and point $\boldsymbol{u}''$ is defined by function $\boldsymbol{g}$ as:

$$\boldsymbol{g}(u'', v'') = (u'', v'', f(u'', v''))^T \quad (1)$$

$$f(u'', v'') = a_0 + a_1\rho'' + \cdots + a_N\rho''^N \quad (2)$$

Where $\rho'' = \sqrt{u''^2 + v''^2}$.

After camera projection and digitalization, point $\boldsymbol{u}''$ is mapped into the image plane as a point $\boldsymbol{u}'$=$(u,' v')^T$. The relationship between these two points in the sensor and image plane is defined as:

$$\boldsymbol{u}'' = \boldsymbol{A}\boldsymbol{u}' + \boldsymbol{t} \quad (3)$$

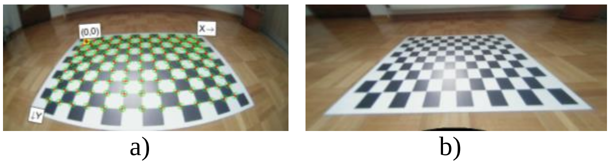
where $\boldsymbol{A}$ is a 2×2 matrix, **t** is a 2×1 vector.

By combining these two relationships, the projection of an omnidirectional camera model can be expressed:

$$\lambda \begin{bmatrix} u'' \\ v'' \\ w'' \end{bmatrix} = \lambda\, \boldsymbol{g}(\boldsymbol{u}'') = \lambda \begin{bmatrix} \boldsymbol{A}\boldsymbol{u}' + \boldsymbol{t} \\ f(u'', v'') \end{bmatrix} = \boldsymbol{P}.\boldsymbol{X} \qquad (4)$$

where λ is a non-zero scalar factor.

The goal of omnidirectional camera calibration is an estimation of $\boldsymbol{g}$ function parameters $a_0 \dots a_N$, matrix $\boldsymbol{A}$, and vector $\boldsymbol{t}$. During the calibration process, images of the checkerboard pattern are captured from several different positions. The checkerboard pattern has a known geometry and it's widely used in calibration techniques.



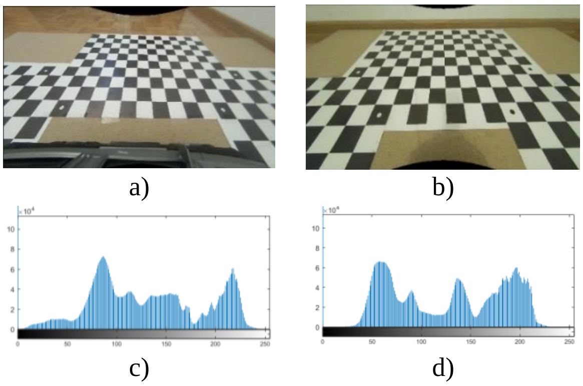
a) b)

Figure 1 Calibration process (a) distorted image and (b) calibrated undistorted image.

The result of calibration is shown in Figure 1b. As is shown, the camera distortion is partially eliminated. Changes are obvious mainly in displaying straight lines, which are not curves anymore.

III. PHOTOMETRIC ALIGNMENT

The brightness and color intensity of the same objects captured by several cameras may be in each picture different. This is caused due to different characteristics of cameras and lighting. Each of these cameras performs automatic exposure (AE) and automatic white balance (AWB) independently [8]. The mismatch of brightness and color intensities can cause noticeable boundaries mainly between nearby images. This mismatch can be reduced by the histogram equalization method.



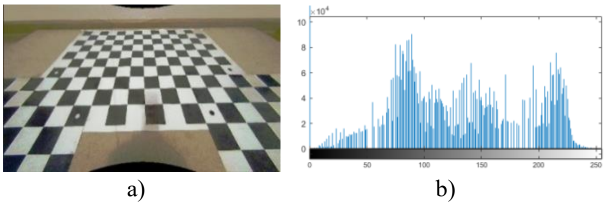
a) b) c) d)

Figure 2 Differences of brightness intensity (a) reference, (b) input image (c) reference, and (d) input histogram.

Histogram equalization is a method that transforms brightness intensity on each level according to the desired brightness distribution. This method creates a map based on a histogram of the reference image. Histograms of input images are changed based on the map derived from the reference image histogram [9]. This method redistributes pixels on each brightness level into new adapted values. Redistribution causes the input histogram's brightness values to be as close as possible to the values in the reference histogram [10]. The map function for histogram equalization can be derived from the distribution functions of an input image $\overline{DF}_{if}$ and reference image $\overline{DF}_{rf}$. Mapping aims to find for each intensity value i, a new intensity value j so that the difference between $\overline{DF}_{if}(i)$ and $\overline{DF}_{rf}(j)$ is minimal [11]. This can be described as follows:

$$\left|\overline{DF}_{zf}(i) - \overline{DF}_{rf}(j)\right| = min\left|\overline{DF}_{zf}(i) - \overline{DF}_{rf}(k)\right| \quad (5)$$

Based on minimalization, it is possible to create a mapping matrix M_{zf}. This matrix describes the mapping of pixels with intensity i to new intensity j as $M_{zf}(i) = j$. This method can be used for derivating the mapping matrix for each input image and each color depth. The result of this operation is a new normalized image with different colors and brightness intensity (Figure 4.). The probability of pixels occurrence with a certain brightness is closer to the probability of occurrence obtained from the reference image [10].



a) b)

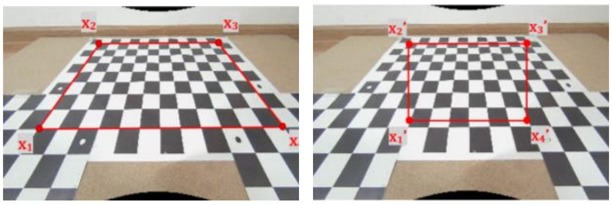
Figure 3 Result of histogram equalization, the normalized (a) image, and (b) histogram.

IV. PERSPECTIVE CORRECTION

Perspective correction is an important image-processing task in computer vision. This operation can adjust distortion created by shooting from different angles and distances [12]. Perspective correction is image transformation from one plane view to another plane view which is desired by the application. In surround view, all captured images need to be transformed into the front-parallel plane from the view of the virtual camera placed above the vehicle. One of the options for performing such a transformation is a homogenous perspective transformation. Homogeneous perspective transformation is an image transformation technique, where points of the input plane are transformed into new points in the output plane [13]. This transformation is defined by the transformation matrix $\boldsymbol{H}$ which transforms point $\boldsymbol{x}$ in input plane to coresponded point $\boldsymbol{x}'$ in output plane. The homography matrix $\boldsymbol{H}$ has 9 entries, but is defined only up to scale [14]. Homography between two corresponding points can be defined by (6).

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{6}$$

where $(x_1, x_2, x_3)^T$ are coordinates of a homogenous point in the input plane, $(x_1', x_2', x_3')^T$ are coordinates of a homogenous point in the output plane, and h_{11}... h_{32} are degrees of freedom of the matrix $\boldsymbol{H}$. As stated in the equation, it is necessary to calculate 8 degrees of freedom to estimate homography. Direct linear transformation (DLT) can be used to calculate unknown entries. For estimation with DLT, it is a necessity to choose four corresponding points in both planes.



a) b)

Figure 4 Coresponding points for DLT, (a) input plane, (b) output plane.

For each point correspondence $x_i \leftrightarrow x_i'$ is necessary to compute the matrix $\boldsymbol{A_i}$:

$$\boldsymbol{A_i} = \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i x_i' & -y_i x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_i y_i' & -y_i y_i' \end{bmatrix} \tag{7}$$

From each point correspondence, two equations are obtained. With four 2×9 matrices $\boldsymbol{A_i}$ a single 8×9 matrix can be assembled. The following equation (8) represents DLT using point correspondence. The matrix $\boldsymbol{H}$ is vectorized into $\boldsymbol{h}$ as follows.

$$\boldsymbol{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})^T \tag{8}$$

$$\boldsymbol{A}.\boldsymbol{h} = \begin{bmatrix} \boldsymbol{A_1} \\ \boldsymbol{A_2} \\ \boldsymbol{A_3} \\ \boldsymbol{A_4} \end{bmatrix} \boldsymbol{h} = 0 \tag{9}$$

The unknown entries of vector $\boldsymbol{h}$ can be calculated by the least squares solution (10).

$$\boldsymbol{h} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{b}\boldsymbol{A}^T \tag{10}$$

Where $\boldsymbol{b} = (x_1', y_1', x_2', y_2', x_3', y_3', x_4', y_4')^T$.

By this algorithm, it is possible to calculate the transformation matrix of the homography $\boldsymbol{H}$ for all four images. By multiplying all points in the image with the matrix $\boldsymbol{H}$, these are transformed into equivalent points in the output image.

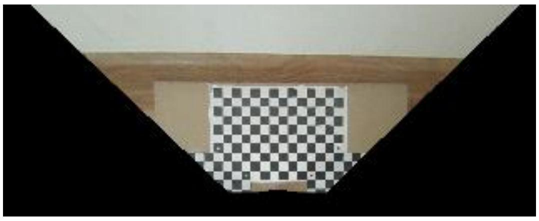


Figure 5 Input image after perspective correction

The transformed image (Figure 6.) represents the input image after perspective correction performed from the perspective of the front-parallel plane above the vehicle. Before stitching images into one unified image, scale transformation of images must be performed.

V. SCALE TRANSFORMATION

After perspective correction, it may happen that The size of the same objects captured by the 4 cameras may not be the same and problems may arise, especially during final image stitching. The goal of planar scale transformation is to match the scale of the images. This transformation is defined by scale factors s_x and s_y. For scale alignment of four views, a checkerboard pattern can be used (Figure 7). From each image, the width and height of the checkerboard pattern can be obtained from corner points. Based on these values, it is possible to calculate the average height and width. These average values represent the unified dimensions of the checkerboard pattern in each image. Thus, the scale matrix based on the checkerboard can be rewritten as follow:

$$S = \begin{bmatrix} E_x/(x_3 - x_2) & 0 & 0 \\ 0 & E_y/(y_2 - y_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Where E_x is mean width and E_y is mean height.

The matrix S is subsequently used for the scale unification of all 4 images. The scaling factors for a particular image can be calculated from the mean values and checkerboard dimensions in the actual image.

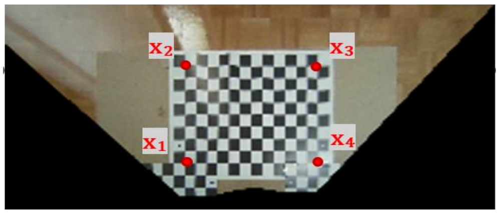


Figure 6 Checkerboard corner points

VI. STITCHING IMAGES

Finding the most accurate match between images is a very important part of image stitching. Defining common points between two adjacent images is a necessity. These common points indicate how joining should be performed. One of the image stitching methods is an algorithm based on imaginary diagonals that define where adjacent views should intersect. In this algorithm, only two points in each adjacent image must be entered. These common points define the imaginary diagonals, which represent the same point within adjacent frames. This is shown in Figure 8.

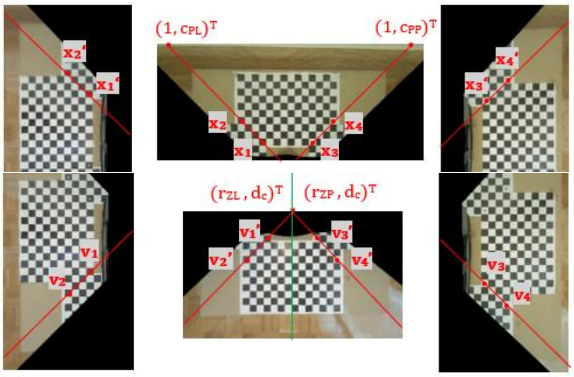


Figure 7 Common points for image stitching

Finally, all images are mapped into one image. The view of the vehicle's front camera can be chosen as the main image. This image also represents the resulting plane of the surrounding view so the resolution must be increased. The resolution must be large enough to map all the remaining images onto it. The remaining images need to be rotated and moved to the specified coordinates. Before joining, the two corresponding common points must lie approximately on the same coordinates. Based on these points, it is necessary to calculate the incremental step (IS). IS is a value that represents how many pixels need to be added/subtracted on each row of the image. The result of the joint appears as a straight line passing through the common points (Figure 8).

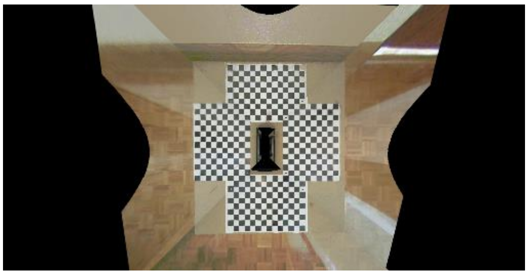
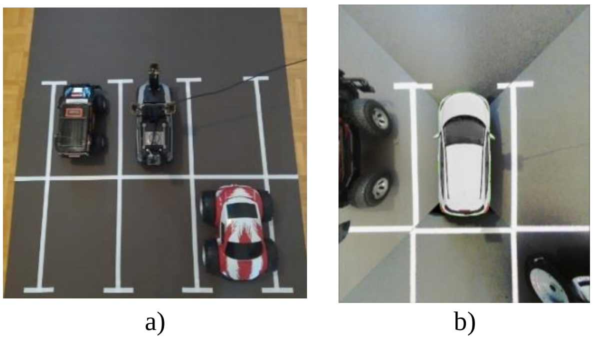


Figure 8 Unified joined image

Next, it is necessary to calculate where the given incremental step should be added/subtracted. In the case of connecting the front, right and left images, it is necessary to calculate the coordinates of the columns c_{PP} and c_{PL}. These coordinates define in which column the pixels from one adjacent image should be added and where from the other (Figure 7). In the case of stitching the rearview, it is necessary to calculate the lines r_{ZP} and r_{ZL}. These lines define where the rear image should be mapped.



a) b)

Figure 9 Estimation of the vehicle surrounding, (a) actual scenery, (b) estimated scenery.

The result of this process is a unified view from a bird's perspective. This view still needs to be cropped to the immediate surroundings of the vehicle. For a better understanding of the perspective, the image of a car can be placed in the middle of the image.

VII. PRACTICAL IMPLEMENTATION

For demonstration purposes, the prototype of a surround-view system was proposed. The software was implemented in MATLAB and partially in Python. The images are acquired by a Raspberry Pi via Python script which captures the images with the highest possible resolution. The images are sent to the MATLAB environment via the local network. Image processing and transformation functions are performed in MATLAB.

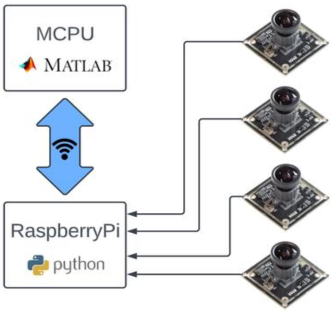


Figure 10 Surround-view system

The prototype uses four wide-angle cameras B0202 with a 160° field of view. The cameras are connected to Raspberry Pi's USB inputs, which are used for image acquisition. For image processing, some functions of the MATLAB Image Processing Toolbox are used, such as checkerboard pattern recognition, fish-eye camera parameters estimation, and 2D image transformation with a specified matrix.

VIII. FUTURE WORK

The final view obtained by the proposed system can be used as a zero-level driver assistant. Based on a bird's eye view it is possible to evaluate the environment around the vehicle. Surround-view cameras can be also used as a sensor system for higher-level ADAS such as park assist or as a redundant sensor in autonomous vehicles. The image can be simply converted into an occupancy binary map, which indicates free or occupied space. Based on the binary values it is possible to track objects in the surroundings of the vehicle and use it for measuring distance from obstacles in parking warning systems. This representation of the environment can be also used for parking space measurement or trajectory estimation in informative parking assistance. Provided that there is suitable lighting, these functions can also be used for semi-automatic and automatic parking systems.

IX. CONCLUSION

This article details the process of implementing a vehicle surround view monitoring system, also known as the bird's eye view. The proposed system utilizes output from four cameras mounted on all sides of a vehicle and includes camera calibration, histogram equalization, perspective projection, and image stitching. In the system calibration process, an automatic checkerboard pattern recognition function is used. By automating these steps, it is possible to eliminate errors and inaccuracies in the manual marking of corner points on the checkerboard pattern by the user. The proposed algorithm was successfully implemented on a car model. The bird's eye view generated by this system can be utilized as a sensor system for parking assistance, a critical aspect of which involves estimating obstacles and maneuvering between them. Existing parking systems rely solely on ultrasound or lidar sensors, which may provide incomplete information about a vehicle's surroundings, increasing the risk of collision. The proposed bird's eye view system is positioned as the foundation for the authors' future work in the field of automated parking systems.

ACKNOWLEDGMENT

The work is a part of the project supported by the Science Grant Agency of the Slovak Republic (No. 1/0413/22).

REFERENCES

[1] Galvani, M. "History and future of driver assistance." 2019 IEEE Instrumentation Measurement Magazine, 22(1), str. 11–16, doi:10.1109/mim.2019.8633345.

[2] X. Yan, L. Wang,Y. Li, "A Calibration Method for the Surround-View Parking Assistant System," 2020 IEEE 11th ICSESS, Beijing, China, 2020, pp. 485-488, doi: 10.1109/ICSESS49938.2020.9237687.

[3] Imarc. (2022) Automotive Surround View Systems Market (Report ID: SR112023A1650). Retrieved from Imarc database.

[4] Howard, Bill. "Why are car surround view cameras and why are they better than they need to be," Extremetech, 18-07-2014, [cit. 25-03-22]. Available at: <https://www.extremetech.com/extreme/186160-what-are-surround-view-cameras-and-why-are-they-better-than-they-need-to-be...>.

[5] Urban, Steffen. Leitloff Jens. "Improved Wide-Angle, FishEye and Omnidirectional Camera Calibration," ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 108, 2015, pp. 72-79, ISSN 0924-2716, DOI: 10.1016/j.isprsjprs.2015.06.005.

[6] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 8, pp. 1335-1340, Aug. 2006, doi: 10.1109/TPAMI.2006.153.

[7] Scaramuzza, D., A. Martinelli, and R. Siegwart. "A Toolbox for Easy Calibrating Omnidirectional Cameras." Proceedings to IEEE International Conference on Intelligent Robots and Systems, (IROS). Beijing, China, October 7–15, 2006.

[8] K. Lee, D. Kim, H. Do, J. Kim, and K. Chae, "Effective Photometric Alignment for Surround View Monitoring System," *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, Berlin, Germany, 2019, pp. 390-391, doi: 10.1109/ICCE-Berlin47944.2019.8966185.

[9] Komal, Vij & Yaduvir, Singh. (2011). Enhancement of Images using Histogram Processing Techniques. International Journal of Computer Technology and Applications. 02.

[10] Grundland, M., Dodgson, N. Color histogram specification by histogram warping. Proceedings of SPIE 5667, 2005 s. 610-621. DOI: 10.1117/12.596953

[11] Chen, Ch. Ishikawa, H. Histogram Matching Extends Acceptable Signal Strenght Range on Optical Coherence Tomography Images. Invest Ophthalmol Vis Sci. 2015;56:3810–3819. DOI:10.1167/iovs.15-16502.

[12] Li, Zeying, et al. "Resource-Efficient Hardware Implementation of Perspective Transformation Based on Central Projection." *Electronics* 11.9 (2022): 1367.

[13] HARTLEY, Richard; ZISSERMAN, Andrew. 2004. Multiple View Geometry in Computer Vision, ver. 3. Cambridge: Cambridge University Press. ISBN 978-0-511-811-685.

[14] Agarwal, Anubhav; Jawahar, C.; Narayanan, P. A survey of Planar Homography Estimation Techniques. International Institute of Information Technology, Hyderabad, India, 2005.