

doi:10.15199/48.2020.04.35

# Modern UVC stereovision camera's calibration and disparity maps generation: mathematical basis, algorithms and implementations

**Abstract.** This paper presents a complete and up-to-date introduction to problems of stereo camera calibration with crucial mathematical models, description of calibration procedure and proposition of evaluation of overall calibration quality. This last step is our original contribution to this subject. To practically demonstrate how the calibration procedure works, we have performed calibration of USB / UVC camera module that can be reproduced and nearly directly applied to other hardware of this type. Our visual data and source codes can be downloaded from digital repository. The subject of this article might be especially interesting to employees of research units, students, doctoral students and electrical engineers who want to get familiar with up-to-date stereovision calibration and processing procedures and who want to perform those actions on their own on popular and available hardware.

**Streszczenie.** W tym artykule przedstawiono pełne i aktualne wprowadzenie do problematyki kalibracji kamery stereo z uwzględnieniem kluczowych modeli matematycznych, opisu procedury kalibracji i propozycji ogólnej oceny jakości kalibracji. Ostatni krok to nasz oryginalny wkład w ten temat. Aby praktycznie przedstawić proces kalibracji, przeprowadziliśmy kalibrację modelu kamery USB / UVC, który można powielić i niemal bezpośrednio zastosować w sprzęcie podobnego typu. Zarówno dane jak i kody źródłowe które użyliśmy przy tworzeniu niniejszego artykułu dostępne są w repozytorium online. Temat poruszany w artykule może być szczególnie interesujący dla pracowników jednostek badawczych, studentów, doktorantów i inżynierów, którzy chcą zapoznać się z aktualnymi procedurami kalibracji i przetwarzania stereowizji oraz którzy chcieliby wykorzystać te mechanizmy na popularnym i dostępnym sprzęcie. (Kalibracja i generowanie map rozbieżności w nowoczesnej kamerze stereowizyjnej UVC: podstawy matematyczne, algorytmy i implementacja)

**Keywords:** Stereovision, UVC camera, speckle filter, disparity map.

**Słowa kluczowe:** Stereowizja, kamera UVC, filtr "speckle", mapa rozbieżności.

## 1. Introduction

Stereovision camera calibration is a laborious process that requires both technical and scientific knowledge in order to perform it correctly. Nowadays we can easily purchase relatively cheap stereovision hardware that have image quality acceptable for the purpose of robotic vision. Low-power consumption cameras with USB interfaces can be easily integrated with popular microcontrollers and microcomputers and serve as a visual module of the robotic system. We can apply this type of visual data for example to binocular odometry [17][18], robotic mapping [19][20] or virtual /augmented reality [21]. The trade off the relatively low price of this type of double-camera circuits is minimal software support for those devices. Quite often manufacturer assures only USB 2.0 / 3.0 and UVC support of compressed video streams. The stereo camera calibration process that is essential for any practical application of acquired data relies on the camera owner.

### 1.1 State – of –the –art in stereovision calibration software

There is a number of free to download software that can be used for camera calibration. Some applications are available as executable files, the others are distributed in the form of programming libraries. Among most popular programs and libraries that can be utilized are Stereo Camera Calibration App available in Matlab / Simulink, BoofCV library available in Java<sup>1</sup>[16] or for example OpenCV library<sup>2</sup> written in C++ but with bindings to many other languages for example JAVA, Python or Matlab. In this paper we used the increasingly popular programming language Python (version 3.6) with OpenCV 3.4. We have chosen this programming language because it is among most popular solution that is used by people specialized in image processing and robotics. Python provides memory management that significantly speeds up programming and

it can be easily redistributed and run on various hardware and system architectures. With it is possible to call fast low-level libraries implemented C++ (like OpenCV). Also Python runtime is available in operating systems of many popular microcomputers (for example Raspberry PI), that are often components of robotic visual systems.

### 1.2 Motivation of this paper

As can be seen in previous section, there are many software packages that can be used for stereo camera calibration and real time image processing in order to obtain an environment disparity maps; however calibration process itself is not an easy task and it has many pitfalls, that are not mentioned in the software documentation. Also as we already mentioned, it is crucial to understand the mathematic model that stands behind the calibration and further image processing. To our surprises we did not find complete up-to-date survey on this subject that summarizes both mathematical and technical aspects of those procedures. That fact was our motivation to write this paper. This paper presents a complete and up-to-date introduction to problems of stereo camera calibration with crucial mathematical models, description of calibration procedure and proposition of evaluation of overall calibration quality. This last step is our original contribution to this subject. To practically demonstrate how the calibration procedure works, we have performed calibration of USB / UVC camera module that can be reproduced and nearly directly applied to other hardware of this type. Our visual data and source codes can be downloaded from digital repository [22]. The subject of this article might be especially interesting to employees of research units, students, doctoral students and electrical engineers who want to get familiar with up-to-date stereovision calibration and processing procedures and who want to perform those actions on their own on popular and available hardware.

<sup>1</sup> <https://boofcv.org>

<sup>2</sup> <https://opencv.org/>

## 2 Camera calibration

In this section we will discuss mathematical basis of the stereovision camera calibration procedure.

By the term stereo camera calibration we mean estimating all required parameters (internal and external) that have to be known to minimize distortion of leans and to enable reliable binocular distance measurements by applying so called disparity map.

Disparity map is a two-dimensional array, that contains information about distances between corresponding pixels on stereo images. Disparity map is often represented as real-value matrix, where each value is proportional to that distance. The disparity map can be also used to generate depth map. Depth map a real-value matrix, where each value corresponds to distance between the viewport (camera leans) and scene's surface.

### 2.1 Acquisition of calibration images pairs

Before the calibration procedure begins, we need to prepare a chessboard with white and black square fields which sizes are known. Then we need to take several dozens of pictures pairs (with left and right camera simultaneously). In each picture the board should face camera at different angle and be positioned in various places of the scene – see Fig. 1.



Fig 1. This figure presents example pairs of pictures with the chessboard that we used for camera calibration. The complete set of calibration pictures is available in digital repository [22].

An example chessboards are presented in Fig. 2. As can be seen it is recommended to use different number of rows and columns in chessboard, for example 6x5, 8x7, 10x9, etc.

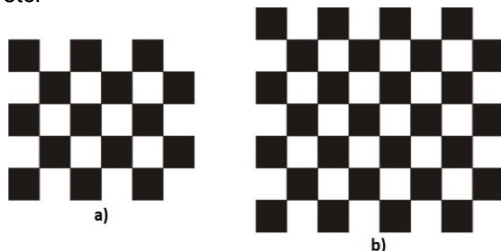


Fig. 2. sample pictures of chessboard a) 6x5 b) 8x7 used in this work

That is because if those numbers are equal, chessboard segmentation algorithm might not correctly estimate chessboard rotation angle around normal axis to chessboard surface. During stereo system calibration we discuss in this paper we used the chessboard with size 7x8.

After preparing a set of calibration pictures we need to undergo several crucial steps. All of them can be done by appropriate algorithm implemented as a computer program:

- Determining chessboard positions in all pictures – the stage at which chessboard corners are detected. Chessboard images will be distorted by a camera leans. Knowing the number of rows and columns in chessboard and the fact, that all fields on chessboard are square we will use this information later on.
- Camera parameters determination – the stage in which we determine the internal and external parameters of camera.
- Fundament and Essential matrix calculation – the part in which the relationship between pixels from both images will be defined.
- Disparity map creation – the last step in which image pair is used to create disparity map and eventually depth map.

### 2.2 Determining chessboard positions in all pictures

After preparing chessboard images the next step is estimating chessboard position by finding its corners. Algorithm that is used to look for corners should be resistant to low quality of image, presence of noises. Most algorithms of this type work on the basis of a simple rule: the corner have a strong edge in the main direction and another edge normal to the first one [14]. Generally algorithms that use the above rule take the first and the second derivatives of the images in the directions  $(u, v)$  to find corners. In this paper we used method developed by **Chris Harris** and **Mike Stephens** in [9], which is so called Harris Corner detector [14]. The basis for the Harris and Stephens method is a calculation based on the first derivative of the picture function  $I(u, v)$  in the horizontal and vertical directions.

$$(2) \quad I_x(u, v) = \frac{\partial I}{\partial x}(u, v) \quad \text{and} \quad I_y(u, v) = \frac{\partial I}{\partial y}(u, v)$$

Let us denote:

$$(2) \quad A = A(u, v) = I_x^2(u, v)$$

$$(2) \quad B = B(u, v) = I_y^2(u, v)$$

$$(4) \quad C = C(u, v) = I_x^2(u, v) * I_x^2(u, v)$$

Then a smooth Gaussian  $H^{G, \sigma}$  liner filter is applied and the  $M$  matrix is created

$$(5) \quad M = \begin{bmatrix} A * H^{G, \sigma} & C * H^{G, \sigma} \\ C * H^{G, \sigma} & B * H^{G, \sigma} \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{C} \\ \bar{C} & \bar{B} \end{bmatrix}$$

Since the matrix  $M$  is symmetrical, we can diagonalized it:

$$(6) \quad \bar{M} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

where  $\lambda_1$  and  $\lambda_2$  are eigenvalues for the matrix.  $\bar{M}$  The calculation of eigenvalues can be done with following equation:

$$(7) \quad \lambda_{1,2} = \frac{\text{trace}(M)}{2} \pm \sqrt{\left(\frac{\text{trace}(M)}{2}\right)^2 - \det(M)} =$$

$$\frac{1}{2} \left( \bar{A} + \bar{B} \pm \sqrt{\bar{A}^2 + 2\bar{A}\bar{B} + \bar{B} + 4\bar{C}^2} \right)$$

where trace is a sum of elements on the main diagonal. If in the given area of an image  $(u, v)$  the values of  $M = 0$  and  $\lambda_1 = \lambda_2 = 0$  then the examined area is "flat", there are no

corners. In other cases image at this point contains some corners. The strength of the corner is proportional to the eigenvalue. Both eigenvectors are perpendicular, so the second eigenvalue is proportional to the strength of the second, "less visible" edge. Since  $|\lambda_1| \geq |\lambda_2|$  we can use the eigenvalue as the threshold. In the corner the difference between first and second eigenvalue should be small (because it is the point where two "strength" edges meet).

For this purpose, Harris define a function that helps in determining the potential corner.

$$(8) \quad Q(u, v) = \det(M) - a * (\text{trace}(M))^2 = (\overline{AB} - \overline{C}^2) - a * (\overline{A} + \overline{B})^2$$

The function  $Q(u, v)$  called „corner response function” and returns the maximum corner value. The  $a$  value is determining the accuracy of the function. The point  $(u, v)$  is a corner candidate if it meets the condition.

$$(9) \quad Q(u, v) > t_H$$

(8) is applied for the whole image, and all points the meets (9) are added to the list.

$$(10) \quad \text{Corners} = [c_1, c_2, \dots, c_n]$$

where  $c_i = (u_i, v_i, q_i)$  and  $q_i = Q(u, v)$ . the list is sorted in descending order by  $q_i$  value.

### 2.3 Camera parameters determination

After we have obtained all coordinates of the chessboard corners the internal and external parameters of the camera can be calculated [1][2]. The external (or extrinsic) parameters are considered information about camera rotation and translation [13] that are represented by a matrix  $[R|T]$ . Using these parameters one can describe the position of camera in scene [15]. Suppose a point in 3D space is represented as  $P = [X, Y, Z]^T$  and a point in 2D space as  $p = [u, v]^T$ . Let's use  $\hat{x}$  to denote the vector to add 1 to the last element, so we get:  $\hat{p} = [X, Y, Z, 1]^T$  and  $\hat{p} = [u, v, 1]^T$ . To present the camera parameters we will use the so-called pinhole-camera model in which the scene views created by projection a 3D point on the image plane using perspective transformation.

$$(11) \quad \hat{s}\hat{p} = W[R|T]\hat{P}$$

where:

$$(12) \quad W \equiv \begin{bmatrix} f_x & 0 & s_x \\ 0 & f_y & s_y \\ 0 & 0 & 1 \end{bmatrix}, R \equiv \begin{bmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{bmatrix}, T \equiv \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

The parameter  $\hat{s}$  is a scale factor attribute, Internal parameters that are presented as the  $W$  matrix store information about the focal length  $(f_x, f_y)$ , the focal length is the distance the centre of the lens to the focal points in the leans. The next parameters is „principal point”  $(s_x, s_y)$  this is the centre point in the scene (Fig. 3). Presenting a point 3D space in 2D plane without taking into account the distortion which is presented in Fig. 3. We obtain the formula:

$$(13) \quad u = f_x \frac{x}{z} + s_x$$

$$(14) \quad v = f_y * \frac{y}{z} + s_y$$

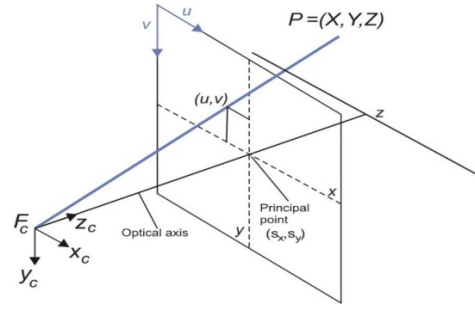


Fig. 3 This figure presents the mapping of the 3D point on the 2D plane.

In cameras used on an everyday basis we can notice some distortions, caused by the use of various types of lenses (e.g. wide-angle, "fish eye"). Distortions generated by such lenses can be divided into two groups.

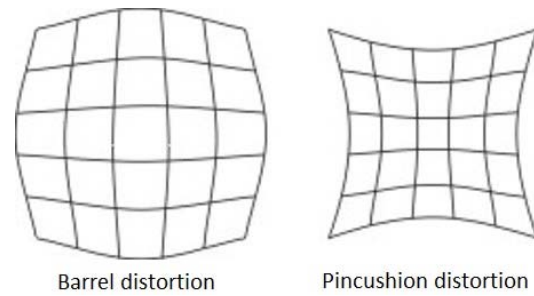


Fig. 4 This figure presents two type of distortion

**Barrel distortion** – it is a convex distortions in which the magnification decreases with distance from the optical axis.

**Pincushion distortion** – it is a concave distortions in which magnification increases with distance from the optical axis.

Above types of distortion can be modelled by using appropriate mathematical equations, for example:

Field of View [6]

$$(15) \quad (x_u, y_u) = \frac{\tan(r_d \omega)}{2r_d \tan\left(\frac{\omega}{2}\right)} (x_d, y_d)$$

Division model [7]

$$(16) \quad (x_u, y_u) = \frac{1}{(1 + kr_d^2)} (x_d, y_d)$$

Rational Function [8]

$$(17) \quad (x_u, y_u) = \left( \frac{W_1^T X(x, y)}{W_3^T X(x, y)} \right), \left( \frac{W_2^T X(x, y)}{W_4^T X(x, y)} \right)$$

Where  $X(x, y) = [x^2, xy, y^2, x, y, 1]^T$

Polynomial model

$$(18) \quad \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \end{bmatrix} \left[ 1 + r_d^2 k_1 + r_d^4 k_2 \right], r_d = \sqrt{x_d^2 + y_d^2}$$

In this paper the Polynomial model was used[4][5]. The value  $(x_d, y_d)$  are the coordinates of the distorted image, the value  $(x_u, y_u)$  are the coordinates of the undistorted image. Using the Maximum Likelihood Estimation function,

we can obtain parameters  $k_1$  and  $k_2$  with which the distortion is generated.

$$(19) \quad \sum_{i=1}^n \sum_{j=1}^m \left| p_{ij} - \tilde{p}(W, k_1, k_2, R_i, T_i, P_j) \right|^2$$

Where  $n$  is the number of model images,  $m$  is point on the model plane. Then  $\tilde{p}(W, k_1, k_2, R_i, T_i, P_j)$  is the projection of point  $P_j$  in image  $i$  according to equation (11). This is a nonlinear minimization problem which is solved with the Levenberg-Marquardt Algorithm[26].

#### 2.4 Fundament and Essential matrix calculation

The Fundamental matrix  $F$  is the algebraic representation of epipolar geometry. The Epipolar geometry[23] is the internal geometry of projecting a point between two views, it depends only on the internal parameters of the camera and the relative position. Suppose point  $P$  is in 3D space,  $p$  and  $p'$  are mapped points in the left and right views. The relationship between points and the centre of camera is that they lie on one plane (Fig. 5) which is called epipolar plane, has been marked as  $\pi$ .

The plane  $\pi$  intersects the image plane delineated by epipolar lines ( $e, e'$ ) on which the points  $p$  and  $p'$ . For each  $p$  point in the first image there is a corresponding epipolar line  $e'$  in the second image. Therefore, each point  $p'$  matching to point  $p$  must lie on line  $e'$ . This representation of points between the images is represented by the Fundamental matrix  $F$  and the matrix definition looks like this[16]:

$$(20) \quad p^T F p' = 0$$

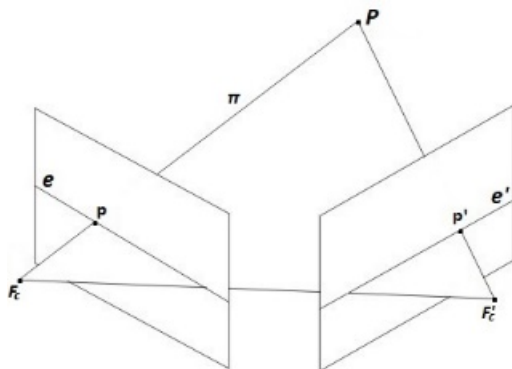


Fig. 5. This figure presents epipolar geometry.

Essential matrix  $E$  store information about the external parameters of the camera i.e. rotation and translation. The matrix definition looks like this[16]:

$$(21) \quad [R | T]_R^T E [R | T]_L = 0$$

#### 2.5 Disparity map creation

This section presents the operation of the SAD algorithm [10][11], which is used to generate a disparity map. The algorithm uses two images from the left and right camera, and then calculates the differences between the pixel intensity in both images[3]. If the calculation result is a low number, the compared regions are "matched".

To calculate the pixel intensity in the SAD algorithm, matching windows are used (Fig 6). The "reference" window is determined in the image from the left camera, and the "candidate" window is determined in the image from the right camera. Then the "candidate" window is moved over

the image, the pixel intensity is calculated and compared with the "reference" window. Using the SAD algorithm:

$$(22) \quad SAD(x, y) = \sum_{i=0}^n \sum_{j=0}^n \left| \hat{I}_L(x, y) - \hat{I}_R(x+i, y+j) \right|$$

where  $n$  is the size of the matching window,  $\hat{I}_R(x, y), \hat{I}_L(x, y)$  means the intensity of the pixels from the left and right image.

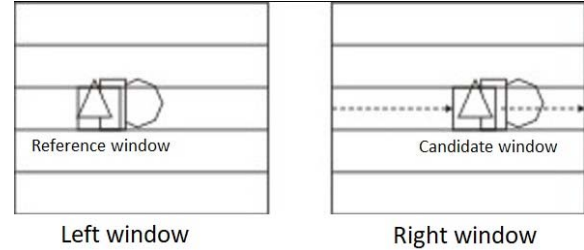


Fig. 6 This figure presents the operation of the SAD method.

The disparity map returned by the SAD algorithm contains some matching errors. This may be due to low image quality, noise, or simply wrong matching (unified background). We will use the "speckle" filter to correct the obtained errors. Filtration consists in checking the value of pixels which adjoin to the tested pixel. If the pixel values are too high, they are modified. The filter operation is presented by using pseudo code.

```

f [width][height] := 0
for i := 0 to height do
  for j := 0 to width do
    if pi [i][j] != nVal then
      if f [i][j] then
        if pi [i][j] <= nVal then
          pi [i][j] := nVal
        end
      else
        // location - checking if the pixel in the image (height, width)
        foreach neighbourhood do
          //look in square neighbourhood m x m of pixel [i,j]
          // n is neighbourhood
          if location and lf[n+1] and
             math.abs(n - n+1) <= minDiff then
             f [n+1] := 1 // visited
             n+1 := n
          end
          if location and lf[n-1] and
             math.abs(n - n-1) <= minDiff then
             f [n-1] := 1 // visited
             n-1 := n
          end
        end
      end
    end
  end
end
end
end

```

### 3 Optimization of parameters for disparity map generation

The previous section presents the algorithm by which the disparity map is generated. In this section we will check if the size of the matching window and the filtration window are important for generating of the map. The test was carried out on 640 x 480 pixel images, the window sizes were modified in the appropriate ranges: the matching window from 5 to 255 with a fixed step of 2 and the "Speckle" filtration window was modified in the range of 50 to 200 with a fixed step of 10. At each stage of the test, the result was a disparity map; pixels from each map with a



value greater than zero were counted for every map (with zero, no distance can be determined). The results of the performed tests are presented in Fig. 7.

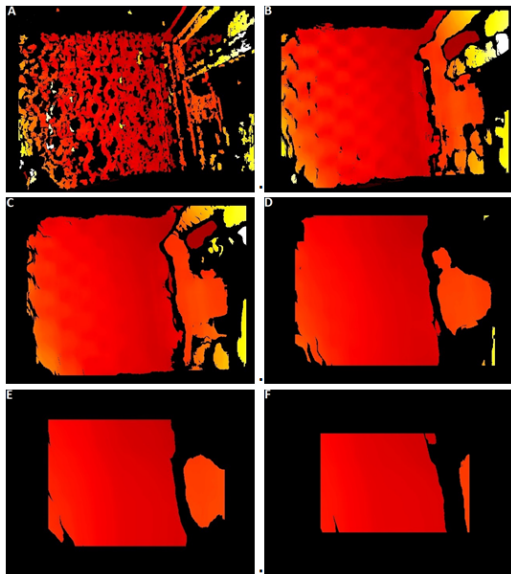


Fig. 7. the figure presents a disparity map after using different window size a) 7x7, b) 29x29 c) 69x69, d) 99x99, e) 151x151, f) 225x225.

The obtained results indicate that the use of an insufficiently big matching window (7x7) causes generation of "sparse" disparity map (with many "holes" – see Fig. 7a). There are many areas that are not connected with each other. Using a large size window (225x225, 255x255) also results in low non-zero disparity pixels count, however regions on the map are more consistent (see Fig. 7f). In the performed test, the largest number of pixels and also the highest map accuracy was obtained by using matching windows sized from 27x27 to 31x31 (see Fig. 7b).

The second parameter that we modified was the size of the "Speckle" filter window. This parameter can easily be tuned by using plot presented in Fig. 8. This filter has much importance in case when disparity map generation algorithm has introduced some errors. Those errors might happen when disparity generation algorithm has relatively small window. When this window has moderate size (in tested cases it was about 41x41) the filter does not have so much importance. This situation is illustrated in Figures 7A and 7C. In order to illustrate the "Speckle" filter behaviour we have examined it on disparity map generated with algorithm with matching window sized 5x5. With those settings after applying "Speckle" filter with window sized 50x50 we have obtained a smooth disparity map with 70000 pixels and with window sized 200x200 a smooth disparity map with 46000 pixels.

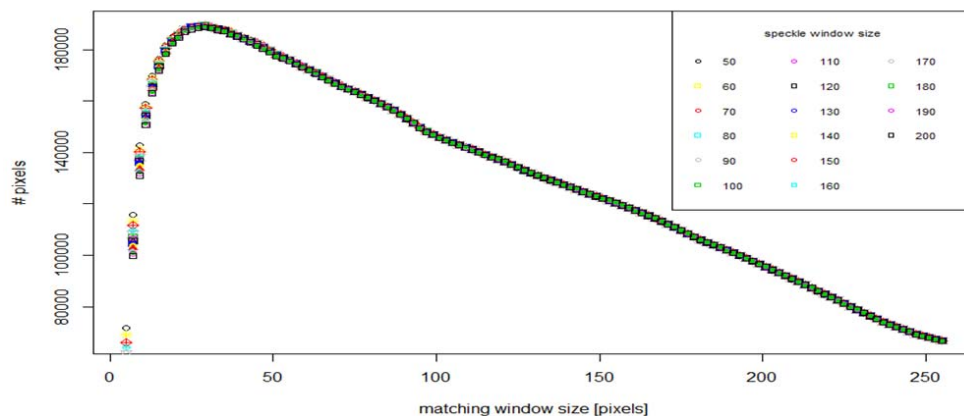


Fig. 8. This figure presents plot of dependency between "Speckle" filter window size and number of non-zero pixels on disparity map.

#### 4 Conclusion

This paper presents a mathematical model of camera calibration for stereo vision. In each of the presented stages we obtained individual camera parameters. A disparity map was obtained in the last calibration step. The SAD algorithm with which we obtained disparity map uses the matching window with adaptable size. Based on the presented results it can be observed that the selection of the right size of the matching window is very important for the final result of the disparity map; the use of too small or too big windows will cause inaccurate results. The "Speckle" filter helps remove incorrect matches from the map, smooths it out. Adjusting the right size of the filtration window matters too, because the use of large sizes may cause losing some of the pixels from the map, making the map inaccurate. The right size of the matching and filtration windows set for the surroundings is an important element of calibration.

**Authors:** dr hab. inż. Tomasz Hachaj, Institute of Computer Science, Pedagogical University of Cracow, Podchorążych 2, 30-084, Cracow, Poland, email: tomasz.hachaj@up.krakow.pl, mgr Patryk Mazurek, Institute of Computer Science, Pedagogical University of Cracow, Podchorążych 2, 30-084, Cracow, Poland, email: patryk.mazurek@up.krakow.pl

#### REFERENCES

- [1]. Z. Zhang. A Flexible "New Technique for Camera Calibration". IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.
- [2]. S. Dai, Q. Ji A "New Technique for Camera Self-Calibration". Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 09 July 2003
- [3]. H. Hirschmuller "Stereo Processing by Semi-Global Matching and Mutual Information", IEEE Transactions on Pattern Analysis and Machine Intelligence, 328 – 341, 18 December 2007
- [4]. Junhee Park, Seong-Chan Byun, Byung-Uk Lee, "Lens Distortion Correction Using Ideal Image Coordinates", IEEE Transactions on Consumer Electronics, 2009
- [5]. R. Tsai, A versatile "Camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV

- cameras and lenses", IEEE Journal on Robotics and Automation, vol. 3, no. 4, pp. 323-344, Aug. 1987
- [6]. F. Devernay, O. Faugeras. "Straight lines have to be straight". *Machine Vision and Applications*, 13:14–24, 2001.
- [7]. A. W. Fitzgibbon. "Simultaneous linear estimation of multiple view geometry and lens distortion". In *Proc. CVPR*, 2001.
- [8]. D. Claus, A.W. Fitzgibbon, "rational function lens distortion model for general cameras", 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005
- [9]. C. Harris, M. Stephens, "A combined corner and edge detector", In *Alvey Vision Conference*, pages 147– 152, 1988.
- [10]. Masanori Hariyama, Michitaka Kameyama. "Design of a Trinocular-Stereo-Vision VLSI Processor Based on Optimal Scheduling", *IEICE Transactions on Electronics* · April 2008
- [11]. Raad H. Thaher, Zaid K. Hussein, "Stereo Vision Distance Estimation Employing SAD with Canny Edge Detector", *International Journal of Computer Applications*, Volume 107 – No. 3, December 2014
- [12]. Seung-min Choi, Jae-chan Jeong, Dae Hwan Hwang, "Post-Processing Algorithms for Real-time Active Stereo Vision", *The 18th IEEE International Symposium on Consumer Electronics*, 2014
- [13]. Jim Hefferon, „Linear Algebra”, Orthogonal Publishing L3C, January 1, 2017
- [14]. Wilhelm Burger, Mark James Burge, „Digital Image Processing”, Springer, e-ISBN 978-1-84628-968-2, 2008
- [15]. Xiangjian He, Huaifeng Zhang, Namho Hur, Jinwoong Kim, Qiang Wu, Taeone Kim, „Estimation of Internal and External Parameters for Camera Calibration Using 1D Pattern”, 2006 IEEE International Conference on Video and Signal Based Surveillance, 11 December 2006
- [16]. Richard Hartley, Andrew Zisserman, „Multiple View Geometry in Computer Vision”, Cambridge University Press, isbn 0-511-18618-5
- [17]. Bo Zhou, Meng Li, Kun Qian, Xianzhong Dai, Fang Fang, "Long-range outdoor localization of a mobile robot using a binocular camera", *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 28 January 2016
- [18]. Wei Bao, Chongwei Zhang, Benxian Xiao, Rongbao Chen, "Self-localization of Mobile Robot Based on Binocular Camera and Unscented Kalman Filter", 2007 IEEE International Conference on Automation and Logistics, 08 October 2007
- [19]. Kunio TakayaKunio Takaya, "Dense stereo disparity map for video by sub-pixel dynamic time warp algorithm". *CCECE 2010*, 16 September 2010
- [20]. Naigong Yu, Chunyan Ma, "Mobile Robot Map Building Based on Cellular Automata", 2011 Third Pacific-Asia Conference on Circuits, Communications and System (PACCS), 18 August 2011
- [21]. Diana Araque, Ricardo Díaz, Byron Pérez-Gutiérrez, Alvaro Joffre Uribe, "Augmented reality motion-based robotics off-line programming", 2011 IEEE Virtual Reality Conference, 29 April 2011
- [22]. GitHub repository: [https://github.com/PatrykMazurek/stereovision\\_camera\\_calibration](https://github.com/PatrykMazurek/stereovision_camera_calibration)
- [23]. Hartley, R.I., *Theory and Practice of Projective Rectification*. *IJCV* 35 2, pp 115-127 (1999)
- [24]. Zhang, Z. "A Flexible New Technique for Camera Calibration". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 22, No. 11, 2000, pp. 1330–1334.
- [25]. Heikkila, J, and O. Silven. "A Four-step Camera Calibration Procedure with Implicit Image Correction." *IEEE International Conference on Computer Vision and Pattern Recognition*. 1997.
- [26]. Zhengyou Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations", *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999