Journal of Intelligent & Robotic Systems
https://doi.org/10.1007/s10846-019-01117-8

3D Vision System for a Robotic Arm Based on Equal Baseline Camera Array

Adam L. Kaczmarek[1]

Received: 8 July 2019 / Accepted: 29 October 2019

Abstract

This paper presents a lightweight 3D vision system called Equal Baseline Camera Array (EBCA). EBCA can work in different light conditions and it can be applied for measuring large range of distances. The system is a useful alternative to other known distance measuring devices such as structured-light 3D scanners, time-of-flight cameras, Light Detection and Ranging (LIDAR) devices and structure from motion techniques. EBCA can be mounted on a robotic arm without putting significant load on its construction. EBCA consists of a central camera and a ring of side cameras. The system uses stereo matching algorithms to acquire disparity maps and depth maps similarly as in case of using stereo cameras. This paper introduces methods of adapting stereo matching algorithms designed for stereo cameras to EBCA. The paper also presents the analysis of local, semi-global and global stereo matching algorithms in the context of the EBCA usage. Experiments show that, on average, results obtained from EBCA contain 37.49% less errors than the results acquired from a single stereo camera used in the same conditions.

Keywords Camera array · Stereo vision · 3D vision · Stereo camera · Robotic arm

1 Introduction

A vision system is one of the most important part of an autonomous robot designed for recognizing objects in its vicinity and interacting with them. If a robot operates a robotic arm for picking up objects, then it is necessary to estimate distances to these objects and their locations. In general, this requires equipping the robot with a 3D vision system. There is a large variety of 3D imaging devices designed for obtaining depth maps consisting of distances from an imaging device to objects located within its field of view [1]. These devices have different characteristics and features.

Commonly used equipment for making depth maps are structured light 3D scanners [2]. Scanners are active devices which emit rays of light in order to perform the measurement. Rays have a form of a precisely defined patterns such as horizontal stripes. Scanners analyze distortions of emitted light on objects. The necessity to illuminate objects has a significant negative consequences. Foremost, the presence of intensive natural light interfere with the measurement making it hard or even impossible to obtain a scan. There is also a problem with 3D imaging of large objects such as buildings because a 3D scanner would require a powerful source of light. 3D scanners are also relatively large devices because they contain a considerable light source.

3D imaging is also performed using lasers. Lasers are used in Time-of-flight cameras (TOF) and Light Detection and Ranging equipment (LIDAR) [3, 4]. LIDAR consists of a mobile part which detects in a single measurement a distance from the device to a single point of an object. Distances to many points are obtained by aiming the ranging system at different directions. On the contrary to LIDAR, TOF simultaneously obtains distances for multiple points by using a single laser beam. Devices based on lasers provide accurate values for large distances and they are resistant to the influence of sunlight more than structured light scanners. However, both of these kinds of devices collect distances as a set of isolated points distant from each other. Therefore, the resulting depth map is sparse in comparison to maps obtained from cameras as resolution

✉ Adam L. Kaczmarek
adam.kaczmarek@pg.edu.pl

[1] Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, ul. G. Narutowicza 11/12, 80-233 Gdansk, Poland

Published online: 04 December 2019

of laser based devices is much lower than resolution of cameras [5]. Moreover, cameras are more flexible regarding its size and range of distances as there are camera arrays whose size is approximately the same as a size of a coin [6].

3D shapes of real objects and depth maps can be also acquired with the use of the Structure from Motion (SfM) technology [7]. Input data to an algorithm using SfM are images taken from different points of view located around the object. The algorithm obtains a 3D scan by matching characteristic points visible in images. The disadvantage of SfM is such that it requires relocating an imaging device or using a set of such devices distant from each other. Therefore, the Structure from Motion technology is not suitable for an autonomous robot that needs to be equipped with a vision system making 3D images from a single point of view.

The solution to these problems is using sets of adjacent cameras such as stereo cameras. 3D data can be retrieved from a pair of images with the use of stereo matching algorithms. Stereo cameras do not need to be relocated to make a 3D image. They are in large extent resistant to a negative influence of intensive natural light. Cameras differ in both their size and vision range. A light and small camera can be mounted on a robot, in particular its robotic arms, without significantly increasing the weight. However, stereo cameras have also a major weakness. The quality of 3D images obtained from stereo cameras is lower than the quality of images acquired with the use of other kind of devices for 3D imaging.

One of the methods of solving this problem is developing better algorithms for processing stereo images in order to retrieve 3D data. This paper focuses on the other approach which is improving the results by taking advantage of a greater number of adjacent cameras. In general, such a set is called a camera array. Different kinds of arrangements of cameras in an array were developed. One of the most popular are cameras placed along a straight line [8]. This paper is dedicated to a specific kind of a camera array in which there is a central camera and a ring of side cameras equally distant from the central one [9, 10]. This camera array was previously called Equal Baseline Multiple Camera Set (EBMCS) [11, 12]. In this paper it will be called Equal Baseline Camera Array (EBCA), because this name is more accurate. The most significant advantage of this array is such that EBCA makes it possible to obtain greater quality of 3D images than a stereo camera preserving valuable features of stereo cameras such as compact size and weight.

The original contributions of this paper include: (1) The development of 3D vision algorithms for Equal Baseline Camera Array which is a lightweight 3D vision system suitable for mounting on a robotic arm in an eye-in-hand configuration. (2) Designing a method for applying stereo matching algorithms to a EBCA. The method makes it possible to improve results between 21.03% and 45.16% in comparison to results obtained from a stereo camera. (3) The analysis of different types of stereo matching algorithms in the context of the EBCA usage. (4) Experiments presenting the quality of algorithms designed for EBCA.

2 Related Work

The main area of application of 3D vision systems for autonomous robots and automated machines is industrial environments, in particular manufacturing. Pérez et al. presented a detailed review on machine vision techniques in this field [1]. The review of 3D data acquisition and processing technologies for industrial applications was also prepared by Bi and Wang [13]

2.1 Types of Robots

In general, automated manufacturing is based on preprogrammed robots which constantly perform the same actions. These robots are not equipped with any kind of a vision systems. Their purpose is to operate in repeatable cycles in which robots interact with the same kind of objects placed in the same locations.

Another type of robots are those which contain a vision system. Such a robot can be either controlled by a human operator or by an autonomous robot control system. In manually controlled robots the operator directs actions of the robot on the basis of the interpretation of robotic sensors readings. There are Remotely Operated Vehicles (ROV). The actions of the robot can be realized by its parts such as robotic hands. The other kind of robots are autonomous ones which process the data from sensors fully by themselves. There robot recognize events captured by sensors, make decisions, plans their response to recognized events and performs actions without direct human control. The 3D vision system proposed in this paper can be applied to both of these kinds of robots.

Manually controlled robots equipped with a vision system are mainly used in the environments where it is hard, dangerous or even impossible for a human being to perform necessary actions. The area of application includes medical procedures [14], underwater operations [15], space technologies [16], rescue operations [17] and various industrial applications such as robots on oil platforms [18].

Although the technology of autonomous robots is still under development, this kind of robots are already used in industrial applications. A large number of researches focused on developing harvesting robots for agriculture. Hayashi et al. developed a strawberry harvesting robot which was later released as a commercial product by

Shibuya Seiki CO., LTD. company [19]. Fruit harvesting robots were also developed for automatically picking up apples [20], cucumbers [21] and oranges [22]. Moreover a research was also performed on assembling 3D objects by robots with the use of 3D vision system. Such a robotic system is often developed for small objects such as toy blocks but the same kind of technology can be used for large objects. Wan et al. proposed a robot that detects 3D objects having different shapes, plan their grasps and assemble them using artificial intelligence searching [23]. Sanchez-Lopez developed an autonomous mobile robot manipulator for picking up objects having different color [24]. There are also unmanned surface vehicles operating on the surface of the water and autonomous on-road vehicles [25, 26]. Moreover, Lin et al constructed an robotic arm for semi-automatically locating and picking up fragments of walls for the purpose of maintaining a vacuum vessel of the superconductor tokamak [27].

2.2 Locations of a Vision System in a Robot

The location of the vision system can be different in relation to the position of a robot end-effector in both manually controlled robots and autonomous ones. In general, eye-to-hand and eye-in hand setups are used. In the eye-to-hand setup the vision system in mounted separately from a robot manipulator such as a robotic arm. Sensors monitor the workspace regardless of the manipulator's current position in this configuration. Therefore, the robot constantly have a wide view on its operating field. The eye-in-hand setup means that the vision system is fixed to the construction of the manipulator close to the end-effector. In this setup the field of view is more narrow than in the eye-to-hand configuration. Moreover, mounting the vision system on the manipulator puts a load on its construction. However, the advantage of this setup is such that the robot can inspect targets from a close distance to the end-effector. The EBCA vision system described in this paper is mainly intended for use with the eye-in-hand setup. However, it can be also applied to the eye-to-hand setup.

The selection of the vision system setup depends on applications in which an autonomous robot is used. The eye-in-hand setup is frequently used in autonomous robots for fruit harvesting [19, 20, 22, 28]. Lin et al. also used a vision system located on the manipulator in the maintenance robot for a vessel of tokamak [27]. Palli et al. used a robot with an eye-in-hand vision system for underwater operations [29]. Jiang and Wang described a space station robot equipped with two monocular cameras and two stereocameras fixed to a robotic manipulator [16]. One of the stereo cameras was mounted near the gripper and headed in its direction. The second setup i.e. eye-to-hand was used by Wan et al. in their autonomous robot for assembling 3D blocks [23]. Sanchez-Lopez also used this setup in the robot for picking up colored objects [24].

2.3 Usage of Cameras in 3D Vision Systems

A 3D vision system of a robot may consist of a structured-light scanner such as Kinect [30], LIDAR [4], TOF camera, a set of cameras distant from each other (Structure from Motion technology) [7], a stereo camera or a camera array. This paper focuses on using cameras mounted close to each other because this kind of a 3D vision system can be light, tiny, compact, energy efficient, applicable to objects of different sizes and usable in various light conditions. Moreover, cameras make it possible to obtain a dense depth map and they can be easily mounted on a robotic arm.

Images from stereo cameras and camera arrays are processed by stereo matching algorithms in order to retrieve information concerning locations of real objects in a 3D space. These algorithms use the fact that the same objects visible in images from cameras placed at different points of view will be located at different coordinates at these images. A stereo matching algorithm calculates a disparity map and on its basis a depth map is acquired. Disparity is the difference between locations of the same object in different images. The set of disparities obtained for many points of images forms the disparity map. Stereo matching algorithms search for occurrences of the same objects in different images. A disparity map can be unambiguously converted to a depth map containing values of distances between the camera set and objects visible in images. In order to perform the transformation it is necessary to obtain data about cameras such as distances between them and focal length of lens. These data can be extracted in the calibration process based on making series of images of a precisely defined image pattern [31].

Algorithms for stereo cameras, camera arrays and camera matrices use the same technology of stereopsis which is based on matching views of objects visible in different images. Stereo cameras are the most commonly used and a large number of different stereo matching methods have been developed. There are also advanced rankings of these algorithms. One of the most popular ranking of stereo matching algorithms is available on Middlebury Stereo Vision Page (http://vision.middlebury.edu/stereo/) [32]. The third version of the ranking includes 97 algorithms as at 18 Jun 2018. Middlebury Stereo Vision Page provides a testbed for evaluating stereo vision algorithm. The testbed consists of datasets with stereo pairs, disparity maps ground truth and SDK for executing tests.

The KITTI Vision Benchmark Suite (http://www.cvlibs.net/datasets/kitti/) provides another well known ranking of stereo matching algorithms [33, 34]. The ranking is oriented towards testing algorithms for the purpose of controlling autonomous cars. The KITTI evaluation is based on different datasets than the Middlebury ranking. 134 algorithms were included in KITTI as of 18 Jun 2018.

Both of these rankings evaluate the semi-global block matching (StereoSGBM) algorithm available in the OpenCV library [31]. The algorithm is based on a stereo matching algorithm introduced by Hirschmuller [35]. StereoSGBM is a popular and widely used algorithm because the OpenCV library provides an optimized and well-tested implementation. The algorithm can generate results in real-time without the necessity for a high computing power. The StereoSGBM algorithm was used in the experiments presented in this paper.

Experiments were also conducted for algorithms provided by Middlebury Stereo Vision Page. The page shares implementations of algorithms for obtaining disparity maps by minimizing the cost function based on Markow Random Fields (MRF) [32]. These algorithms iteratively improves the quality of disparity maps. Graph Cuts using Expansion Moves (GC Expansion) is one of algorithms provided by Middlebury Stereo Vision Page [36]. Previous research performed on EBCA by the author of this paper showed that the best results are obtained when this algorithm is used with the Exceptions Excluding Merging Method (EEMM) designed for applying stereo matching algorithms to EBCA [11]. This paper presents more advanced methods of applying stereo matching algorithms to EBCA. Section 5 of this paper presents a further description of stereo matching algorithms in the context of using them with EBCA.

A set of over two adjacent cameras forms a camera array. Venkataraman et al. presented work related to the usage of camera arrays [6]. They also developed an ultra-thin camera array called PiCam that has approximately the same size as a coin. Their solution consists of 16 sensors in the 4x4 configuration. Each adjacent sensors was equally distant from each other. Okutomi and Kanade presented a paper on obtaining depth maps with the use of a linear camera array [8]. Wilburn proposed an array consisting of 100 cameras [37]. They have experimented with different arrangements of cameras in the set. This paper focuses on an array proposed by Park and Inoue [9]. Their five camera array was named by the author of this paper Equal Baseline Camera Array (EBCA).

3 Equal Baseline Camera Array

Equal Baseline Camera Array (EBCA) consists of a central camera and side cameras that are equally distant from the central one. The requirement that side cameras are equally distant is an essential EBCA feature that significantly improves the usability of this set for the purpose of obtaining disparity maps. Side cameras are located above, below and at both sides of the central one in EBCA consisting of five imaging devices. All cameras are aimed in the same direction.

This kind of a camera set was introduced by Park and Inoue [9]. Fehrman and McGough also performed research on such a set [38, 39]. They had a camera matrix build from 16 cameras in 4x4 configuration, however they analyzed the usage of the selected five cameras forming EBCA from the matrix. The author of this paper further developed stereo matching algorithms for EBCA and researched its capabilities [10–12, 40]. The photo of an operational real EBCA used in the experiments presented in this paper is presented in Fig. 1. The array consists of MS LifeCam Studio web cameras with the 1080p HD sensor fixed to an aluminum frame.

Parameters of the array are presented in Table 1 [41]. The aluminum rig used for holding cameras was not optimized in terms of weight minimization as it is possible to use a lighter rig which is not as robust as the used one. Mounting elements originally provided with cameras were removed. Cameras were fixed to rig using screws. An operating range of the whole EBCA coincides with ranges of cameras. A maximum distance is additionally limited by a resolution of cameras because only large objects are visible in images taken from long distances.

EBCA consisting of five cameras forms a set of four stereo cameras that share a common, central camera. There are, in total, ten pairs of cameras in this set. However, pairs which do not consist of a central camera are excluded from calculations in order to unify the viewpoint for all used stereo pairs. Stereo matching algorithms process images from stereo cameras by distinguishing between the image from the reference camera and the image from a side

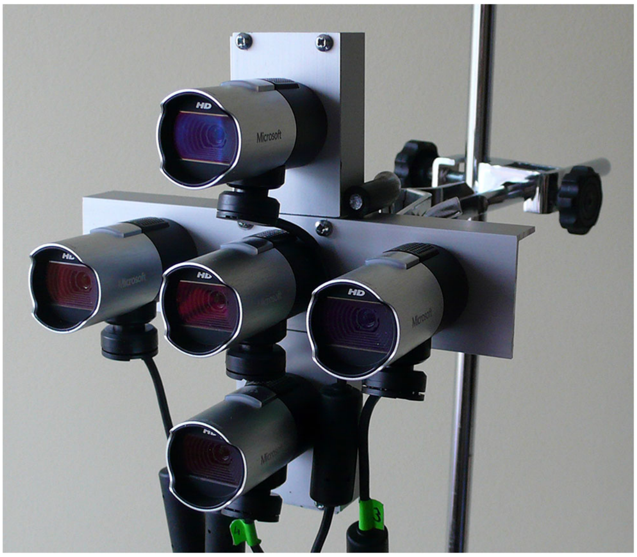

Fig. 1 Real EBCA used in the experiments

Table 1 Parameters of EBCA used in the experiments

Parameters of EBCA used in experiments	
Dimensions ($w \times d \times h$) [mm]	165 × 80 × 165
Distance between cameras	50mm
Weight	450g
Total price	400 USD
Parameters of each camera included in EBCA	
Camera model	MS LifeCam Studio
Sensor Resolution	1280 x 720
Field of view (diagonal)	75°
Operating range	from 0.1m
Frames per second	up to 30
Weight (without parts not used in EBCA)	30g
Price	75 USD

camera. The reference camera makes a image whose points corresponds to points of a disparity map obtained by a matching algorithm. This camera is a point of view of a stereo set. A side camera is used to determine values of disparities. In EBCA the reference camera is always the central one so all stereo cameras have the same reference camera. It is a crucial feature characterizing EBCA along with the requirement of preserving the same distances between a central camera and side ones.

EBCA can be perceived as a sequence of cameras similar to a camera array with cameras placed along a straight line. Okutomi and Kanade wrote an influential paper on obtaining disparity maps with the use of a linear camera array [8]. They have processed images from the array as a set of pairs of images such that every pair consisted of an image from the first camera of the array and some other camera. The first camera was a reference camera for every pair likewise a central camera is a reference camera in every camera pair considered in EBCA.

Considered pairs from a camera array have different baselines i.e. distances between cameras forming a stereo camera. It is a significant problem in case of obtaining disparity maps [8]. The greater is the baseline, the higher is the disparity for objects located in the same distance from a stereo camera. As a consequence, disparities in different stereo cameras selected from the array are different for the same objects visible in images from these cameras. It cause that merging data from all cameras included in the array is more difficult. Okutomi and Kanade resolved this problem by processing values of inverse distances between cameras and objects visible in images instead of disparities. The inverse distance is the same for every stereo camera whose reference camera is the first camera in the array. However, using this method complicates calculations and it requires matching parts of images that do not coincide with an integer number of image points. As far as EBCA is concerned distances between stereo cameras used in EBCA are the same, therefore there is no need to use inverse distances instead of disparities. This is a significant advantage of the set.

In order to use EBCA it is necessary to connect it to a computer. The greatest problem with connecting the array to a computer is a large number of USB ports which needs to be used. EBCA can be used with several kinds of equipment including the following:

- desktop computer with at least 6 USB interfaces
- mobile computer with PCMCIA, ExpressCard/34 or other expansion card
- any kind of computer using USB Hub
- a separate computer (such as Raspberry Pi) for every camera

In our experiments we used mobile computers with both a 4 port USB 2.0 ExpressCard/34 expansion card produced by Gembird Electronics Ltd. and a 4 port USB 2.0 hub produced by A4TECH. We used Fujitsu Siemens Esprimo Mobile U9200 and Lenovo ThinkPad SL510 laptops. Taking images does not require high computing power. Therefore, old and low-cost computers can be used for this purpose.

The easiest method of using the array is to use a desktop computer. 6 USB ports are then required because mouse and keyboard input devices can be connected via an USB hub to a single USB port. In this case all five cameras can be directly plugged into a computer. Using independent USB port for each camera makes it possible to use all cameras simultaneously. If many cameras are connected using a USB hub to the same USB port then turning on one camera requires switching off all other cameras using the hub. It is sufficient when only images needs to be taken, however such a solution makes it impossible to record videos. If a desktop computer is not equipped with 6 USB ports it is easy to expand the machine using adapter cards, e.g. an PCI-E USB add-on card.

Nevertheless using a desktop computer excludes the possibility of usage in out-door environment. It is harder to increase the number of USB ports in mobile computers. However, there is a possibility to insert to some laptops an expansion card with additional USB ports. All five cameras can always be connected to a computer using an USB hub if it is not required to record videos using the array.

Another possibility of using five cameras is connecting each camera to a separate computer (e.g. such as Raspberry Pi) responsible for acquiring images. Then, images using Internet interfaces can be transferred to another computer on which stereo matching algorithms are executed.

EBCA is intended for use as a 3D vision system for an autonomous robot. In particular EBCA can be mounted on a robotic arm in an eye-in-hand configuration. Figure 2 visualized sample methods of placing EBCA on an arm. There are applications in which it is necessary to provide 3D vision in the close vicinity to a gripper. In such cases the gripper is always visible by cameras included in EBCA. EBCA with five cameras presented in Fig. 2a provides 3D vision of objects near the gripper and above it. If cameras have narrow fields of view then the configuration with four cameras presented in Fig (b) is more sufficient. Placing a central camera closely to the arm ensures that the area near the gripper is covered by the 3D system. EBCA can also be mounted in such a way that optical axes of cameras are not parallel to the arm. However, it needs to be considered that aiming EBCA towards some direction limits areas visible by the array in the other direction. EBCA can also be used as a 3D vision system of an autonomous robot which is not equipped with a robotic arm.

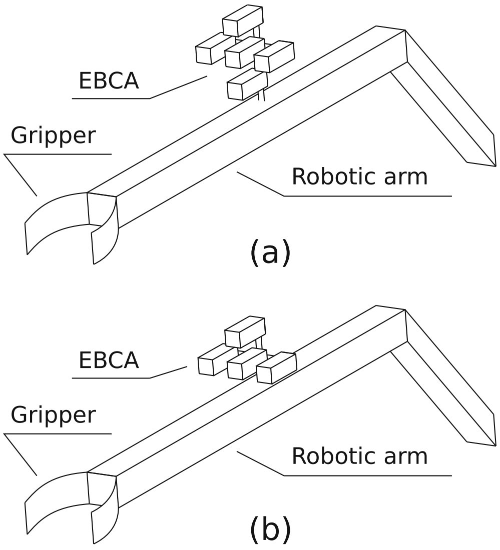

Fig. 2 Sample methods of mounting EBCA on an robotic arm

4 Algorithms for EBCA

Park and Inoue proposed using a set of five cameras that has a form of EBCA, however they also introduced a stereo matching algorithm designed for this set [9]. Their algorithms assigned cameras into two groups. The first group contained cameras located along the horizontal line while the second one contained cameras located along the vertical line. The central camera was included in both of these groups. The algorithm proposed by Park and Inoue is further described in Section 6.1 as is used in the research presented in this paper.

The author of this paper has also developed methods for obtaining disparity maps with the use of EBCA. Exceptions Excluding Merging Method (EEMM) and the Multiple Similar Areas algorithm (MSA) [10, 11] are the most significant ones.

Exceptions Excluding Merging Method obtains a disparity map on the basis of disparities acquired independently from each other by using four stereo cameras considered in EBCA. Every stereo camera consisting of a central camera and a side camera is used to generate a disparity map with the use of some stereo matching algorithm. Results from four stereo cameras taken into account in calculations resemble four measurements of the same quantity. EEMM is a method of processing these four results in order to obtain a disparity map which has a higher quality than constituent disparity maps. The method is based on excluding values of disparities that significantly differ from values acquired from other stereo cameras. Details of EEMM and its performance are described in [11].

Multiple Similar Areas is another algorithm for EBCA introduced by the author of this paper [10]. MSA does not merge disparities acquired from different stereo cameras like EEMM. The algorithm obtains disparities directly from all images obtained from the camera set. The MSA algorithm focuses on identifying monochromatic areas in images. For each point of a central image, the algorithm searches for sequences of points having a similar color. The search is performed simultaneously in all side images in the areas corresponding to the location of the considered points of the central image. This concept leads to poor results in case of having only a pair of cameras but is justified when there are side cameras placed in four different directions from the point of view of the central camera. The algorithm is precisely described in [10].

Considering that EEMM merged results of stereo matching algorithm and MSA is an algorithm dedicated for EBCA, a different kind of algorithms for EBCA is proposed in this paper. These algorithms are based on modifying the cost function that compares similarities between different areas in images. Such a function is used in existing stereo matching algorithms. The function is modified in order to take advantage of five camera set instead of using a pair. Stereo matching algorithms have different characteristics such as execution time and error rates of results. Regardless of their possibilities, the results of existing algorithm for stereo vision can be improved by appropriately modifying their cost function and applying them to EBCA as described in Sections 5 and 6.

5 Structure of Stereo Matching Algorithms

Algorithms that obtain disparity maps by taking advantage of EBCA can be based on matching algorithms designed for stereo cameras. Using these algorithms with EBCA requires modifying them. This subsection describes types of algorithms and their structure in the context of the EBCA usage.

Figure 3 shows types of stereo matching algorithms and their main phases. In general, algorithms are classified as local, global, semi-global and other ones. Input images to all stereo matching algorithms can be subjected to preprocessing such as blurring and the output data can be postprocessed. Let us assume that the left image is the reference one. Points of the disparity map obtained as the result of a stereo matching algorithm corresponds to points of this image. The other image in a stereo pair, called the side image, is used to determine values of disparities of points visible in the reference image. Figure 3 presents for each phase the area of the side image that influences the value of the disparity in the single point of the reference image.

Algorithms of a local type have the simplest structure. The disparity of a point p in the reference image is determined regardless of the entire contents of the reference image and the side image. Points of the reference image affecting the disparity of the point p are those that are located in the vicinity of p. The vicinity is called an aggregating widow [12, 40]. Aggregating windows may have different shapes and sized.

Points in the aggregating window are matched with corresponding points in the side image. The search for corresponding points is performed in a limited area of the side image. The size of the area is determined by input parameters of the stereo matching algorithm, in particular the minimum (D_{min})and maximum (D_{max}) value of disparity accepted by the algorithm.

In Fig. 3 a point is marked with a grey color in the left image in the matching phase of a local algorithm type. The disparity of this point is determined by a grey colored rectangular area placed in the right image. The area consists of points included in all aggregating windows located along a line of a size $D_{max} - D_{min}$. A stereo matching algorithm searches within this area for the point corresponding to a point in the left image.

Corresponding points are identified with the use of a matching cost function. A matching function takes as arguments points within the aggregating window in the reference image and points included in the aggregating

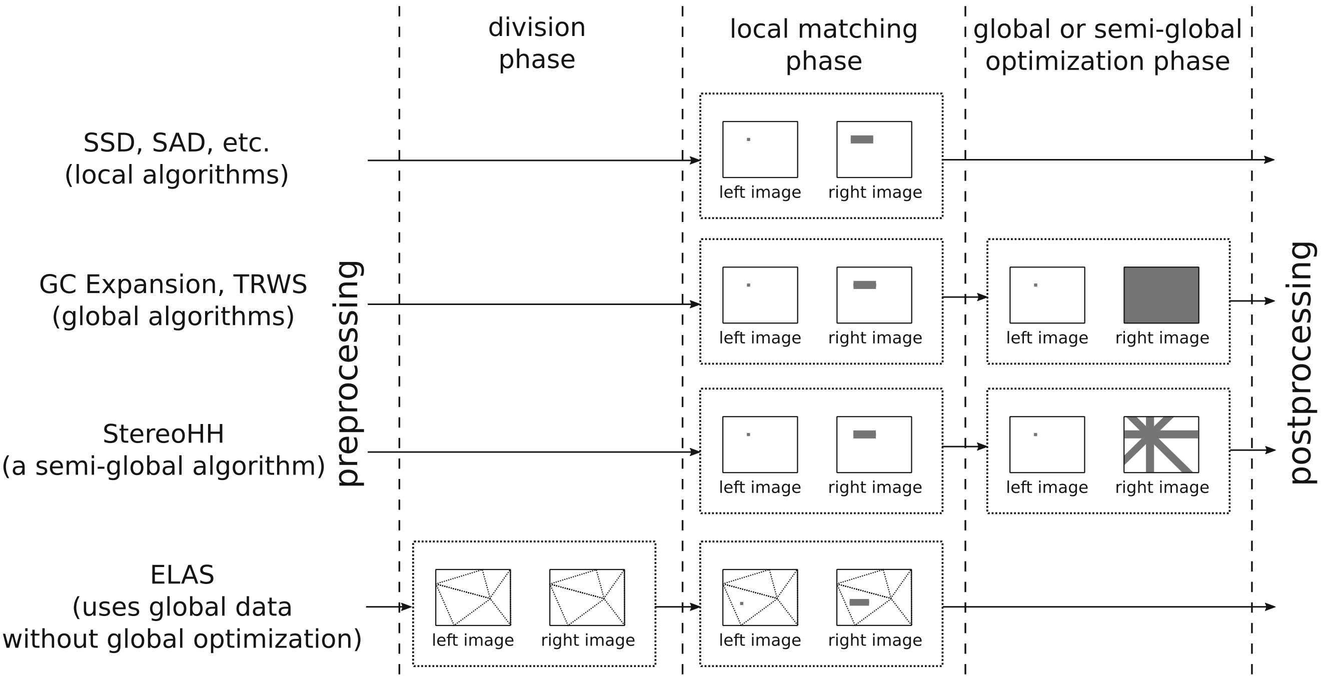
division phase
local matching phase
global or semi-global optimization phase
SSD, SAD, etc. (local algorithms)
GC Expansion, TRWS (global algorithms)
StereoHH (a semi-global algorithm)
ELAS (uses global data without global optimization)
preprocessing
postprocessing
left image
right image

Fig. 3 The structure of stereo matching algorithms

window in the side image. The function returns the level of differences between these areas. Commonly used cost function are Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD) and a matching function proposed by Birchfield and Tomasi [10, 42]. Other matching cost measures are described in [32] and [10]. A matching cost function is modified when EBCA is used instead of a stereo camera. Modification possibilities are described in Section 6. A general formula of a matching cost function for stereo cameras is presented in Eq. 1.

$$c\left(d, \mathbf{p}\right) = m\left(W_0(\mathbf{p}), W_q\left(\mathbf{p} - \mathbf{d}\right)\right) \tag{1}$$

where c is the result of the matching function for the point p and the disparity d, vector $\mathbf{d} = (d, 0)$ refers to disparity in x axis, m is a matching function, W_0 is the set of adjacent points included in the aggregating window in the reference image and W_q is the set of points in the aggregating window from the side image.

The stereo matching algorithm of a local type calculates matching cost within the range of disparities between D_{min} and D_{max}. The disparity producing the lowest value is selected as a resulting disparity for which the match occurred. This operation is presented in Eq. 2.

$$c_{result}\left(d, \mathbf{p}\right) = \underset{d \in [d_{min}, d_{max}]}{\text{argmin}} \; c\left(d, \mathbf{p}\right) \tag{2}$$

There are also stereo matching algorithms of a global type [43] as shown in Fig. 3. This kind of an algorithm is in fact an extended version of a local type of an algorithm. An algorithm of a global type consists of a phase which is a stereo matching performed similarly as in local algorithms. The difference in this step between these two types of algorithms is such that global algorithms do not select disparity by finding the minimum matching cost. In global algorithms, data acquired from local matching is further processed in the phase called global optimization. In this step, the algorithm computes a global minimum with the use of optimization methods. The global optimization causes that each point of a disparity map depends on the entire input stereo pair and, in particular, the entire content of the side image. Therefore, in Fig. 3 the whole right image is marked grey in case of global algorithms.

A variety of optimization methods used for obtaining disparity maps exists including Iterated Conditional Modes (ICM) [44], Graph Cuts [36], Max-product Loopy Belief Propagation (LBP) [45] and Sequential Tree-reweighted Message Passing (TRWS) [46, 47]. These methods minimize an energy function based on Markov Random Fields (MRF) [44].

Figure 3 presents also the structure of a semi-global matching algorithm designed by Heiko Hirschmüller [35]. It will be denoted as StereoHH. The original version of the algorithm considered 16 paths that affect the value of the disparity in a single point. These paths are marked with grey color in the side image in the last phase of the algorithm presented in Fig. 3. Heiko Hirschmüller's algorithm was implemented in the OpenCV library [31]. The implementation called the semi-global block matching algorithm (StereoSGBM) differs in the number of considered paths. Additionally, StereoSGBM include block matching instead of individual pixels.

Phases of both StereoHH and StereoSGBM are similar to the phases of global algorithms based on MRF. These algorithms consist of a local stereo matching phase and a semi-global phase which covers cost aggregation, disparities selection, consistency check, disparity refinement and other procedures. Hirschmüller indicated that matching cost calculated in the local phase should be based on either Mutual Information or the Birchfield-Tomasi metric [35, 42]. Other semi-global matching algorithms also consist of a local phase and an optimization phase.

Andreas Geiger introduced the Efficient Large-Scale Stereo Matching (ELAS) algorithm that is the last type of algorithm presented in Fig. 3 [48]. Its structure significantly differs from the structure of other algorithms. At first, ELAS identifies characteristic points called support points using global data and analyzing all images. However the algorithm is not classified as a global one, because it does not perform a global optimization. ELAS matches support points which corresponds to each other in two different images. Then, the ELAS algorithm performs triangulation of these points. The triangulation results in splitting two images into a set of pairs of triangles. Each pair consists of one fragment from the left image and one from the right image. After this step, the ELAS algorithm performs a phase of local stereo matching of points in corresponding triangle fragments of images. The results of local matching are then further processed in the algorithm.

In general, all stereo matching algorithms have a phase in which matching costs are estimated in the process of local matching. The acquired data is used in algorithms to determine disparities selected to a disparity map obtained as a result of these algorithms. This paper introduces a method for modifying the local phase in order to make it possible for a stereo matching algorithm to take advantage of EBCA. Every algorithm for obtaining disparity maps that have such a phase can be applied to images from EBCA with the use of methods presented in this paper.

6 Adaptation of Matching Algorithms to EBCA

This paper introduces methods of improving disparity maps by modifying the local phase of stereo matching algorithms.

The modification is based on taking advantage of five cameras included in EBCA instead of two cameras from a single stereo camera. Processing five images from EBCA makes it possible to retrieve more data and set the matching cost more precisely than using two images from a stereo camera. The concept of using EBCA instead of a stereo camera is presented in Fig. 4.

Matching cost based on EBCA is determined on the basis of four stereo cameras included in EBCA. These cameras will be marked with C_1, C_2, C_3 and C_4. Subsequent indexes respectively refer to right, top, left and bottom side cameras. There are four stereo matching costs (marked with c_1, c_2, c_3 and c_4) retrieved from each stereo camera from EBCA and there is a compound, resulting matching cost (marked with c_e) which is acquired from these four constituent costs.

Equation 1 from the previous section applies to the case when a cost c_1 is calculated with the use of a single stereo camera in which the side camera is mounted to the right of the central one. Objects in the side camera are then shifted to the left with regard of their locations in a central image. In EBCA, side cameras are placed in different directions from the central camera. When costs c_2, c_3 and c_4 are calculated, Eq. 1 needs to be altered and instead of using a disparity $\mathbf{d} = (d, 0)$ which refers only to horizontal dimension the vector $\mathbf{d_i}$ need to be used such that $\mathbf{d_1} = (d, 0)$, $\mathbf{d_2} = (0, d)$, $\mathbf{d_3} = (-d, 0)$ and $\mathbf{d_4} = (0, -d)$. These values correspond to different directions in which objects are shifted in side images.

Different functions can be used to calculate the resulting matching cost c_e from stereo matching costs c_1, c_2, c_3 and c_4. In the research presented in this paper, three types of functions were considered that were based on

1. Park and Okutomi stereo matching algorithm
2. Selecting a single value from the sorted list of costs
3. Obtaining a composite value derived from the sorted list of costs

6.1 Matching Based on the Park and Okutomi Algorithm

Park and Inoue are the authors of the first research paper describing EBCA (Section 3) together with an algorithm designed for this camera set. The algorithm divides four stereo cameras included in EBCA into two pairs. The first pair consists of cameras located along the vertical line (C_1 and C_3). The second pair are cameras in the horizontal line (C_2 and C_4). The algorithm searched for a match separately for each pair of stereo cameras. It is motivated by the problem with visibility of objects which are not located in the foreground of a viewed scene.

Let us suppose that there are objects O_a and O_b located beside each other and O_a is partly hidden behind O_b from the point of view of a central camera in EBCA. In this case, O_a is more visible from either left or right camera than from the central one. Additionally, O_a is more hidden behind the object O_b from the point of view of the other side camera. Therefore, there is an area visible in the central image which is also visible in only one of side images. Park and Inoue assumed that the algorithm can find matching areas in such cases by analyzing images from side cameras placed on the opposite sides of the central camera. The same rule applies to cameras located both in horizontal axis and vertical one.

In each pair of stereo cameras, the algorithm selects the camera for which the matching cost is lower. Then, the algorithm sums matching costs obtained for two pairs. Park and Inoue executed this kind of matching for different sizes of aggregating windows. As the result the resolution pyramid was obtained which was used to generate a final disparity map. The method of merging stereo matching costs used by Park and Inoue is presented in Eq. 3. This method will be denoted by PaI.

$$c_e(d, \mathbf{p}) = min\,(c_1(d, \mathbf{p}), c_3(d, \mathbf{p})) + min\,(c_2(d, \mathbf{p}), c_4(d, \mathbf{p})) \tag{3}$$

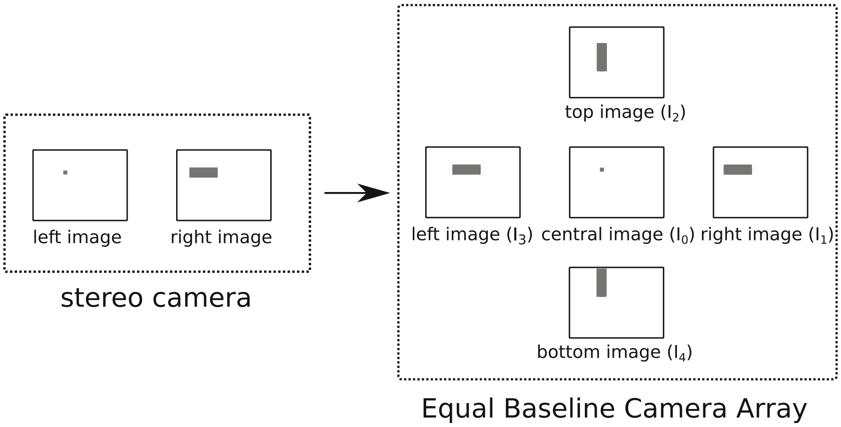

Fig. 4 Applying matching cost functions used for stereo cameras to five cameras included in EBCA

where c_1, c_2, c_3, c_4 are matching costs for the right, top, left and bottom camera.

Park and Inoue used the SSD matching measure. However, their method of merging stereo matching costs can be applied to other matching cost functions and different stereo matching algorithms. This paper introduces such an application. The author of this paper modified local phases of algorithms provided in the OpenCV library and in the Middlebury Stereo Vision Project in order to process images from EBCA using the cost merging method proposed by Park and Inoue. The modification is based on using the formula presented in Eq. 3 as an extension of the formula presented in Eq. 1. The results are presented in Section 9 entitled Experiments.

6.2 Matching Based on a Sorting Matching Costs

There are also other possibilities of obtaining resulting matching cost c_e on the basis of constituent stereo matching costs c_1, c_2, c_3 and c_4 acquired for the same disparity d. Stereo matching costs differ between each other because of differences in identifying corresponding areas in four stereo images obtained from EBCA. The values of the matching costs can be sorted. It results in having a sorted list s_1, s_2, s_3 and s_4, where $s_1 \leq s_2 \leq s_3 \leq s_4$. Sorted values identify the camera with the lowest matching cost and the camera with the greatest one.

In general, low matching cost indicate that two parts of images are similar to each other and they contain a view of the same part of the real object. Therefore, it may seem that selecting the lowest matching cost s_1 is the best method for obtaining the resulting matching cost. However, it does not need to be the best solution. Let us suppose that some part of a real object is visible from the reference camera and n side cameras. In such a case the matching cost should be low in each stereo pair consisting of a side camera with visible part of an object similarly as in case of the stereo camera for which it is the lowest. Therefore, selecting second or third value from a sorted list (i.e. s_2 or s_3) can correctly identify areas of images showing the same object, if $n \geq 2$ or $n \geq 3$, respectively.

Furthermore, matching functions do not always return the lowest value for parts of images that correspond to the same real object. There may be many reasons for obtaining low value of costs in cases when high ones would be more appropriate. One of them is such that a side image may contain a specific area that inappropriately matches with many areas of the reference image. Supposing that this problem occurs in stereo camera x, the cost value of c_x will be excessively low. If camera x is the only affected one, then c_x will be the lowest value in the sorted list of matching costs, i.e s_1 will be equal to c_x. Subsequent values s_1, s_2, s_3 are more resistant to such problems.

The resulting matching cost can be equal to one of values included in the sorted list s_1, s_2, s_3, s_4 as presented in Eq. 4.

$$c_e(d, \mathbf{p}) = s_n(d, \mathbf{p}), n \in [1, 4] \tag{4}$$

where n is the index of a value in the sorted list of costs. For example, if $c_e = s_2$, then the algorithm is dedicated for matching areas which are visible in at least two side cameras. Section 9 presents the results obtained for the different values of n.

The main application of the merging method described in this subsection is EBCA with five cameras. However, the method can be also applied to EBCA consisting of a different number of cameras. In such case, a sorted list of matching costs is also prepared, but its size depends on the number of considered stereo cameras. The resulting cost c_e is equal to one of costs selected from the list for every kind of EBCA.

The merging method based on selecting a single value from a sorted list of costs will be denoted in this paper by a label which has a form Mn/N, where M stands for a merging method, n is an index of a value s_n selected from a sorted list and N is equal to the number of cameras. For example, a method based on selecting the second value when five cameras are used will be denoted by M2/5.

6.3 Matching Based on a Composite Value

The previous subsection described the method for obtaining the resulting matching cost by selecting a single value from a sorted list of costs. This section describes a modification of this method in which the resulting matching cost is equal to a composite value obtained on the basis of sum of at least two stereo matching costs from a sorted list. Adding a certain cost to the sum can either lead to the improvement or the deterioration of the results depending on its usability for obtaining disparity maps. A general formula for calculating resulting cost with the use of this method is presented in Eq. 5.

$$c_e(d, \mathbf{p}) = \sum_{s_n(d,\mathbf{p}) \in S} s_n(d, \mathbf{p}) \tag{5}$$

where S is the set containing sorted values considered in calculations. In particular, c_e can be equal to a sum of s_1 and s_2 which are the lowest values in a sorted list of costs as presented in Eq. 6.

$$c_e(d, \mathbf{p}) = s_1(d, \mathbf{p}) + s_2(d, \mathbf{p}) \tag{6}$$

Equation 7 presents analogous calculations for c_e value equal to a sum of a second and third value from the sorted list.

$$c_e(d, \mathbf{p}) = s_2(d, \mathbf{p}) + s_3(d, \mathbf{p}) \tag{7}$$

These methods of obtaining c_e were tested in the experiments described in this paper. The results are described in Section 9.

Merging methods based on a composite value will be denoted by labels M$n_1, n_2, ..., n_k/N$, where $n_1, n_2, ..., n_k$ are indexes of matching costs included in a sorted list and N is the number of cameras. For example, if five cameras are used then the function presented in Eq. 6 will be marked by M1,2/5 and the function from Eq. 7 will be denoted by M2,3/5.

7 Data Sets

Test data used in the experiments presented in this paper was based on the same images as those which were used in previous research on EBCA [11]. The test data consists of six sets where each set contains five images of a plant taken with the use of EBCA and ground truth representing real values of disparities. The images and the camera set were calibrated using the OpenCV library [31]. A detailed description of these sets is presented in [11]. The total number of points considered in the experiments was equal to 212800.

Figure 5 present a complete data set containing images of a strawberry while Fig. 6 presents central images from remaining five sets.

Images in data sets consist of a matching area for which ground truth is prepared and a margin which is the area located around the matching area. Although the margin extends the size of input images, it does not cause the increase in the number of test points. However, margins cause that algorithms achieve better results for the matching area, because they can analyze a larger vicinity of points included in tests. In previous experiments, different margin sizes were used for different data sets. In the experiments presented in this paper, margins were unified and they were equal to 100 points for all data sets.

8 Quality Metrics

Disparity maps obtained in experiments were evaluated with the use of three quality metrics. The first one is the percentage of bad matching pixels (BMP). BMP is one of the most important metrics used for estimating the quality of disparity maps [32]. Its formula is presented in Eq. 8.

$$BMP = \frac{1}{N}\sum_{\mathbf{x}}(|D_M(\mathbf{x}) - D_T(\mathbf{x})| > Z) \tag{8}$$

where $D_M(\mathbf{x})$ is the disparity of the point $\mathbf{x}$ in the evaluated disparity map, $D_T(\mathbf{x})$ is the correct disparity obtained from ground truth, N is the total number of points and Z is the threshold.

BMP depends on the number of points for which the difference in values of disparities between the disparity map and ground truth is not lower than a certain threshold Z. Such points are considered to be matched incorrectly. Results presented in this paper are calculated for $Z = 2$.

The BMP metric considers points included in ground truth for which it was possible to determine real disparities, because object composed from these points are visible in both images from a stereo camera. Disparity maps contain also points from the background for which determining disparities is impossible [11, 33]. Stereo matching algorithms can either provide such data or produce incorrect values of disparities for areas in the background. The metric called percentage of bad matching pixels in background (BMB) was used in order to estimate the influence of merging methods presented in this paper on points in background. Its formula is presented in Eq. 9.

$$BMB = \frac{1}{N_B}\sum_{\mathbf{x}}(D_M(\mathbf{x}) \neq 0 \wedge D_T(\mathbf{x}) = 0) \tag{9}$$

where N_B is the number of points in the background and other symbols are the same as in Eq. 8.

BMB presents the percentage of points which were inappropriately classified as not background by a stereo matching algorithm. The equation applies to disparity maps in which values of points with undetermined disparities are set to 0.

The third metric considered in the evaluation is the coverage (COV) which is related to points classified as

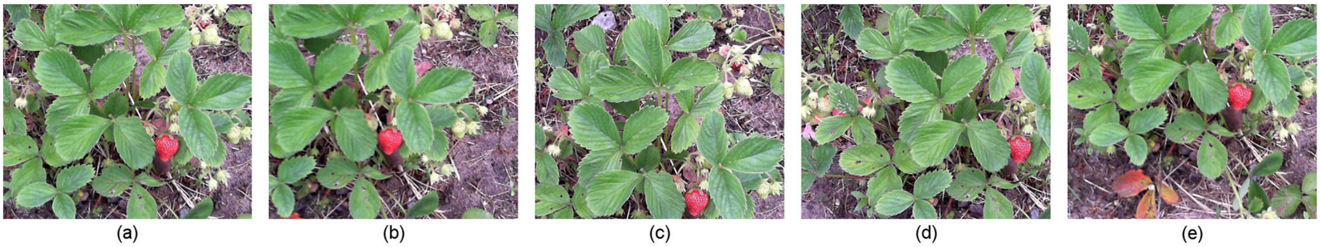

Fig. 5 Images of the strawberry data set (**a**) central **b** right **c** top **d** left **e** bottom

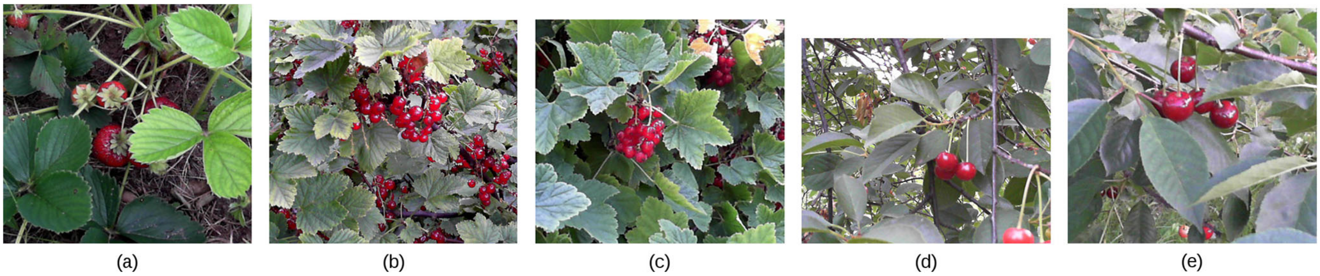

Fig. 6 Central images of five data sets used in the experiments

background [11]. Coverage corresponds to the number of points with disparities included in disparity map as presented in Eq. 10. The COV metric is used to verify whether merging methods cause that stereo matching algorithms provide disparities for more or less points with regard to versions of these algorithms which do not use merging methods.

$$COV = \frac{N_L}{N} \qquad (10)$$

where N_L is the number of points with assigned disparities in a disparity map and N is the total number of points.

9 Experiments

Fig. 8 presents results of different cost merging methods used in EBCA with the number of cameras varying from two to five. The figure presents values of the BMP metric. In case of using five cameras the following merging methods are included in the figure: M1/5, M2/5, M3/5, M1,2/5, M2,3/5 and PaI (using notations described in Section 6. Merging methods M4/5 and M3,4/5 were also tested, however they are not presented in Fig. 8, because their results were worse than the results of presented methods. Similarly, some cost margining methods tested for EBCA with other number of cameras are also not included in Fig. 8. The results presented for EBCA containing four cameras include merging methods M1/4, M2/4 and M1,2/4. In case of EBCA with three cameras the results of methods M1/3 and M1,2/3 are shown. EBCA with two cameras is in fact a stereo camera therefore no cost mergining method can be used. The results for two cameras are denoted by M1/2. Therefore, this case shows results of using a stereo camera.

Experiments were performed using the implementation of GC Expansion and TRWS algorithms provided by Middlebury Stereo Vision Page described in Sections 2.3 and 5 [32, 36, 46, 47]. These algorithms were selected on the basis of previous research concerning EBCA [11]. In previous research, the best results were obtained for these algorithms. Experiments were also executed using the StereoSGBM algorithm available in the OpenCV library [31]. StereoSGBM was included in tests because it is a commonly used algorithm provided in the OpenCV library which is one of the most significant programming libraries used in the field of computer vision [31].

Figure 7 visualizes sample results obtained in the experiments. Subfigures (a)-(d) presents disparity maps obtained for the cherry data set using the StereoSGBM algorithm. Subfigure (e) is an input image corresponding to these maps. Subfigures (f)-(i) presents results of processing redcurrant data set with the use of the TRWS algorithm and the input image corresponding to these maps. In case of both presented data sets subsequent disparity maps show results for merging methods M1/2, M1/3, M1,2/4 and M1,2/5.

Images show that disparity maps obtained with the use of better merging methods are more consistent and contain less errors. In is particularly visible when image Fig. 7d is compared to image Fig. 7a. Contours of leaves areas are more shredded in image Fig. 7a than in image Fig. 7d. Moreover, there are more black parts showing that disparities were not obtained in areas of image Fig. 7a representing leaves than in the same areas in image Fig. 7d. The improvement is also visible in case of images Fig. 7f and i obtained with the use of a different stereo matching algorithms than images Fig. 7a and d. Image Fig. 7f contained bright points which do not represent correct values. The number of these kind of points is reduced in image Fig. 7i.

Figure 8 presents values of the BMP metric obtained for the tested algorithms and average values. The best results were obtained when the TRWS algorithm was used with the M1,2/5 merging method for EBCA consisting of five cameras. The BMP metric was then equal to 12.9%. On average, the M2/5 method produced disparity maps with the lowest BMP equal to 14.09%. It was only 0.03% worse than average results generated for M1,2/5. Considering EBCA with four cameras the best configuration is TRWS with M2/4. The outcome is BMP equal to 17.58%. The M1/3 metric applied to the GC Expansion algorithm leads to the best BMP equal to 18,68% when three cameras are used. As far as results for a pair of cameras are concerned, the best disparity maps were acquired using StereoSGBM with BMP

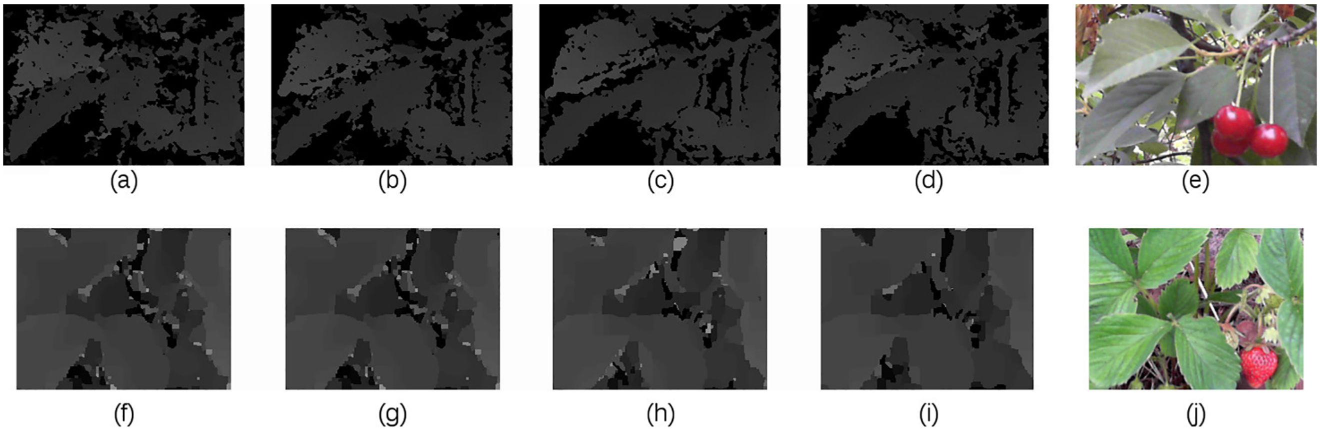

Fig. 7 Sample disparity maps: (**a**)-(**d**) the StereoSGBM algorithm and cherry tree data set, (**f**)-(**i**) the TRWS algorithm and redcurrant data set. The following merging methods are presented: (**a**), (**e**) M1/2; (**b**), (**f**) M1/3; (**c**), (**g**) M1,2/4; (**d**), (**h**) M1,2/5. Subfigures (**e**) and (**j**) show corresponding areas in input images

equal to 20.46%. Taking into account that the best result for EBCA with five cameras was equal to 12.9%, the increase of cameras caused that the number of bad matching points was reduced by 36.95% of its best value for two cameras.

The level of improvement caused by using EBCA is different depending on the stereo matching algorithm. The benefit of using EBCA is the lowest for StereoSGBM. In this case the best result for a configuration with 5 cameras was achieved with the use of the PaI method. BMP improved 21.75% in comparison to its value obtained for two cameras. The GC Expansion using the M2/5 merging method improved 45.56%. The improvement equal to 45.15% was obtained for TRWS using M1,2/5 which is the best merging method for this algorithms. On average, the improvement was equal to 37.49% for the best results. Improvements for the most important configurations are presented in Table 2. Values presented in the table are calculated for each stereo matching algorithm with regard to the results obtained for a stereo camera. Table 2 also presents average results for considered merging methods.

In general, the increase in the number of cameras leads to improvement of results. However, in case of some algorithms, using a greater number of cameras can deteriorate the results if an inappropriate merging method is used. For example, in case of the StereoSGBM algorithm, the results for M1/4 and M1/5 methods are worse than the results for M1/3.

There is no merging method which is the most suitable one regardless of the used matching algorithm and the number of cameras in EBCA. Experiments showed that the best merging methods are: M1,2/5 for TRWS, M2/5 for GC Expansion and PaI for StereoSGBM. On average, M2/5 is the best method, although its results differ only slightly from the results of M1,2/5. Experiments also identified methods that are not suited for any of these algorithms, in particular, methods M1/5 and M1/4. As described in Section 6.2, the first value selected from the sorted list of matching costs is susceptible to errors and it is advisable to take an advantage of other matching costs. In case of using EBCA with 4 cameras, the best method are: M2/4 for GC Expansion, M1,2/4 for TRWS and M1,2/4 for StereoSGBM. If 3 cameras are used, the M1/3 method proved to be the best one.

Disparity maps obtained in the experiments were also evaluated on the basis of the COV metric. Foremost, results of this metric depends on the type of the stereo matching

Fig. 8 Values of the BMP metric acquired for disparity maps obtained with the use of different cost merging methods and stereo matching algorithms

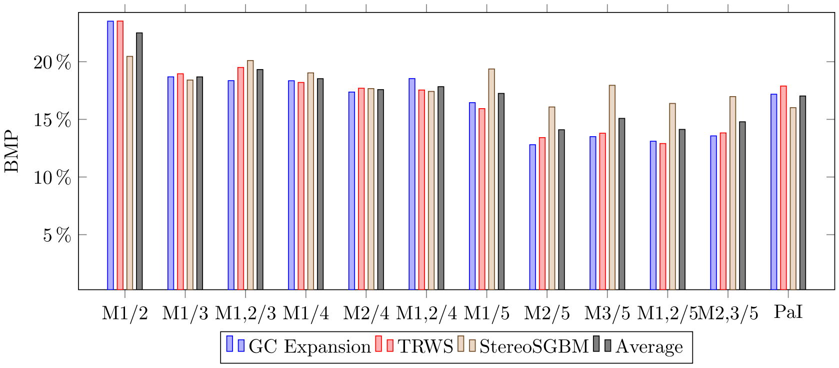

Table 2 Improvements caused by the usage of EBCA and merging methods with regard to results obtained for a stereo camera

	M1/3	M1/4	M1/5	M2/5	M1,2/5	PaI
GC Expansion	20.53 %	21.97 %	30.05 %	45.56 %	44.28 %	26.94 %
TRWS	19.43 %	22.63 %	32.28 %	42.97 %	45.15 %	23.95 %
StereoSGBM	10.03 %	6.99 %	5.34 %	21.45 %	19.95 %	21.75 %
Average	16.66 %	17.2 %	22.56 %	36.66 %	36.46 %	24.21 %

algorithm that was used. Values of COV do not significantly differ from each other for considered merging methods. In case of GC expansion and TRWS algorithms, average values of the COV metric were always above 93.7%. Thus, these algorithms generate disparity maps that have a low rate of unfilled areas for every merging method. For StereoSGBM, the results of COV were in the range between 87,46% and 90,38%. Therefore, merging methods presented in this paper do not change characteristics of stereo matching algorithms in terms of disparity map coverage.

As far as the BMB metric is concerned the influence of merging methods on results of this metric is greater than in case of COV however it is not crucial in both of these metrics. The results of BMB are in between 91.74% and 97.42% for all merging methods with the use of GC Expansion and TRWS algorithms. Such a high value is the consequence of the fact that these algorithms have a high level of COV, thus they provide disparities in areas that should be classified as background. In case of StereoSGBM, the lowest value of BMB equal to 36.94% was obtained for M3/5 and the highest value equal to 50.41% was acquired for M1/4. BMB was equal to 43.86% when a single stereo camera (the M1/2 method) was used. Merging methods M1/3, M1/4 and M1/5 produced a result with a higher value of BMB than M1/2. Therefore, results of BMB indicate, similarly as results of BMP, that selecting the first value from a sorted list of costs is not a suitable method. The best methods selected on the basis of BMP, i.e. M2/5 and M1,2/5, resulted in values of BMB equal 45.99% and 48.5%, respectively. These values are worse than results for M1/2, however the difference with regard to M1/2 is not high. Nevertheless, experiments show that using these merging methods do not lead to better results of the BMB metrics.

Using a greater number of cameras included in EBCA also results in an increase in the processing time. It is particularly important in real-time applications. Algorithms for obtaining disparity maps from stereo images intended to use in real-time are ranked with the use of the KITTI benchmark [33]. The benchmark was prepared for evaluating algorithms applicable for autonomous vehicles. Therefore, the speed of obtaining disparity maps is a crucial parameter in this evaluation. Algorithms used with EBCA are derived from algorithms designed for stereo cameras. The increase in the processing time depends on the number of cameras included in the array, images resolution and the type of used algorithm. In our previous research the processing time using images from EBCA with five cameras was at least four times longer than time required for images from a stereo camera [11]. The algorithm proposed in this paper adapts stereo matching algorithms to EBCA in such a way that only parts of algorithms need to be rerun. The increase in the processing time depends on the type of algorithm.

In case of the StereoSGBM algorithm the disparity map is obtained within 600 ms for a single pair images with the resolution of 1000x800 using Intel Pentium G4560 3.5GHz processor. The calPixelCostBT function which is the part of the algorithm needs to be executed four times more often because of using EBCA with five cameras. The function takes 78ms when there are two input images. Thus, the version of the StereoSGBM algorithm for five camera EBCA produces results within 840 ms. The increase in the processing time is below 40 %. Therefore, if there is a stereo matching algorithm suitable for real-time applications, than its version taking advantage of EBCA can also remain within acceptable time limits. Moreover, methods presented in this paper increase the processing time over 10 times less than previously developed EEMM method requiring running the entire stereo matching algorithm four times when the array with five cameras was used [11].

10 Summary

Using merging methods presented in this paper improves results between 21.75% and 45.56% for stereo matching algorithms GC Expansion, TRWS and StereoSGBM. There is no merging method that is the most suitable for every stereo matching algorithm. The best results are obtained when a method is selected with regard to the algorithm which is used. Experiments also showed that results of some algorithms obtained with the use of EBCA can be better than other algorithms regardless of their performance with a stereo camera. For example, StereoSGBM returned better results for a stereo camera than TRWS, however results of TRWS were better when these two algorithms were applied to EBCA.

The research presented in this paper is particularly important because of its potential applications of EBCA in autonomous robots which needs to collect visual 3D information about their surrounding. EBCA can be used in autonomous cars, drones, underwater robots and any other applications concerning operations in out-door environment such as autonomous robots designed for facilitating workers in construction sites. EBCA can be mounted on a robotic similarly easily as a stereo camera. However EBCA provides higher quality of 3D data with the use of methods presented in this paper.

Our plans for further research concerning EBCA are focused on three areas of development: increasing the number of test data sets, verifying the performance of EBCA in different light conditions and releasing the source code of algorithms for EBCA. We are going to prepare more test data using EBCA presented in this paper and possibly another EBCA consisting of cameras with a greater resolution than the used ones. Moreover, we are planning to evaluate results of using EBCA with different kinds of light sources such as spot or diffused light. These experiments will also cover verifying the influence of intensity of light on the quality of results. The third area of future work is releasing the source code of the application used in our experiments. The program will be released under open source license in order to support developing algorithms for EBCA and using the array in different domains of applications.

References

1. Pérez, L., Rodríguez, İ., Rodríguez, N., Usamentiaga, R., García, D.F.: Robot guidance using machine vision techniques in industrial environments: A comparative review. Sensors **16**(3), 335 (2016)
2. Jang, W., Je, C., Seo, Y., Lee, S.W.: 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)
3. Eichhardt, I., Chetverikov, D., Jankó, Z.: Image-guided tof depth upsampling: a survey. Mach. Vis. Appl. **28**(3), 267–282 (2017)
4. Tse, RO.C., Gold, C., Kidner, D.: Advances in 3D Geoinformation Systems, chapter 3D City Modelling from LIDAR Data, pp. 161–175. Springer, Berlin (2008)
5. Kazmi, W., Foix, S., Alenya, G.: Plant Leaf Imaging Using Time of Flight Camera under Sunlight, Shadow and Room Conditions. In: 2012 IEEE International Symposium On Robotic and Sensors Environments (ROSE), pp. 192–197 (2012)
6. Venkataraman, K., Lelescu, D., Duparré, J., McMahon, A., Molina, G., Chatterjee, P., Mullis, R., Nayar, S.: Picam: An ultra-thin high performance monolithic camera array. ACM Trans. Graph. **32**(6), 166:1–166:13 (2013)
7. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 1, pp. 519–528 (2006)
8. Okutomi, M., Kanade, T.: A multiple-baseline stereo. IEEE Trans. Pattern Anal. Mach. Intell. **15**(4), 353–363 (1993)
9. Park, J.-I., Inoue, S.: Acquisition of sharp depth map from multiple cameras. Signal Process. Image Commun. **14**(1-2), 7–19 (1998)
10. Kaczmarek, A.L.: Improving depth maps of plants by using a set of five cameras. J. Electron. Imaging **24**(2), 023018 (2015)
11. Kaczmarek, A.L.: Stereo vision with equal baseline multiple camera set (ebmcs) for obtaining depth maps of plants. Comput. Electron. Agricult. **135**, 23–37 (2017)
12. Kaczmarek, A.L.: Stereo Camera Upgraded to Equal Baseline Multiple Camera Set (Ebmcs). In: 2017 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), pp. 1–4 (2017)
13. Bi, Z.M., Wang, L.ihui.: Advances in 3d data acquisition and processing for industrial applications. Robot. Comput.-Integr. Manuf. **26**(5), 403–413 (2010)
14. Daneshmand, M., Bilici, O., Bolotnikova, A., Anbarjafari, G.: Medical robots with potential applications in participatory and opportunistic remote sensing: A review. Robot. Auton. Syst. **95**, 160–180 (2017)
15. Ludvigsen, M., Sørensen, A.J.: Towards integrated autonomous underwater operations for ocean mapping and monitoring. Ann. Rev. Control **42**, 145–157 (2016)
16. Jiang, W., Wang, Z.: Calibration of visual model for space manipulator with a hybrid lm-ga algorithm. Mech. Syst. Signal Process. **66-67**, 399–409 (2016)
17. Murphy, R.R., Kravitz, J., Stover, S.L., Shoureshi, R.: Mobile robots in mine rescue and recovery. IEEE Robot. Autom. Mag. **16**(2), 91–103 (2009)
18. Fjerdingen, S.A., Bjerkeng, M., Transeth, A.A., Kyrkjebø, E., Røyrøy, A.: A learning camera platform for remote operations with industrial manipulators. IFAC Proc. Vol. **45**(8), 39–46 (2012). 1st IFAC Workshop on Automatic Control in Offshore Oil and Gas Production
19. Hayashi, S., Shigematsu, K., Yamamoto, S., Kobayashi, K., Kohno, Y., Kamata, J., Kurita, M.: Evaluation of a strawberry-harvesting robot in a field test. Biosyst. Eng. **105**(2), 160–171 (2010)
20. Baeten, J., Donné, K., Boedrij, S., Beckers, W., Claesen, E.: Autonomous fruit picking machine: A robotic apple harvester. In: Laugier, C., Siegwart, R. (eds.) Field and Service Robotics, volume 42 of Springer Tracts in Advanced Robotics, pp. 531–539. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-75404-6_51
21. van Henten, E.J., Hemming, J., van Tuijl, B.A.J., Kornet, J.G., Meuleman, J., Bontsema, J., van Os, E.A.: An autonomous robot for harvesting cucumbers in greenhouses. Auton. Robot. **13**(3), 241–258 (2002)
22. Muscato, G., Prestifilippo, M., Abbate, N., Rizzuto, I.: A prototype of an orange picking robot: past history, the new robot and experimental results. Indust. Robot: Int. J. **32**(2), 128–138 (2005)
23. Wan, W., Lu, F., Wu, Z., Harada, K.: Teaching robots to do object assembly using multi-modal 3d vision. Neurocomputing **259**(Supplement C), 85–93 (2017). Multimodal Media Data Understanding and Analytics
24. Sanchez-Lopez, J.R., Marin-Hernandez, A., Palacios-Hernandez, E.R., Rios-Figueroa, H.V., Marin-urias, L.F.: A real-time 3d

pose based visual servoing implementation for an autonomous mobile robot manipulator, vol. 7. 3rd Iberoamerican Conference on Electronics Engineering and Computer Science CIIECC (2013)

25. Shin, B.-S., Mou, X., Mou, W., Wang, H.: Vision-based navigation of an unmanned surface vehicle with object detection and tracking abilities. Mach. Vis. Appl. **29**(1), 95–112 (2018)
26. Jia, B., Chen, J., Zhang, K.: Recursive drivable road detection with shadows based on two-camera systems. Mach. Vis. Appl. **28**(5), 509–523 (2017)
27. Lin, L., Song, Y., Yang, Y., Feng, H., Cheng, Y., Pan, H.: Computer vision system r&d for east articulated maintenance arm robot. Fusion Eng. Des. **100**, 254–259 (2015)
28. Plebe, A., Grasso, G.: Localization of spherical fruits for robotic harvesting. Mach. Vis. Appl. **13**(2), 70–79 (2001)
29. Palli, G., Moriello, L., Scarcia, U., Melchiorri, C.: An underwater robotic gripper with embedded force/torque wrist sensor. IFAC-PapersOnLine **50**(1), 11209–11214 (2017). 20th IFAC World Congress
30. Chéné, Y., Rousseau, D., Lucidarme, P., Bertheloot, J., Caffier, V., Morel, P., Belin, É., Chapeau-Blondeau, F.: On the use of depth camera for 3d phenotyping of entire plants. Comput. Electron. Agricult. **82**, 122–127 (2012)
31. Bradski, G.R., Opencv, A.K. Learning Gary Rost, 1st edn. O'Reilly Media, Inc., Newton (2008)
32. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis. **47**(1), 7–42 (2002). Microsoft Research Technical Report MSR-TR-2001-81, November 2001
33. Menze, M., Heipke, C., Geiger, A.: Joint 3D Estimation of Vehicles and Scene Flow. In: ISPRS Workshop on Image Sequence Analysis (ISA) (2015)
34. Menze, M., Heipke, C., Geiger, A.: Object scene flow ISPRS Journal of Photogrammetry and Remote Sensing (JPRS) (2018)
35. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 328–341 (2008)
36. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
37. Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E., Barth, A., Adams, A., Horowitz, ., Levoy, M.: High performance imaging using large camera arrays. In: ACM SIGGRAPH 2005 Papers, SIGGRAPH'05, pp. 765–776. ACM, New York (2005)
38. Fehrman, B., McGough, J.: Depth Mapping Using a Low-Cost Camera Array. In: 2014 Southwest Symposium on Image Analysis and Interpretation, pp. 101–104 (2014)
39. Fehrman, B., McGough, J.: Handling Occlusion with an Inexpensive Array of Cameras. In: 2014 Southwest Symposium on Image Analysis and Interpretation, pp. 105–108 (2014)
40. Kaczmarek, A.L.: Influence of aggregating window size on disparity maps obtained from equal baseline multiple camera set (ebmcs) Choraś, R.S. (ed.) Springer International Publishing, Cham (2017)
41. Microsoft lifecam studio product information. http://download.microsoft.com/download/0/9/5/0952776D-7A26-40E1-80C4-76D73FC729DF/TDS_LifeCamStudio.pdf. Accessed: 2019-09-08
42. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. IEEE Trans. Pattern Anal. Mach. Intell. **20**(4), 401–406 (1998)
43. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 1068–1080 (2008)
44. Besag, J.: On the statistical analysis of dirty pictures. J. R. Stat. Soc. Ser. B (Methodol.) **48**(3), 259–302 (1986)
45. Tappen, M.F., Freeman, W.T.: Comparison of Graph Cuts with Belief Propagation for Stereo, Using Identical Mrf Parameters. In: 2003. Proceedings. Ninth IEEE International Conference On Computer Vision, pp. 900–906 Vol.2 (2003)
46. Wainwright, M.J., Jaakkola, T.S., Willsky, A.S.: Map estimation via agreement on trees: message-passing and linear programming. IEEE Trans. Inf. Theory **51**(11), 3697–3717 (2005)
47. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. **28**(10), 1568–1583 (2006)
48. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) Computer Vision ACCV 2010, 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part I, volume 6492 of Lecture Notes in Computer Science, pp. 25–38. Springer, Berlin (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Adam L. Kaczmarek received the Ph.D. degree in informatics in 2012 from Gdansk University of Technology, Gdansk, Poland. He also received the M. Sc., Eng. degrees in informatics from this university. Mr. Kaczmarek worked as an Intern at Intel Technology Poland from 2005 to 2007. Since 2007 he is working at Gdansk University of Technology. Currently, he is an Assistant Professor. His research interests include camera arrays, stereo cameras and 3D vision.