Journal of
Electronic Imaging

JElectronicImaging.org

Improving depth maps of plants by using a set of five cameras

Adam L. Kaczmarek

SPIE.

IS&T
imaging.org

Improving depth maps of plants by using a set of five cameras

Adam L. Kaczmarek*
Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Ulica Gabriela Narutowicza 11/12,
80-233 Gdansk, Poland

Abstract. Obtaining high-quality depth maps and disparity maps with the use of a stereo camera is a challenging task for some kinds of objects. The quality of these maps can be improved by taking advantage of a larger number of cameras. The research on the usage of a set of five cameras to obtain disparity maps is presented. The set consists of a central camera and four side cameras. An algorithm for making disparity maps called multiple similar areas (MSA) is introduced. The algorithm was specially designed for the set of five cameras. Experiments were performed with the MSA algorithm and the stereo matching algorithm based on the sum of sum of squared differences (sum of SSD, SSSD) measure. Moreover, the following measures were included in the experiments: sum of absolute differences (SAD), zero-mean SAD (ZSAD), zero-mean SSD (ZSSD), locally scaled SAD (LSAD), locally scaled SSD (LSSD), normalized cross correlation (NCC), and zero-mean NCC (ZNCC). Algorithms presented were applied to images of plants. Making depth maps of plants is difficult because parts of leaves are similar to each other. The potential usability of the described algorithms is especially high in agricultural applications such as robotic fruit harvesting. [DOI: 10.1117/1.JEI.24.2.023018]

Keywords: computer vision; image acquisition/recording; image processing; stereoscopy.

Paper 14602 received Sep. 25, 2014; accepted for publication Mar. 3, 2015; published online Mar. 24, 2015.

1 Introduction

Depth maps can be obtained on the basis of two images from a stereo camera. The increase in a depth map precision is often achieved by using more advanced algorithms with higher computational complexity.[1,2] This paper presents a different approach for improving depth map quality. The improvement is achieved by taking advantage of a larger number of cameras.

Determining depth map is particularly difficult in case of images of plants. Tan et al.[3] presented a spectrum of plants according to a varying leaf size. Making depth maps of plants is the easiest when plants are small and they have big leaves. Processing stereo images of trees is the most difficult. It is caused by the fact that leaves are similar to each other and they have many areas with the same color. This is problematic in stereo matching algorithms as there are many areas of one image that have multiple candidate matches in the other image. This problem is reduced when a multicamera set is used.

This paper presents an application of a set of five cameras to making depth maps of plants. The set consists of a central camera and four cameras around it. This kind of camera arrangement was first described by Park and Inoue.[4] In order to make depth maps, they used a matching measure based on the sum of sum of squared differences (SSSD). This paper describes the result of applying this depth map making method to images of plants. The paper also presents the application of other matching measures to the set of cameras described by Park and Inoue.[4] Moreover, this paper introduces the new algorithm for making depth maps called the multiple similar areas (MSA) matching algorithm. The algorithm was specially designed for the set of five cameras described in this paper.

The original contributions of this paper are the following: (1) The analysis of using a five camera set to obtain depth maps of a plant on the basis of the SSSD measure and other measures. (2) The design of the novel MSA algorithm for making depth maps on the basis of images from a set of multiple cameras. (3) Results of using the MSA algorithm for a set of plants images. (4) Providing images of plants from a set of five cameras and ground truth data for these images.

2 Related Work

There is a large variety in the algorithms designed to create depth maps. Most of the algorithms calculate depth maps on the basis of images from a two-frame stereo vision system. Some of these algorithms were applied for making depth maps of plants. There are also stereo matching algorithms designed for different kinds of multicamera sets.

2.1 *Two-Frame Stereo Vision System*

Scharstein and Szeliski[2] presented an in-depth analysis of stereo matching algorithms designed for a pair of cameras. They created a test bed for a quantitative evaluation of stereo algorithms. They also implemented these algorithms, performed experiments, and provided taxonomy of stereo matching algorithms.

A result of a stereo matching algorithm is either a depth map or a disparity map. A disparity is the difference between the location of a viewed object in the first image from a stereo camera and the location of this object in the other image. On the basis of a disparity map, a depth map can be obtained when some additional data are available. These data include the distance between cameras and the focal length of

*Address all correspondence to: Adam L. Kaczmarek, E-mail: adam.l.kaczmarek@eti.pg.gda.pl camera lenses.[5] There are two main types of methods for obtaining a depth map: local methods and global ones.[1] In local methods, the disparity of each point is calculated on the basis of only a part of an image. On the contrary, in global methods, the disparity of every point depends on the whole image. This paper is mainly concerned with local methods. The algorithm based on the SSD function used with the set of five cameras by Park and Inoue is a kind of a local method.[4] The MSA algorithm introduced in this paper is also a local method.

As far as depth maps of plants are concerned, general purpose stereo matching algorithms can be used to obtain these maps. Nielsen et al.[6] presented experiments of this kind. In order to make depth maps of plants, the authors applied an algorithm which used the SSD measure with symmetric multiple windows.[7] Nielsen et al.[6] compared making depth maps of real plants with making depth maps of plants shown in computer-rendered images. In these images, they implemented a random noise and other modifications to make rendered plants look more natural.

Chn et al.[8] applied Microsoft Kinect camera to make depth maps of plant leaves. Chn et al.[8] explained that their motivations for the choice of this camera were small size, low weight, and low cost. In their paper, they focused on the problem of plant phenotyping and proposed an algorithm for segmentation of leaves from a depth image.

Depth maps are also created in order to make three-dimensional (3-D) models of plants. Biskup et al.[9] presented research in which they performed a 3-D reconstruction of plants' canopies on the basis of images from a stereo camera.[9] A 3-D reconstruction of plants can be also performed without making depth maps. It can be achieved by taking images of a plant from different overlapping views around this plant. As a result of this process, a model of the viewed object can be obtained. This kind of model is called a structure from motion. Structures from motion of plants were acquired by Quan et al.[10] and Tan et al.[3] Quan et al. made images of small plants in pots. Tan et al. performed experiments with trees.

Another area of technology where depth maps of plants are calculated is robotic fruit harvesting. In order to pick up a fruit, a harvesting robot needs to determine the location of this fruit and the distance to it. This can be achieved by making a depth map. Designers of harvesting robots most usually do not design new methods for measuring distance, but they take advantage of already existing ones. For example, they use the Point Grey BumbleBee2 binocular camera.[11,12] It is a commercial camera which makes it possible to acquire a depth map of viewed objects similarly to Microsoft Kinect.

2.2 *Multicamera Vision Systems*

Depth maps of various objects can be made on the basis of images from many cameras. Making a depth map with the use of three cameras is called trinocular stereovision.[13] Three cameras can be placed in different locations. Many research papers are concerned with a right angled configuration where there is the base camera, a camera located at a side of the base one and a third camera located above or below a base camera. Agrawal and Davis[13] proposed an algorithm that computes a disparity by finding in the images a path with the least matching cost. The algorithm is based on dynamic programming. Williamson and Thorpe[14] presented a trinocular stereo algorithm for the Highway Obstacle Detection system. The algorithm was designed to detect small objects on roads at a long range. Authors used the SSSD-in-inverse-distance matching algorithm.[5] The SSSD-in-inverse-distance matching algorithm is a popular algorithm used for obtaining depth maps on the basis of images from a set of multiple cameras. This algorithm can be applied to various camera configurations including trinocular stereovision. This algorithm is further described in Sec. 4.2 of this paper.

Depth maps are also prepared using images from arrays of cameras. A camera array is a set of many cameras located either on the same plane or on the same sphere.[15,16] Matusik and Pfister[15] presented a camera array which consists of 16 regularly spaced cameras located along the horizontal line. They used these arrangement for making 3-D TV scenes with moving objects. Wilburn et al.[16] from Stanford University presented the biggest camera array described in research papers. Their array consists of 100 cameras. These cameras were used for high-speed video capture. The authors analyzed various camera arrangements, e.g., 8 rows with 12 cameras in a row.

Park and Inoue[4] proposed obtaining depth maps with the use of a set of five cameras. This kind of a set was used in the research presented in this paper. In this solution, there is a central camera and four side cameras. Side cameras create with a central camera a set of four stereo cameras. Park and Inoue[4] made depth maps on the basis of a modified version of the SSSD function. The arrangement of five cameras is described in detail in Sec. 3 of this paper. The algorithm proposed by Park and Inoue[4] is described in Sec. 4.2.

A similar arrangement of cameras was applied by Hensler et al.[17] However, they used four cameras to make depth maps. There were a central camera and three side cameras. Distances between side cameras and the central one were the same for all side cameras. Hensler et al.[17] made depth maps on the basis of all pairs of cameras included in the set. Thus, also pairs consisting of two side cameras were taken into account. Hensler et al.[17] used this set to make depth maps of faces.

2.3 *Determining Distance Without Stereo Cameras*

Depth maps can be also obtained by other methods than those that use stereo cameras. Instead of a stereo camera, a structured-light 3-D scanner can be used.[18] This kind of scanner emits bands of light and it records the distortion of light on a viewed object. Moreover, there are time of flight (TOF) cameras which also emits light to measure the distance.[19] However, the major disadvantage of making depth maps of plants with the use of structured-light 3-D scanners and TOF cameras is such that the performance of these devices deteriorates when a viewed object is located in intensive natural light. Plants in a field are exposed to this kind of light. A distance to an object can also be measured by lasers.[20] This method is very precise, but a single measurement with a laser specifies a distance to only one particular point. In order to obtain a depth map of an object, a series of measurements needs to be performed. A distance can also be estimated by analyzing the size of objects in an image.[21] This requires predefining a set of objects and their sizes. It also requires that a depth map making algorithm can recognize these objects in an image.

2.4 *Related Work Summary*

In general, there are many works concerned with making depth maps with multiple vision systems and there are some works on making depth maps of plants with the use of a single stereo camera. However, apart from the work by Nielsen[6] and structure from motion systems,[3,10] there are no significant works on applying multiple vision systems to making depth maps of plants. This paper addresses this research area.

3 Five-Cameras Set

The arrangement of five cameras used in this paper is the same as the arrangement proposed by Park and Inoue.[4] Axes of all five cameras are parallel to each other. There is a central camera and four side cameras. Side cameras are located above, below, and at both sides of the central camera. Thus, there is a central (No. 0), right (No. 1), down (No. 2), left (No. 3), and up (No. 4) camera. The distance between every side camera and the central one is the same. Let I denote an image and the index of I denote the camera that made the image (I_0 is the image made by the central camera). Figure 1 presents the layout of images acquired with the use of this set of cameras.

Figure 1 also shows point P which represents an object visible by every camera in the set. The location of this point is different in every image. The point is shifted depending on the location of the cameras. Figure 1 shows the coordinates of the point P in every cameras' views. In the central image, the point P is located at coordinates (x_p, y_p). In every side image, the point P is shifted by the d value, however, the shift occurs in different directions. The value of shift is the same in all side images, because the distance between every side camera and the central camera is the same.

A standard coordinates system for digital images is used with coordinates (0,0) located at the left up corner of the image. Images made with the camera arrangement presented in this section are input data to stereo matching algorithms presented further in this paper. In these algorithms, there is an assumption that input images were calibrated and rectified.[22,23] Experiments presented in this paper are also concerned with a set of calibrated and rectified images.

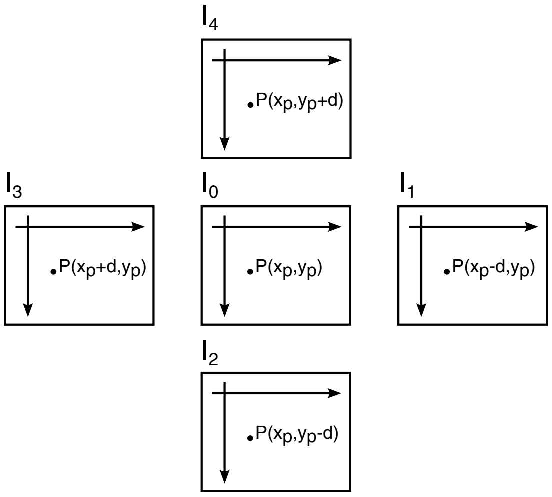

Fig. 1 The arrangement of five cameras used in this paper.

4 Matching Cost Functions

Algorithms designed to obtain depth maps take advantage of matching cost functions, which measure the level of similarity between a part of an image and a part of another image. Matching cost functions used for sets of multiple cameras are derived from cost functions used for a pair of cameras. Different functions can be applied to obtain depth maps with the use of the set of five cameras.

4.1 *Matching Cost Functions for a Stereo Camera*

In algorithms for making depth maps, the most commonly used matching cost functions are sum of absolute differences (SAD) and sum of squared differences (SSD).[1,2] However, other functions are also used, including the following: zero-mean sum of absolute differences (ZSAD), locally scaled sum of absolute differences (LSAD), zero-mean sum of squared differences (ZSSD), locally scaled sum of squared differences (LSSD), normalized cross correlation (NCC), zero-mean normalized cross correlation (ZNCC).[24] Equations (1)–(8) present equations of these functions.

The equations refer to calculations made on a pair of monochromatic images taken with the use of two cameras from the set of five cameras presented in Fig. 1. Equations are valid for pairs of images which consist of an image I_0 from the central camera and an image I_i from the side camera i. The central image is the reference one, i.e., points of a depth map corresponds to points of this image. Matching cost functions compare parts of images specified by an aggregating window W.

$$E_{\mathrm{SAD},i}(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{b}\in\mathbf{W}} |I_0(\mathbf{p}+\mathbf{b}) - I_i(\mathbf{p}+\mathbf{b}+\mathbf{d}_\mathbf{i})|, \tag{1}$$

$$E_{\mathrm{SSD},i}(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{b}\in\mathbf{W}} [I_0(\mathbf{p}+\mathbf{b}) - I_i(\mathbf{p}+\mathbf{b}+\mathbf{d}_\mathbf{i})]^2, \tag{2}$$

$$E_{\mathrm{ZSAD},i}(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{b}\in\mathbf{W}} |I_0(\mathbf{p}+\mathbf{b}) - \overline{I_0(\mathbf{W})} - [I_i(\mathbf{p}+\mathbf{b}+\mathbf{d}_\mathbf{i}) - \overline{I_i(\mathbf{W},\mathbf{d}_\mathbf{i})}]|, \tag{3}$$

$$E_{\mathrm{ZSSD},i}(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{b}\in\mathbf{W}} \{I_0(\mathbf{p}+\mathbf{b}) - \overline{I_0(\mathbf{W})} - [I_i(\mathbf{p}+\mathbf{b}+\mathbf{d}_\mathbf{i}) - \overline{I_i(\mathbf{W},\mathbf{d}_\mathbf{i})}]\}^2, \tag{4}$$

$$E_{\mathrm{LSAD},i}(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{b}\in\mathbf{W}} \left| I_0(\mathbf{p}+\mathbf{b}) - \frac{\overline{I_0(\mathbf{W})}}{\overline{I_i(\mathbf{W},\mathbf{d}_\mathbf{i})}} I_i(\mathbf{p}+\mathbf{b}+\mathbf{d}_\mathbf{i}) \right|, \tag{5}$$

$$E_{\mathrm{LSSD},i}(\mathbf{p},\mathbf{d}) = \sum_{\mathbf{b}\in\mathbf{W}} \left[I_0(\mathbf{p}+\mathbf{b}) - \frac{\overline{I_0(\mathbf{W})}}{\overline{I_i(\mathbf{W},\mathbf{d}_\mathbf{i})}} I_i(\mathbf{p}+\mathbf{b}+\mathbf{d}_\mathbf{i}) \right]^2, \tag{6}$$

$$E_{\mathrm{NCC},i}(\mathbf{p},\mathbf{d}) = \frac{\sum_{\mathbf{b}\in\mathbf{W}}[I_0(\mathbf{p}+\mathbf{b})\cdot I_i(\mathbf{p}+\mathbf{b}+\mathbf{d_i})]}{\sqrt{\sum\limits_{\mathbf{b}\in\mathbf{W}}[I_0(\mathbf{p}+\mathbf{b})^2]\cdot\sum\limits_{\mathbf{b}\in\mathbf{W}}[I_i(\mathbf{p}+\mathbf{b}+\mathbf{d_i})^2]}}, \tag{7}$$

$$E_{\mathrm{ZNCC},i}(\mathbf{p},\mathbf{d}) = \frac{\sum_{\mathbf{b}\in\mathbf{W}}(J_0\cdot J_i)}{\sqrt{\sum\limits_{\mathbf{b}\in\mathbf{W}}J_0^2\cdot\sum\limits_{\mathbf{b}\in\mathbf{W}}J_i^2}}$$
$$J_0 = I_0(\mathbf{p}+\mathbf{b}) - \overline{I_0(\mathbf{W})}$$
$$J_i = I_i(\mathbf{p}+\mathbf{b}+\mathbf{d_i}) - \overline{I_i(\mathbf{W},\mathbf{d_i})}, \tag{8}$$

where $\mathbf{p}$ is the location of a point, $\mathbf{d}$ is the disparity, I is the function that returns the intensity of a point in a monochromatic image, $\mathbf{W}$ is the aggregating window, and $\bar{I}$ is the average value of points included in the aggregating window located with regard to the disparity $\mathbf{d_i}$.

Matching cost functions are calculated for a disparity d. Equations (1)–(8) contain a disparity in the form of a vector $\mathbf{d_i}$ such that $\mathbf{d_i} = |d|$. Values of these vectors refer to the location of the i'th camera in the set of five cameras. Thus, $\mathbf{d_1} = (d,0)$, $\mathbf{d_2} = (0,d)$, $\mathbf{d_3} = (-d,0)$, and $\mathbf{d_4} = (0,-d)$. The stereo matching algorithms based on these functions calculate their results for a range of disparities $|d| \in [d_{\min}, d_{\max}]$. For Eq. (1)–(6), the matching algorithm selects the disparity for which the result of a matching cost function is the lowest. This disparity is included in the disparity map created as a result of the algorithm. In the case of algorithms using Eqs. (7) and (8), the disparity for which the result of the function is the greatest is selected to a disparity map.

4.2 *Matching Cost Functions for a Set of Multiple Cameras*

One of matching functions used for making depth maps on the basis of images from a set of multiple cameras is SSSD.[4,5] This measure can be used with various cameras' arrangements. In the research presented in this paper, SSSD is applied to a set of five cameras described in Sec. 3. Park and Inoue,[4] who took advantage of the same kind of a camera arrangement, also used the matching measure based on SSSD.

The SSSD function is a sum of SSD functions. SSSD is often used with an array of cameras placed along a horizontal line when optical axes of cameras are perpendicular to this line. When the number of cameras in such an array is equal to n, the array is regarded as a set of $n-1$ pairs of cameras. Each pair consists of a reference camera C_0 and some C_j where $1 \le j \le n$. The numeration of cameras in an array is such that a camera with the number 0 is located the farthest to the left from the point of view behind a camera array. Subsequent cameras are denoted with consecutive natural numbers.

A camera array is a kind of a multiple baseline stereo vision system.[5] The distance between a reference camera and a matching camera is different for every pair of cameras. Therefore, values of disparities are different for the same object visible in different pairs of cameras. For this reason, Okutomi and Kanade[5] proposed the SSSD-in-inverse-distance function. This function is calculated with respect to the ζ value that is the inverse of the distance to the viewed point. The value of the inverse distance is the same in all pairs of cameras. The SSSD-in-inverse-distance function is presented in Eq. (9):

$$\mathrm{SSSD}(\mathbf{p},\zeta) = \sum_{1\le j\le n}\sum_{\mathbf{b}\in\mathbf{W}}[I_0(\mathbf{p}+\mathbf{b}) - I_j(\mathbf{p}+\mathbf{b}+\mathbf{B_j}F\zeta)]^2, \tag{9}$$

where ζ is the inverse distance, I_j denotes the intensity of a point in the image from the camera C_j, F is the focal length of a camera, and B_j is the baseline referring to the distance between the camera C_0 and the camera C_j.

In the case of a five camera set presented in this paper, there is no need to use the inverse distance as an argument instead of a disparity, because distances between cameras are the same in all pairs of cameras. The application of the SSSD function to a set of five cameras is presented in Eq. (10):

$$\mathrm{SSSD}(\mathbf{p},\mathbf{d}) = \sum_{1\le i\le 4} E_{\mathrm{SSD},i}(\mathbf{p},\mathbf{d}). \tag{10}$$

Apart from the function SSSD, this paper also considers other matching cost functions. The application of these functions to the set of five cameras is one of the novelties introduced in this paper. The function SSSD derives from the function SSD by summing the results of the function SSD for four pairs of cameras in the set. Similarly, other matching cost functions for two cameras can be generalized to the set of five cameras. The matching cost function for the set of five cameras is equal to the sum of matching functions for every pair of images consisting of an image from the central camera and an image from the side camera. The central image is the reference one in every pair. The general matching cost function for the set of five cameras is presented in Eq. (11):

$$\mathrm{SE}_F(\mathbf{p},\mathbf{d}) = \sum_{1\le i\le 4} E_{F,i}(\mathbf{p},\mathbf{d}), \tag{11}$$

where the index F refers to one of the matching functions for two images presented in Eqs. (1)–(8).

Equations (10) and (11) can be also used when fewer than four pairs of images are taken into account. In such cases, results of functions $E_{F,i}$ are always equal to 0 for the camera i which is not included in calculations.

5 MSA Matching Algorithm

This section introduces a novel algorithm which has been developed by the author of this paper. The algorithm has been called MSA matching algorithm.

5.1 *MSA Algorithm for a Pair of Cameras*

Let us first suppose that there is only one pair of cameras consisting of a central camera 0 and a right camera 1. The central image is the reference one. In the MSA algorithm, the disparity is calculated for every point of an image independently from other points. Thus, the MSA algorithm is a matching algorithm of a local type.[1] In the algorithm, there are three variables that need to be set before the algorithm is applied.

$d_{\min}$—the minimum disparity
$d_{\max}$—the maximum disparity

H—the value of the threshold which determines whether two points have a similar intensity or not.

The algorithm uses a similarity function S. The function determines if intensities of two points are similar to each other or not. When the difference in points' intensities is not higher than the threshold H, then the result of function S is equal to 1 and points have a similar intensity. The result of the function S is equal to 0 otherwise. The function S is defined by Eq. (12):

$$S_n(x,y,x_n,y_n) = \begin{cases} 1 & \text{if } |I_0(x,y) - I_n(x_n,y_n)| \le H \\ 0 & \text{if } |I_0(x,y) - I_n(x_n,y_n)| > H \end{cases}, \tag{12}$$

where n is the index of a camera and the expression $I_n(x,y)$ denotes the intensity of a point with coordinates (x,y) in the image I_n.

Moreover, the MSA algorithm uses the function m. The function is calculated for disparities d that range from $d_{\min}$ to $d_{\max}$ and coordinates (x,y), which values are limited by the size of images. The result of the function m for all arguments out of these boundaries is equal to 0. The function m verifies if a point located at coordinates (x,y) in the reference image I_0 has a similar intensity as a corresponding point from the other image I_1. Coordinates of a point in the image I_1 differ from coordinates of the point in the image I_0 with respect to the value of disparity. The corresponding point is located at coordinates $(x-d,y)$. The function m for images from cameras 0 and 1 is presented in Eq. (13):

$$m_R(x,y,d) = S_1(x,y,x-d,y). \tag{13}$$

Most often, there will be many values of disparities for which the result of the function m will be equal to 1, therefore, there will be a match with a corresponding point that has a similar intensity. There will also be a sequence of disparities for which there will be this kind of a match. The next function used in the MSA algorithm which is denoted by u addresses the issue of such sequences. The function u specifies a range of possible changes in the disparity value for which a point from the reference image I_0 has a similar intensity as a corresponding point from the image I_1, i.e., the result of the function m is equal to 1.

Let us suppose that there is a point (x,y) and a disparity d for which $m_1(x,y,d)=1$. The function u depends on a maximum value of $T \in \mathbb{N}$ for which all values of $m_1(x,y,d+t)$ are equal to 1 for $-T \le t \le T$, $t \in \mathbb{Z}$. The fact that the value of T is maximum implicates that either $m_1(x,y,d+T+1)=0$ or $m_1(x,y,d-T-1)=0$. The function u is also specified for a point (x,y) and disparity d such that $m_1(x,y,d)=0$. In this case, the result of function u is equal to 0. The definition of the function u is presented in Eq. (14):

$$u(x,y,d) = \begin{cases} 0 & \text{if } m_1(x,y,d)=0 \\ T+1 & \text{if } m_1(x,y,d)=1 \end{cases}, \tag{14}$$

where T is equal to the value which satisfies the conditions presented in Eq. (15).

$$\underset{-T\le t\le T}{\forall} m_1(x,y,d+t) = 1 \wedge$$
$$\underset{\substack{t=-(T+1)\vee\\ t=T+1}}{\exists} m_1(x,y,d+t) = 0. \tag{15}$$

In the final disparity map created by the MSA algorithm, the disparity in the point (x,y) is the one for which the result of the function u is the greatest. The function D presented in Eq. (16) determines the value of a disparity inserted into a disparity map which is a result of the MSA algorithm:

$$D(x,y) = \underset{d}{\operatorname{argmax}}\, u(x,y,d). \tag{16}$$

There can be some points for which function D does not determine disparities. These are the points such that the results of the function u are equal to 0 for all analyzed disparities $d \in [d_{\min}, d_{\max}]$. In a disparity map generated as a result of the MSA algorithm, values of disparities in these points are undefined.

5.2 *MSA Algorithm for the Set of Five Cameras*

The MSA algorithm is not intended for use with a pair of cameras, but it is designed for multiple stereo cameras. In the case of a five camera set described in Sec. 3, there are four pairs of cameras taken into account.

The MSA algorithm first computes the results of the function m which is based on the similarity measure S, Eq. (12). The function m identifies points similar to each other. Equation (13) presented in the previous section defines the function m for a pair which consists of a right camera and a central one. The central camera is present in every considered pair of cameras. When a set of five cameras is used, the function m is calculated for every pair taken into account. Equation (17) presents functions for pairs containing down (m_2), left (m_3), and up (m_4) cameras:

$$\begin{aligned} m_2(x,y,d) &= S_2(x,y,x,y+d) \\ m_3(x,y,d) &= S_3(x,y,x+d,y) \\ m_4(x,y,d) &= S_4(x,y,x,y-d). \end{aligned} \tag{17}$$

After calculating the results of function m for the given range of input values, the algorithm computes results for the function u. Equation (14) in the previous subsection presents the equation of this function for a right camera. In the case of four pairs of cameras, this function is defined on the basis of the single pair version. When all four pairs of cameras are taken into account, the function u simultaneously matches points in all four pairs of cameras. The function u identifies sequences of disparities for which there are points similar to each other in every pair of cameras. The equation of the function u for four pairs of cameras is presented in Eq. (18):

$$u(x,y,d) = \begin{cases} 0 & \text{if } \underset{1\le i\le 4}{\forall} m_i(x,y,d)=0 \\ T+1 & \text{if } \underset{1\le i\le 4}{\exists} m_i(x,y,d)=1 \end{cases}, \tag{18}$$

where T matches the conditions presented in Eq. (19):

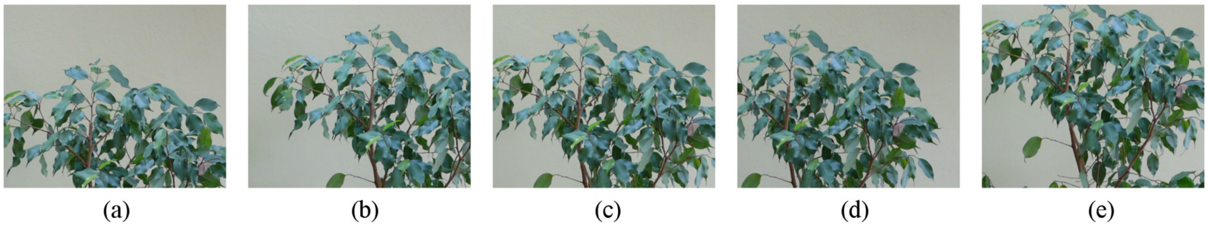

Fig. 2 Images of the ficus tree: (a) up, (b) left, (c) central, (d) right, and (e) down.

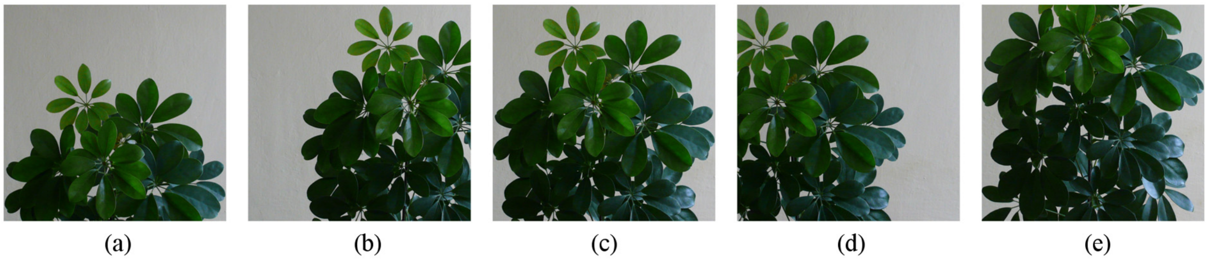

Fig. 3 Images of the dwarf umbrella tree: (a) up, (b) left, (c) central, (d) right, and (e) down.

$$\underset{-T\leq t\leq T}{\forall} m_i(x,y,d+t)=1 \quad \wedge$$
$$\underset{\substack{t=-(T+1)\vee t=T+1\\0\leq i\leq 4}}{\exists} m_i(x,y,d+t)=0. \tag{19}$$

In the last step of the MSA algorithm, results of the function u are used to obtain values of disparities that are inserted into a disparity map as a final result of the algorithm. In the MSA algorithm for two cameras, these values are obtained with the use of Eq. (16). The same equation is used in the MSA algorithm for five cameras.

The MSA algorithm can also be used with fewer than five cameras. Some side cameras may be excluded from the camera arrangement described in Sec. 3. For example, only central, right, and up cameras may be available. If the MSA algorithm is used for fewer than five cameras, the disparity map is obtained with the use of the equations presented in this section, but the parts of equations corresponding to the excluded cameras are disregarded in the calculations. For example, when a right camera is excluded, then the function m_1 is not taken into account.

6 Evaluation

The stereo matching algorithms presented in Secs. 4 and 5 were evaluated on the basis of representative test data. The results of performed experiments were evaluated with the use of quality metrics.

6.1 *Test Data*

Test data used in the evaluation consist of two sets of plant images. The first set contains images of a ficus tree (*Ficus benjamina*). The second one includes images of a dwarf umbrella tree (*Schefflera arboricola*). These sets of images are presented in Figs. 2 and 3, respectively.

The images were acquired using the camera arrangement presented in Sec. 3. A Panasonic Lumix DMC LF1 camera was used in the experiments. This model has 10.2 megapixels resolution and a focal length equal to 28 mm.

Ground truths were prepared for both set of images. Ground truth is a map of real disparities of objects visible in images. Ground truths were calculated manually by matching points in the central image with points in the side images. The values of disparities for which a match was found were marked in ground truth maps.

Parts of a central image that are close to the border of an image are occluded areas. They are not included in a ground truth map. The size of an occluded area is determined by the maximum value of the disparity $d_{\max}$ taken into account in the stereo matching algorithm. The occluded area comprises all points within the range of $d_{\max}$ points from the border of an image. For these points, the stereo matching algorithm cannot verify the matching of points for all analyzed disparities.

There are also other areas in images for which it is not possible to determine values of disparities. These are areas containing parts of objects located at the back of the viewed scene. These parts can be visible from only one camera. They are partly hidden behind other parts of objects located closer to cameras. For some points in parts of objects located far from cameras, it is not possible to determine matching points in the images from other cameras. For these points, values of disparities in ground truth are undefined.

Figures 4(a) and 5(a) present parts of images without occluded areas. Ground truth for the first set of images is presented in Fig. 4(b). Figure 5(b) presents ground truth for the second set of images. In these images, brightness intensity refers to the distance from a camera. Points with greater intensity are located closer to a camera.

6.2 *Quality Metrics*

There are two metrics commonly used to estimate the quality of disparity maps and algorithms that create them:[2] root-mean-squared (RMS) error and the percentage of bad matching pixels.

The quality estimation of disparity maps is based on comparing them with ground truth maps. The RMS error metric is a quadratic mean of differences between disparities in

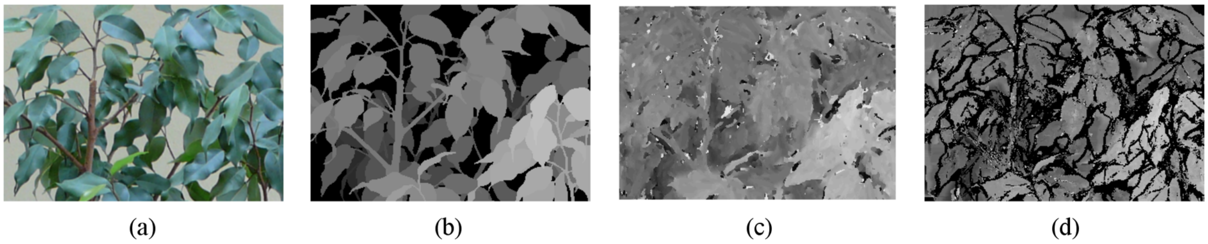

Fig. 4 Ficus Tree—ground truth and disparity maps: (a) original image, (b) ground truth, (c) the SSSD measure for five cameras, and (d) the MSA algorithm for five cameras.

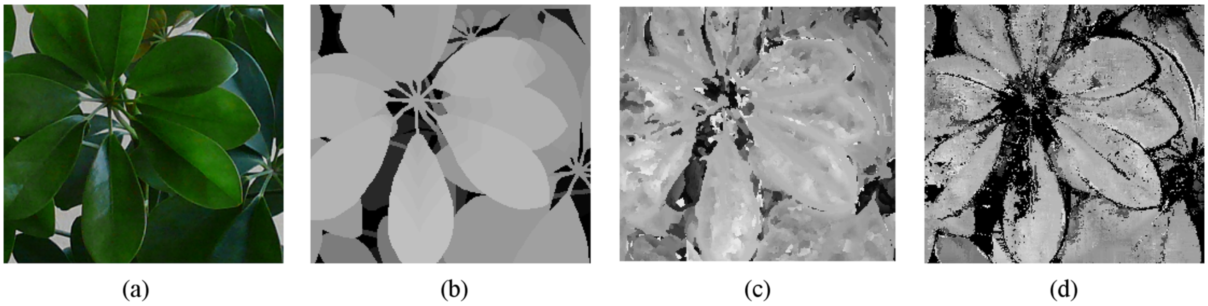

Fig. 5 Dwarf Umbrella Tree—ground truth and disparity maps: (a) original image, (b) ground truth, (c) the SSSD measure for five cameras, and (d) the MSA algorithm for five cameras.

a map acquired as a result of a stereo matching algorithm and disparities in a ground truth map. The equation of the RMS metric for disparity maps is presented in Eq. (20):

$$R = \sqrt{\left[\frac{1}{N}\sum_{(x,y)}|D_M(x,y) - D_T(x,y)|^2\right]}, \tag{20}$$

where R is the RMS error, (x, y) are coordinates, the function $D_M(x, y)$ returns the value of the disparity in the evaluated disparity map, $D_T(x, y)$ is the value of the disparity in ground truth, and N is the number of points taken into account.

The second metric, which is the percentage of bad matching pixels, specifies two kinds of points in disparity maps. There are points for which a stereo matching algorithm calculated a disparity correctly and there are points for which the value of the disparity is incorrect. The distinction between correctly and incorrectly matched points is based on a threshold. If the difference between the disparity of a point in the disparity map and the disparity of the corresponding point in ground truth is less than the value of the threshold, the disparity of the point is assumed to be correct. The disparity is incorrect otherwise. The equation for the percentage of bad matching pixels is presented in Eq. (21):

$$B = \frac{1}{N}\sum_{(x,y)}[|D_M(x,y) - D_T(x,y)| > Z], \tag{21}$$

where B is the percentage of bad matching pixels, Z is the threshold defining the disparity error tolerance, and other symbols are the same as in Eq. (20).

In the experiments presented in this paper, a level of coverage was also calculated. The coverage level is the percentage of points in ground truth that are available in the disparity map regardless of the disparity values in these points. When a stereo matching algorithm processes images to obtain a disparity map, there can be some points for which the algorithm is not able to find a match. The value of disparity is unknown for those points so they are not included in the disparity map. The coverage level is equal to the number of points that are present both in the disparity map and in ground truth divided by the number of points available in ground truth. If ground truth is not available, then the total number of points in the reference image is considered instead of the number of points included in ground truth.

6.3 *Results*

In order to analyze the performance of stereo matching algorithms for a set of five cameras, the author of this paper performed a series of experiments. The first one measured the performance of stereo matching algorithms based on different matching measures, in particular, the SSSD measure. The second experiment measured the performance of the MSA algorithm. The third experiment is concerned with selecting the parameter H for the MSA algorithm.

6.3.1 *Experiments with different matching measures*

The first experiment was focused on the quality of disparity maps obtained with the use of the SSSD measure and other matching measures. In the experiment, both set of images described in Sec. 6.1 were considered.

The ground truth for the first set of images contains values of disparities that range from 77 to 102. In the experiment, measures were not calculated for all possible values of disparities from 0 to a value significantly beyond the maximum disparity in ground truth. Experiments were conducted for the estimated range of disparities that embrace points of a viewed object. In the experiment with the first set of images, the following boundaries were set: $d_{\min} = 70$ and $d_{\max} = 110$. For the second set of images, disparities in ground truth range from 142 to 188 and boundaries were set to $d_{\min} = 140$ and $d_{\max} = 200$.

Figure 6(a) presents results for the first set of images containing views of Ficus Tree. Figure 6(b) presents results for images of Dwarf Umbrella Tree. These figures are concerned with the SSSD measure defined by Eq. (10) presented in Sec. 4.2. Charts were made for monochromatic images.

The figures show the relation between the percentage of bad matching pixels and the number of used stereo cameras. One stereo camera refers to a pair of cameras consisting of a central camera and up one. Two stereo cameras include central, left and up camera. Three stereo cameras include central, right, left, and up camera. Lastly, four stereo cameras refer to all four pairs of cameras taken into account in the five cameras set.

The percentage of bad matching pixels was calculated with the use of Eq. (21). The threshold level Z was set to 4 for both sets of images. The percentage of bad matching pixels was calculated for all points included in ground truth.

There are three plots in each figure. Each plot refers to a different size of aggregating window used in the SSSD measure. The experiment was performed for square windows of sizes 1×1, 3×3 and 5×5. Charts show that the percentage of bad matching pixels decrease with respect to the number of stereo cameras. The best results are obtained for the 5×5 window when all four pairs of cameras are taken into account. For this window size, the percentage of bad matching pixels improved from 27.97% to 15.46% for Ficus Tree and from 64.92% to 28.92% for Dwarf Umbrella Tree, when five cameras were used instead of two. On average, the number of bad matching points was reduced by 50.01%. Disparity maps created with the use of the SSSD measure in this configuration are presented in Figs. 4(c) and 5(c).

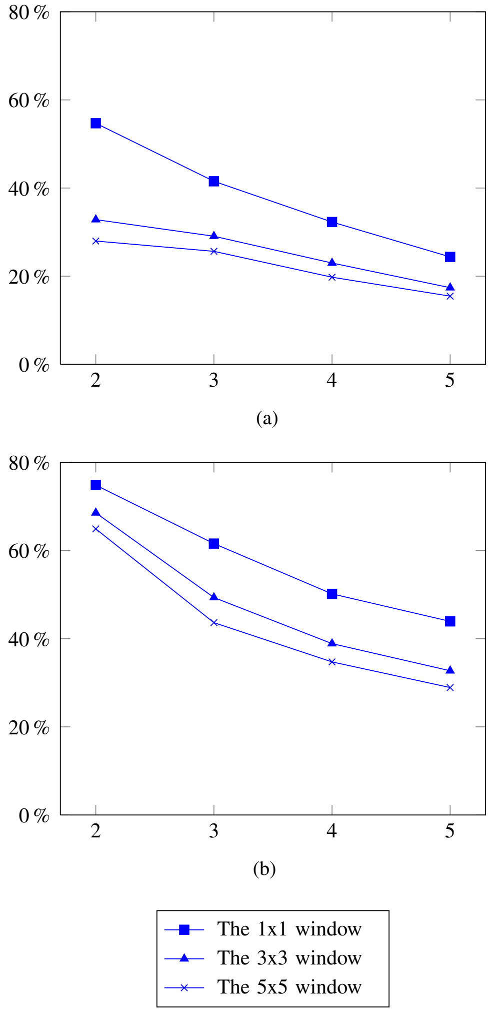

Fig. 6 The percentage of bad matching pixels for the SSSD measure: (a) Ficus Tree and (b) Dwarf Umbrella Tree.

Apart from the SSSD measure, experiments were also performed with other measures. Tables 1 and 2 present the percentage of bad matching pixels in disparity maps obtained with the use of the following measures: SAD, SSD, LSAD, LSSD, ZSAD, ZSSD, NCC, and ZNCC. Calculations were made on the basis of Eq. (11). Disparity maps were calculated with the use of all these measures for window sizes 3×3 and 5×5. The window 1×1 was used only for measures SAD and SSD. This kind of a window was not relevant for other measures. Table 1 presents results for the set of images of Ficus Tree. Table 2 shows results for Dwarf Umbrella Tree. Tables 1 and 2 show the percentage of bad matching

Table 1 Results for images of Ficus Tree.

Matching measure	Window size	Number of cameras			
		2	3	4	5
SAD	1 × 1	54.7	42.26	33.25	24.92
SSD	1 × 1	54.7	41.53	32.28	24.37
SAD	3 × 3	33.46	28.64	22.59	16.15
SSD	3 × 3	32.83	29.06	22.98	17.38
LSAD	3 × 3	41.47	42.35	36.62	31.94
LSSD	3 × 3	40.91	42.87	37.83	33.95
ZSAD	3 × 3	41.29	42.36	36.32	31.85
ZSSD	3 × 3	40.73	42.9	37.7	33.74
NCC	3 × 3	40.85	42.78	37.71	33.85
ZNCC	3 × 3	46.26	47.1	40.93	34.92
SAD	5 × 5	28.02	24.3	18.64	13.6
SSD	5 × 5	27.97	25.63	19.76	15.46
LSAD	5 × 5	31.02	33.54	30.28	25.22
LSSD	5 × 5	31.32	35.61	33.02	28.35
ZSAD	5 × 5	30.19	33.67	30.04	25.34
ZSSD	5 × 5	30.45	35.6	32.95	28.54
NCC	5 × 5	31.18	35.46	32.75	28.06
ZNCC	5 × 5	33.15	35.42	30.54	23.53

pixels when a different number of input images was taken into account.

The best results for both sets of input images were obtained when measures SAD and SSD were used with the window 5 × 5 when all five images were considered. In case of images of Ficus Tree, using the SAD measure led to better results than using the SSD measure. In case of Dwarf Umbrella Tree, the SSD measure was better than SAD. On average, the SSD measure returns better results than the SAD measure. The differences in the results obtained with measures LSAD, LSSD, ZSAD, ZSSD, and NCC were not significant for the same window size. Results which differed more from results obtained with these measures were acquired with the use of the ZNCC measure. This measure generated in some cases better results than LSAD, LSSD, ZSAD, ZSSD and NCC measures, however, in other cases, results deteriorated. Results for measures LSAD, LSSD, ZSAD, ZSSD, NCC, and ZNCC were sometimes better than results for measures SAD and SSD when two images were taken into account. However, measures SAD and SSD always proved to be better with the use of all five images.

Table 2 Results for images of Dwarf Umbrella Tree.

Matching measure	Window size	Number of cameras			
		2	3	4	5
SAD	1 × 1	74.86	61.41	51.66	45.37
SSD	1 × 1	74.86	61.59	50.18	43.94
SAD	3 × 3	69.14	49.89	40.27	34.59
SSD	3 × 3	68.57	49.35	38.89	32.73
LSAD	3 × 3	69.87	62.84	57.51	52.76
LSSD	3 × 3	69.45	62.88	57.54	52.81
ZSAD	3 × 3	68.22	63.08	56.21	51.83
ZSSD	3 × 3	67.99	63.08	56.45	51.98
NCC	3 × 3	69.42	62.78	57.39	52.67
ZNCC	3 × 3	72.61	66.08	62.2	59.13
SAD	5 × 5	66.24	44.64	36.21	30.74
SSD	5 × 5	64.92	43.65	34.74	28.92
LSAD	5 × 5	57.35	46.25	39.2	34.63
LSSD	5 × 5	56.61	46.48	40.23	35.64
ZSAD	5 × 5	55.31	46.4	38.07	33.81
ZSSD	5 × 5	54.44	46.35	38.81	34.83
NCC	5 × 5	56.52	46.28	39.98	35.32
ZNCC	5 × 5	58.79	48.02	42.5	38.9

For all measures, the percentage of bad matching pixels was lower for the window 5 × 5 than for the window 3 × 3 when the same number of input images were included in the calculations. In general, the increase in the number of cameras leads to the decrease in the percentage of bad matching pixels. However, there are some exceptions from this rule. For example, in the case of images of Ficus Tree, all measures apart from SAD and SSD generated better results for two images than for three images. Nevertheless, in all cases, results obtained with the use of five images were better than results acquired with the use of two images.

6.3.2 *Experiments with the MSA algorithm*

In the experiments with the MSA algorithm, the same values of $d_{\min}$ and $d_{\max}$ were used as in the experiments presented in the previous subsection. MSA requires an additional parameter H that is a threshold defining points regarded as similar to each other [Eq. (12)]. The value of H was set to 17 for the set of images containing Ficus Tree. The parameter was set to 13 for the set with views of Dwarf Umbrella Tree. The problem of selecting the value of H parameter is discussed in the next Sec. 6.3.3. Disparity maps generated by the MSA algorithm for these values of the parameter H are presented in Figs. 4(d) and 5(d).

Figure 7 presents the results for the MSA algorithm for both set of images. The figure contains charts calculated for the 1 × 1 window for monochromatic images. The figure presents the percentage of bad matching pixels. Figure 7 also contains coverage levels defined in Sec. 6.2.

The increase in the number of cameras taken into account causes the decrease in the percentage of bad matching pixels. The percentage of bad matching pixels decreased from 49.04% to 13.88% for Ficus Tree and from 65.33% to 30.28% for Dwarf Umbrella Tree. On average, the number of bad matching points was reduced by 62.67%.

Moreover, the MSA algorithm for a window of size 1 × 1 provides similar values of the percentage of bad matching pixels as the algorithm based on the SSSD measure for the 5 × 5 window when five cameras are used. The average difference is equal to 0.11%. On average, SSSD for the 5 × 5 window provides a percentage of bad matching pixels equal to 22.19%. In MSA, the average percentage of bad matching pixels is equal to 22.08%. MSA achieves similar results as SSSD, but it requires a much smaller window size. It is an important advantage of the MSA algorithm that makes it possible to obtain disparity maps faster and with the use of fewer computations. However, the coverage level is decreasing in the MSA algorithm when more cameras are used. The algorithm does not provide results for all points, but for the resolved ones, the results are accurate.

This feature of the MSA algorithm determines the algorithm's robustness. The algorithm can be executed on images containing a random noise and images that have low quality for other reasons. The quality of the input images has a major impact on the coverage level of disparity maps obtained with the use of MSA. In the case of low-quality images, the algorithm will not be able to match points from different cameras. This will occur in particular in the case of taking into account all images from the five cameras set. However, the percentage of bad matching pixels for points included in a disparity map will not be affected by the quality of the input images as much as the coverage level. Nevertheless, the coverage level,

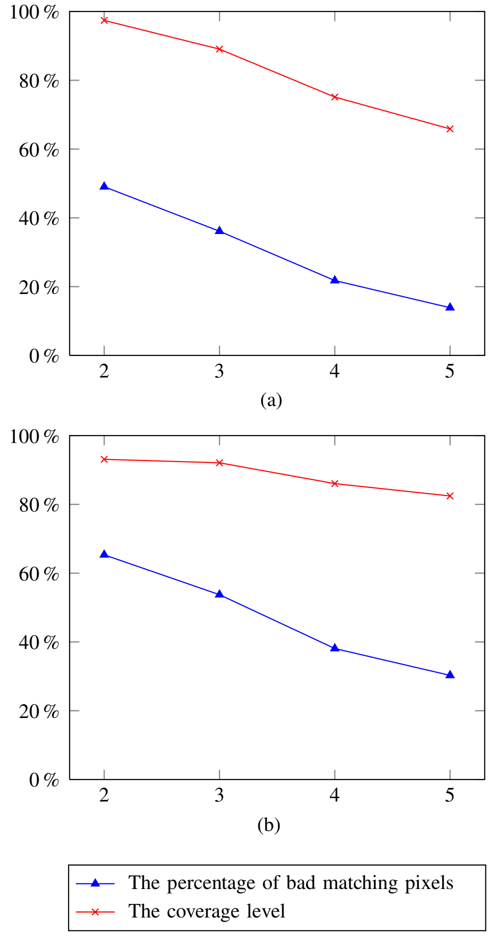

Fig. 7 The results of the MSA algorithm: (a) Ficus Tree and (b) Dwarf Umbrella Tree.

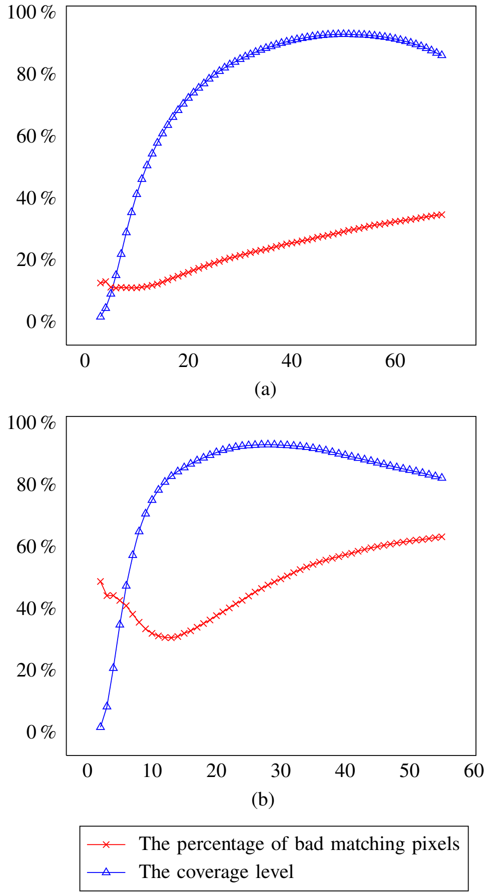

Fig. 8 Differences in the threshold level in the MSA algorithm: (a) Ficus Tree and (b) Dwarf Umbrella Tree.

regardless of the quality of the input images, can be managed by adjusting the threshold level used in the algorithm. The problem of selecting the value of the threshold is described in the next subsection.

6.3.3 *Threshold value in the MSA algorithm*

Figure 8 presents the results of the MSA algorithm for different values of the threshold H. The figure shows the percentage of bad pixels and coverage levels for both set of images when all five cameras are taken into account.

In every chart, there is a value of a threshold for which the percentage of bad matching pixels and coverage levels reach the best result. The best result is a minimum in the case of the percentage of bad matching pixels metric and a maximum in the case of a coverage level. These extrema are reached in every chart for different values of thresholds.

For Dwarf Umbrella Tree, the best value of the percentage of bad matching pixels is equal to 30.28%. It is reached when the threshold is equal to 13. The maximum coverage level for this set of images is equal to 92.63% when the threshold is 28. The percentage of bad matching pixels is a more important metric than the coverage level. Therefore, the best value of a threshold for images of Dwarf Umbrella Tree is set to 13. The coverage level is then equal to 82.44%.

The percentage of bad matching pixels for Ficus Tree reaches the best value 10.63% when the threshold is equal to 6. However, the coverage level is equal to 14.75% for this threshold. It is a very low value. In the range of thresholds between 6 and 50, there is a tradeoff between the coverage level and the percentage of bad matching pixels. For the threshold equal to 17, the percentage of bad matching pixels is equal to 13.88% and the coverage level is equal to 65.87%. The coverage level is acceptable for this threshold and the value of the percentage of bad matching pixels does not differ significantly from the maximum value. Therefore, the threshold in this set of images is set to 17.

In the process of selecting the value of the threshold, both the bad matching points measure and coverage measure need to be taken into account. When the threshold is too high, the results of these measures deteriorate. Similarly, there are poor results of both of these measures when the threshold value is too low. There is some range of threshold values for which it is needed to make a compromise between the number of bad matching points and the coverage level. On the basis of images presented in this paper, on average, the value of the threshold should be equal to 15.

The value of the threshold can be improved for a particular set of input images that the MSA algorithm processes. The algorithm is intended to use when there is a set of input images without ground truth available for these images. Ground truth is necessary to calculate the percentage of bad matching pixels. However, it is not needed to calculate the coverage level. The coverage level is determined only on the basis of a disparity map generated by the algorithm.

If the coverage level obtained as a result of using some threshold is not suitable, then the threshold can be changed and the MSA algorithm can be rerun. Subsequent values of thresholds can be estimated with regard to the desired difference in the coverage level. This process can be executed iteratively in order to achieve a demanded level of coverage. Nevertheless, it is time consuming.

Figure 8 shows that there is a relation between the coverage level and the percentage of bad matching pixels. Therefore, selecting the threshold appropriate for the coverage level has an impact on the percentage of bad matching pixels. For example, if a target coverage level for input images presented in this paper is set to 70%, the threshold level needs to be equal to 19 in the case of Ficus Tree images and it needs to be equal to 9 for Dwarf Umbrella Tree. For these levels, the percentage of bad matching pixels is equal to 15.15% for Ficus Tree and 33.16% for Dwarf Umbrella Tree. These are not optimal values, but they differ from the best values of percentages of bad matching pixels less then 5%. Therefore, setting the target coverage level to 70% and rerunning the algorithm leads to obtain such results.

7 Conclusions

Taking advantage of the five camera set makes it possible to increase the quality of disparity maps of plants both with the use of different kinds of matching measures and the MSA algorithm. In case of the SSSD measure with an aggregating window of size 5×5, the number of bad matching pixels is reduced by ca. 50% when five cameras are used instead of two. The average improvement in case of the MSA algorithm is equal to ca. 63%. The MSA algorithm is dedicated for a camera arrangement described in this paper. This algorithm for a window of size 1×1 acquires similar results as the algorithm based on the SSSD measure with the 5×5 window. Moreover, measures based on sums of SSD and SAD return better results than measures based on LSAD, LSSD, ZSAD, ZSSD, NCC, and ZNCC when the set of five cameras is used. On average, SSD is also better than SAD for this set.

The research presented in this paper can be used in control systems of autonomous robots. Such robots can perform actions in a real environment. For example, they are used in robotic fruit harvesting. This kind of robot needs to determine the distance to a plant and its parts. Using the set of five cameras instead of a single stereo camera would have a great influence on the efficiency of harvesting robots.

The five cameras set can be applied to determine not only disparity maps of plants but also disparity maps of other objects. The importance of methods for obtaining this kind of map rises because of a growing popularity of 3-D TV and interactive 3-D video games. The research presented in this paper can be used in these applications. In future work, the author plans to develop methods of estimating the best value of the threshold used in the MSA algorithm on the basis of features of the input images and their characteristics. The estimation will be performed before the execution of the algorithm. The future work also includes application of the presented set of cameras to objects other than plants. In particular, objects for which it is difficult to obtain a disparity map will be examined.

References

1. N. Lazaros, G. C. Sirakoulis, and A. Gasteratos, "Review of stereo vision algorithms: from software to hardware," *Int. J. Optomechatron.* **2**(4), 435–462 (2008).
2. D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vision* **47**, 7–42 (2002).
3. P. Tan et al., "Image-based tree modeling," *ACM Trans. Graph.* **26**(3), **26**(3), 87 (2007).
4. J.-I. Park and S. Inoue, "Acquisition of sharp depth map from multiple cameras," *Sig. Process. Image Commun.* **14**(1–2), 7–19 (1998).
5. M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(4), 353–363 (1993).
6. M. Nielsen et al., "Ground truth evaluation of computer vision based 3D reconstruction of synthesized and real plant images," *Precision Agric.* **8**(1–2), 49–62 (2007).
7. A. Fusiello, V. Roberto, and E. Trucco, "Symmetric stereo with multiple windowing," *Int. J. Pattern Recognit. Artif. Intell.* **14**, 1053–1066 (2000).
8. Y. Chn et al., "On the use of depth camera for 3D phenotyping of entire plants," *Comput. Electron. Agric.* **82**(0), 122–127 (2012).
9. B. Biskup et al., "A stereo imaging system for measuring structural parameters of plant canopies," *Plant Cell Environ.* **30**(10), 1299–1308 (2007).
10. L. Quan et al., "Image-based plant modeling," *ACM Trans. Graph.* **25**, 599–604 (2006).
11. L. Yang et al., "A fruit recognition method for automatic harvesting," in *14th Int. Conf. Mechatronics and Machine Vision in Practice, 2007*, pp. 152–157 (2007).
12. Q. Feng et al., "A new strawberry harvesting robot for elevated-trough culture," *Int. J. Agric. Biol. Eng.* **5**(2), 1–8 (2012).
13. M. Agrawal and L. Davis, "Trinocular stereo using shortest paths and the ordering constraint," *Int. J. Comput. Vision* **47**(1–3), 43–50 (2002).
14. T. Williamson and C. Thorpe, "A trinocular stereo system for highway obstacle detection," in *1999 IEEE Int. Conf. Robotics and Automation*, Vol. 3, pp. 2267–2273 (1999).
15. W. Matusik and H. Pfister, "3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes," *ACM Trans. Graph.* **23**, 814–824 (2004).
16. B. Wilburn et al., "High performance imaging using large camera arrays," *ACM Trans. Graph.* **24**, 765–776 (2005).
17. J. Hensler et al., "Hybrid face recognition based on real-time multi-camera stereo-matching," *Adv. Visual Comput.* **6939**, 158–167 (2011).
18. W. Jang et al., "Structured-light stereo: Comparative analysis and integration of structured-light and active stereo for measuring dynamic shape," *Opt. Lasers Eng.* **51**(11), 1255–1264 (2013).
19. W. Kazmi, S. Foix, and G. Alenya, "Plant leaf imaging using time of flight camera under sunlight, shadow and room conditions," in *IEEE Int. Symp. Robotic and Sensors Environments (ROSE)*, pp. 192–197 (2012).
20. K. Tanigaki et al., "Cherry-harvesting robot," *Comput. Electron. Agric.* **63**(1), 65–72 (2008).
21. J. Baeten et al., "Autonomous fruit picking machine: a robotic apple harvester," *Field and Service Robotics* **42**, 531–539 (2008).
22. J. Yang et al., "Multiview image rectification algorithm for parallel camera arrays," *J. Electron. Imaging* **23**(3), 033001 (2014).
23. Y.-S. Kang and Y.-S. Ho, "An efficient image rectification method for parallel multi-camera arrangement," *IEEE Trans. Consumer Electron.* **57**, 1041–1048 (2011).
24. A. Giachetti, "Matching techniques to compute image motion," *Image Vision Comput.* **18**(3), 247–260 (2000).

Adam L. Kaczmarek is an assistant professor at Gdansk University of Technology, Poland. He received his MSc, Eng. degrees in informatics from the university in 2005. He also received his PhD degree in informatics from Gdansk University of Technology in 2012. He is the author of more than 20 papers and book chapters. His current research interests include vision systems, information management, and information retrieval.