robotics

MDPI

Review

Heterogeneous Map Merging: State of the Art

Ilze Andersone

Department of Artificial Intelligence and Systems Engineering, Riga Technical University, Riga LV-1658, Latvia; ilze.andersone@rtu.lv; Tel.: +371-29386371

Received: 24 April 2019; Accepted: 15 August 2019; Published: 20 August 2019

Abstract: Multi-robot mapping and environment modeling have several advantages that make it an attractive alternative to the mapping with a single robot: faster exploration, higher fault tolerance, richer data due to different sensors being used by different systems. However, the environment modeling with several robotic systems operating in the same area causes problems of higher-order—acquired knowledge fusion and synchronization over time, revealing the same environment properties using different sensors with different technical specifications. While the existing robot map and environment model merging techniques allow merging certain homogeneous maps, the possibility to use sensors of different physical nature and different mapping algorithms is limited. The resulting maps from robots with different specifications are heterogeneous, and even though some research on how to merge fundamentally different maps exists, it is limited to specific applications. This research reviews the state of the art in homogeneous and heterogeneous map merging and illustrates the main research challenges in the area. Six factors are identified that influence the outcome of map merging: (1) robotic platform hardware configurations, (2) map representation types, (3) mapping algorithms, (4) shared information between robots, (5) relative positioning information, (6) resulting global maps.

Keywords: robotic mapping; map merging; multi-robot systems; heterogeneous robots

1. Introduction

The cooperative mapping is an important task for any multi-robot system that requires a model of the environment. Data sharing to achieve quicker creation of the environment map is critical in time-sensitive situations, such as rescue operations, but is also useful in everyday situations to improve overall multi-robot system performance.

However, environment modeling with several robotic systems operating in the same area simultaneously is not a simple task. A detailed review of approaches and challenges in the multi-robot SLAM (Simultaneous Mapping and Localization) can be found in [1]. If multiple robots are used for the exploration of the environment, their collected information should be fused into a global map that can then be used for navigation. Many methods have been developed that deal with the map merging, and they generally address one or both of the following problems:

1. The map fusion. If the correspondences between the maps are at least approximately known, the map fusion methods are used to merge the data from both maps [2–9]. The correspondences between the robot maps can be acquired in several different ways: they may be known from the start [2], acquired from mutual observations [3], or calculated by map matching [10–22].
2. The map matching. The methods that deal with the map matching [10–22] offer solutions to find the correspondences between two robot maps, when they are unobtainable by other means.

When the robots with different specifications are used in the same environment, the resulting maps are heterogeneous (see Figure 1 for example). In this paper, two maps are considered to be heterogeneous in respect to one another, if their representations of the same environment part are different, and the differences are caused at least partially by the robot mapping system (such as map format, map scale or used sensors).

While the existing robot map and environment model merging techniques allow us to merge certain homogeneous map types [10,13,23], the possibility to use robots that produce heterogeneous maps is still limited and relatively little researched. Heterogeneous robot map merging is a novel research field and currently there are few solutions even for the most common map types. However, the rising importance of robotic technologies in both industry and household motivates the development of more universal and cooperative systems in the future [24]. Therefore this work serves as both a review and as a problem definition article to highlight research problems in heterogeneous map merging.

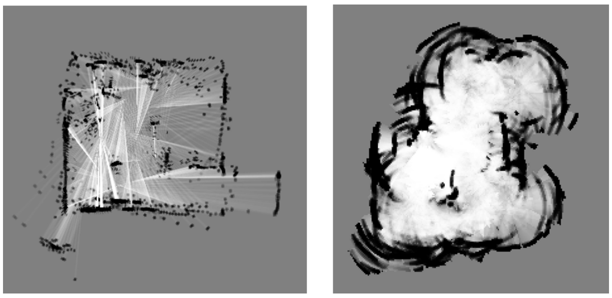

Figure 1. Heterogeneous map examples with the same room is depicted in both pictures (maps the courtesy of collegues in Riga Technical university): (**left**) Occupancy grid map created with a non-rotating LIDAR. (**right**) Occupancy grid map created with ultrasound sensor.

The goal of this paper is to review the state of the art in the homogeneous and heterogeneous map merging research areas and to determine the main challenges in this field. An important part is to determine limitations and how much the heterogenous map merging task can be abstracted and where only specific solutions will work. Compared to the review in [1], this review focuses specifically on robot map merging.

2. Homogeneous Map Merging

To give the context of the heterogeneous map merging problem, the homogeneous map merging will be reviewed first.

2.1. The Map Fusion Methods

The map fusion methods are applied, when the correspondences between both maps are at least approximately known. The knowledge of the relative positions simplifies the map merging process significantly, but there are still challenges to be addressed. The relative positions are often known only approximately, and the fusion method should account for the uncertainty [2,8,9,25]. There is also a possibility that the maps themselves are inconsistent and require modifications [6].

2.1.1. Metric Grid Map Fusion Methods

The early multi-robot map fusion methods are adaptations of existing single robot mapping methods for the multi-robot case [2,3,25,26]. In the simplest case [2] it is assumed that all robots start the mapping close to each other, have significant overlaps in their initial range scans or their initial relative positions are determined with other means.

An example of an extended single robot mapping method is the work by Thrun in [2]. Thrun uses the combination of maximum likelihood estimation and particle filters to create consistent metric grid maps from multiple robots. All involved robots share and update a single global map while maintaining their own individual pose estimates. To be successful, this approach requires the storage and sharing of action/observation sequences.

The assumption of known robot relative starting positions requires that all robots start the mapping in one place or determine their relative positioning with other means. This is a serious limitation, which can be softened by introducing the concept of the robot meeting or rendezvous [3,25]. In this scenario, the robots map their environment independently until they observe another robot and determine their relative positions.

One example of such an approach is the work by Howard [3], which uses Rao–Blackwellized particle filter mapping and places no restrictions on the robot starting positions. Instead, it is assumed that the robots will meet at some point during the mapping and accurately determine the relative positions. Starting from this point, all the future data from both robots are merged into one common global map. The past data is incorporated by creating and running two virtual robots backward in time. For this approach to be successful, all the past data must be stored.

Adluru et al. [17] propose a method, where the global map construction is treated as a mapping with a single virtual robot performing particle filter based SLAM, where the sensor readings from the individual robots are merged in a global map by using sequential Monte Carlo estimation. The odometry information of the virtual robot is acquired by matching local maps with the current global map, and the matching is guided by shape information extracted from the maps (corner features).

Carlone et al. [25] offer to extend the Rao–Blackwellized particle filter based mapping algorithm by incorporating the data from other robots during the encounters and considering the uncertainty in relative position estimations. Every time a robot encounters another robot, they exchange the data acquired since the last meeting. Then the data is transformed using the reference frame calculated from the communicated data and the relative position measurements. Finally, the received data is filtered and incorporated in the robot's map as if it was collected by the robot itself.

The data fusion method by Carlone et al. [25] has a significant difference from the previously mentioned works [2,3,17] in that it does not create one global map, but instead allows the robots to continue the exploration independently. The requirement of constant data exchange and centralized map computation is limiting in practice, therefore many researchers have worked on distributed solutions, where each involved robot produces its own map and is not dependent on the continued communication or a centralized computing node (metric map examples include [25,26]).

The methods of metric grid map fusion often incorporate the data from the other map as if it was just another measurement. An example of a simple map fusion algorithm is employed by Burgard et al. in [26] to fuse occupancy grid map cell probabilities from multiple maps (see Equations (1) and (2)). Here, the $P(occ^i_{x,y})$ represents the probability that the location corresponding to the grid cell $< x, y >$ is occupied by an obstacle in the map of robot i. This merging method can be used at any time, but requires the knowledge of the relative robot positions with a certainty.

$$P(occ_{x,y}) = \frac{odds_{x,y}}{1 + odds_{x,y}} \tag{1}$$

$$odds_{x,y} = \prod_{i=1}^{n} \frac{P(occ^i_{x,y})}{1 - P(occ^i_{x,y})}. \tag{2}$$

Another commonly used sensor data fusion method is the log odds update [27]. In this approach, additionally to the occupancy $P(occ^t_{x,y})$ each cell stores an additional log odds value $L(x, y)$, which represents the measurement history. The probability values are updated with a Bayesian filter (Equations (3)–(5)).

$$P(x, y) = 1 - \frac{1}{1 + exp(L(x, y))} \tag{3}$$

$$L(x, y|z_{1:t}) = L(x, y|z_{1:t-1}) + L(x, y|z_t) \tag{4}$$

$$L(x,y) = log\left(\frac{P(x,y)}{1-P(x,y)}\right) \quad (5)$$

The summary of the reviewed metric grid map fusion methods is shown in Table 1. Abbreviations: PF—particle filter; EM—expectation maximization; RBPF—Rao–Blackwellized particle filter.

Table 1. The summary of the reviewed metric grid map fusion methods.

Method	Map Type	Algorithm	Relative Pos.	Global Map	Shared Data	Hardware
Thrun [2]	Metric grid	PF-EM	Known init.	Shared	All	High communication requirements.
Howard [3]	Metric grid	RBPF	Known (meet.)	Shared	All	Must support relative position estimation. High communication requirements.
Adluru et al. [17]	Metric grid	RBPF	Matching	Shared	Map only	-
Carlone et al. [25]	Metric grid	RBPF	Known (meet.)	Separate	All	Must support relative position estimation.
Burgard et al. [26]	Metric grid	-	Known	Separate	Map only	-

2.1.2. Feature Map Fusion Methods

The main difference of the metric grid map and the feature map fusion is the procedure of how the two maps are integrated when the transformation is found. Metric grid map merging generally treats the maps as rigid bodies and integrates the new map cell occupancy values with the mapping algorithm. The feature maps, on the other hand, are feature lists that must be fused with as few unnecessary duplicates as possible.

Similarily to the cooperative metric grid map fusion [2,3], the global feature maps can be created by extending existing mapping methods for multiple robots. An example of one such research is done by Fenwick [4], where extended Kalman filter (EKF) point feature SLAM is extended for the multi-robot case. This method is very similar to the single robot EKF SLAM, and the main difference is the addition of all robot positions to the state vector. Such implementation requires that the initial relative positions of the robots are known.

Rodriguez-Losada et al. [7] do not assume that robots work in a common coordinate system. The relative locations are known, but each robot creates its own local line feature map with EKF SLAM. These local maps are at some point fused in a global map with the possibility to correct inconsistencies by updating robot positions. A special attention is paid to the matching of the features and ensuring that the global map does not diverge due to the features not being correctly associated. This is done by adding constraints between the features observed in the last step and all other features, and then inverting the innovation covariance matrix with full pivot Gauss–Jordan elimination to avoid numerical errors due to introduced constraints.

The work by Zhou and Roumeliotis [8] is an example of an EKF multi-robot mapping approach, where the robot relative positions are discovered during mutual observations later in the mapping. Zhou and Roumeliotis [8] address the EKF mapping with the corner features by identifying feature duplicates with a fast neighborhood matching algorithm. The distance of the features from the rendezvous point is used as one of the parameters when considering matching, to account for the relative pose estimation inaccuracies. After the fusion the robots continue to update the global map together.

Thrun and Liu in [28] focus on another mapping algorithm and fuse the data of the multi-robot SLAM with a sparse extended information filter. This mapping method has two advantages over the EKF-based mapping methods: the additivity and locality properties. The additivity property allows multiple robots to fuse their data incrementally, and the data update is tolerant of the network latencies. The locality property allows the robots to update only their own pose and detected landmarks, which means that each robot can continue to maintain a separate local map. The properties of the SEIF SLAM reduce the map fusion problem to the concatenation of the corresponding information states and information matrices and the incorporation of the correspondences with the collapsing operation [28].

Apart from the EKF SLAM and SEIF SLAM, the particle filter based SLAM methods are commonly used both in single robot and multi-robot SLAM [9]. Ozkucur and Akin [9] merge the point maps created with particle filter based Fast SLAM algorithm. Given that the particle-based SLAM returns a set of particles, where each particle represents one map, the first problem that must be addressed is which particles to use for the merging. Ozkucur and Akin use the weighted mean of the map estimation of particles with the importance of weight, and the other robot's mean map estimation is integrated into the map of each particle. To find the duplicate features, the nearest neighbor method is used.

A related problem to the feature map fusion is the reduction of the feature maps by only keeping the relevant features. Such sparsification is a necessity to support a long-term multi-robot feature mapping system. Several authors have addressed the problem of the long-term mapping [29–31] by discarding part of the collected data and only keeping the relevant features. The problem has been addressed by only adding new features after they have been observed for some time and confirmed not to be erroneous observations [29], by clustering the features and keeping only some representatives [30], or summarizing the map and describing it in a simplified way [31].

The summary of the reviewed feature map fusion methods is shown in Table 2. Abbreviations: EKF—extended Kalman filter; SEIF—sparse extended information filter; RBPF—Rao–Blackwellized particle filter.

Table 2. The summary of the reviewed feature map fusion methods.

Method	Map Type	Algorithm	Relative Pos.	Global Map	Shared Data	Hardware
Fenwick [4]	Feature	EKF	Known init.	Shared	All	-
Rodriguez-Losada et al. [7]	Feature	EKF	Known (meet.)	Shared	Map only	Must support relative position estimation.
Zhou et al. [8]	Feature	EKF	Known (meet.)	Shared	Map only	Must support relative position estimation.
Thrun Liu. [28]	Feature	SEIF	Unknown	Separate	Map only	Tolerant of the network latencies.
Ozkucur et al. [9]	Feature	RBPF	Known (meet.)	Separate	Map only	Must support relative position estimation.

2.1.3. Graph-Based Map Fusion Methods

The graph-based map fusion methods differ from the metric grid and feature map fusion in that the maps are fused on the graph level. The graph-based maps considered here are topological maps [5], topological-metric maps [32] and pose graphs [6,33,34].

Dudek et al. [5] propose a topological multi-robot exploration method that assumes that the robots start in one place and later meet at one common point. There the graph nodes that were visited by both robots are fused while the robots are not moving. To avoid duplicate nodes, a node that is found in only one map, is marked with a physical movable marker. The other robot then visits all the nodes in its own map, and only if it does not discover the marker, the node is added to the global map.

Chang et al. [32] use hybrid metric-topological maps. Each node in this map type is represented by a local occupancy grid, and the nodes are connected by edges that can be traversed. This map representation allows us to fuse two maps simply by adding and optimizing an edge between two maps after the robots mutually observe each other.

Work by Bonanni et al. in [6] assumes a known relation between two maps and relies on any existing method to find the correspondence. Their research focuses on another problem: how to correct the errors when the maps created by individual robots are inconsistent, not mergeable by a linear transformation and only resulting maps without full source data are available. Even with known relative positions, this problem is not trivial, and Bonanni et al. address this by treating the maps as deformable bodies and searching for the nonlinear transformation between two pose graphs. Pose graphs are normally an output of graph SLAM-based algorithms [35], but the authors provide an algorithm to extract pose graphs from occupancy grid maps.

Cunningham et al. in [33,34] propose a decentralized cooperative map sharing and fusion approach for pose graph maps constructed with DDF-SAM (decentralized Ddta fusion smoothing and mapping). To reduce the communication load, the local map summarization is implemented before transferring it to the other robots. The early version of DDF-SAM [33] enforced the maintenance of separate local and neighborhood maps for each robot, which resulted in two incomplete maps of the environment. The later improved version [34] introduced the augmented local system map, which blends both local and neighborhood information and acts as a replacement for the two maps.

The summary of the reviewed graph-based map fusion methods is shown in Table 3. Abbreviations: EM—expectation maximization; RBPF—Rao–Blackwellized particle filter.

Table 3. The summary of the reviewed graph-based map fusion methods.

Method	Map Type	Algorithm	Relative Pos.	Global Map	Shared Data	Hardware
Dudek et al. [5]	Topological	Marker based	Known (meet.)	Separate	Map only	Requires physical markers. Low sensor requirements.
Chang et al. [32]	Top./Metric	RBPF-EM	Known (meet.)	Separate	Map only	The map optimization is a high cost process.
Bonanni et al. [6]	Pose graphs	Graph-based	Known	Separate	Map only	-
Cunnigham et al. [33,34]	Pose graphs	Graph-based	Known	Separate	Map only	High computational scalability. Low communication bandwidth.

2.1.4. Three-Dimensional (3D) Map Fusion Methods

The methods listed in the previous sections merge two-dimensional (2D) maps, but in the recent years three-dimensional (3D) grid maps have become commonly used. Although the 3D maps discussed in this section are metric maps (voxel grids, octrees, point clouds) [36–43], there are many commonalities in the merging of 3D maps, which is why they are discussed together in this review.

The map fusion task of 3D maps is generally performed in two steps:

1. Some version of 3D iterative closest point (ICT) [44] or other algorithm is used to refine the transformation [37,39].
2. The map data is fused based on the acquired more accurate transformation. The implementation of this step depends of the used map representation.

The simplest 3D map fusion case is the fusion of discretized maps, e.g., voxel grids or octrees. Voxel grid maps are an extension of occupancy grid maps in 3D space, where each voxel represents the cell occupancy probability. The octree is a tree-based representation of the occupied, free and unknown space, which when compared to the voxel maps, requires less memory and allows us to update the map more efficiently [45].

The fusion of the voxel grid data is straightforward when the transformation between the maps is known, and is very similar to the occupancy grid cell updates. One example of voxel grid data fusion is performed by Michael et al. in [37], where they merge the 3D voxel grid maps from manually operated ground and aerial vehicles. The ground vehicle generates most of the map as a sparse 3D voxel grid (10 cm resolution), and the aerial vehicle is transported along and only maps the inaccessible places to increase the map coverage. It is noted that due to the technical limitations of the aerial vehicle, it abstracts the environment as a 2.5D elevation map. The map fusion is performed by registering both maps via an initialization point near the aerial vehicle take-off location, and further refining the merge with a version of the iterative closest point (ICP) algorithm [46]. The maps are fused by using the multi-volume occupancy grid model, which employs the hit and miss model to provide a probabilistic measure of the cell occupancy [47].

Yue et al. [40] fuse the data of two voxel grid maps by using the log-odds based probability update [27].

Jessup et al. [38,39] provide a method to merge octree based 3D voxel grid maps. Early work by Jessup et al. [38] discusses how to fuse the data from two octree maps if the transformation between the maps is known with certainty from the observations between the two robots. The authors offer the solutions for four distinct fusion cases:

1. The new data is from an area not on the map. In this case a new leaf is added to an octree.
2. The new data is in a mapped area of the same level. In this case the value of the leaf node is updated. The node's probability is changed with the log-odds probability update.
3. The new data is in a mapped area of higher resolution. In this case lower level leaf nodes are added to the node in the mapped area and updated.
4. The new data is in a mapped area of lower resolution. In this case lower level leaf nodes are added to the new data node, updated and then integrated onto the map.

The later research [39] updates the method by acknowledging that an exact transformation is hard to acquire and instead operates on the assumption of approximate relative positions. The uncertainty regarding the relative robot positions is addressed by transforming the octree map to a point cloud and refining the initial transformation with the 3D ICP algorithm [44].

The 3D point cloud representation differs from the voxel grids and octree maps in that they only represent the occupied spaces. Additionally, unlike discrete maps, the point clouds can grow indefinitely with new measurements or data from other robots, if they are not downsampled in some way. The unbounded growth of the somewhat similar feature maps is usually reduced by finding the duplicate features and fusing them [7–9,28], but the points of the 3D point clouds are not features and remain in the final fused map. To avoid the unbounded growth of a 3D point cloud, various resampling methods have been developed to reduce the point cloud size [48–50].

A different type of the 3D map fusion is performed, when the maps are represented as pose graphs, which is a common output of graph-based SLAM. In such case the map fusion is usually performed by adding a new set of constraints that expresses the relative positioning between the nodes of both maps [36,41,42].

In the approach by Schuster et al. [41] the robots create point cloud submaps that are matched by marker-based visual robot detections (if available) and a similarity search within their geometric structure with CSHOT 3D feature descriptor [51]. The 6D transformation acquired from the matching is then refined by applying the ICP algorithm on the full point cloud and the pose graph constraints are accordingly updated.

Mohanarajah et al. [36] treat the fusion of multiple key-frame pose graphs as an optimization problem of a single pose graph. The matching of key-frames is a background process and, if a match between two key-frames from separate maps is found, then the smallest map is integrated into the largest and the optimization continues as normally.

Bonanni et al. [42] search for the constraints between the 3D point cloud pose graphs by performing a thorough matching between the two maps. They focus on deformed input maps and delay the addition of new constraints until a sufficiently large area around the match supports the merging hypothesis.

The summary of the reviewed 3D map fusion methods is shown in Table 4. Abbreviations: PC PG—point cloud pose graph. RBPF—Rao–Blackwellized particle filter.

Table 4. The summary of the reviewed 3D map fusion methods.

Method	Map Type	Algorithm	Relative Pos.	Global Map	Shared Data	Hardware
Michael et al. [37]	3D voxel	Scan matching	Known	Separate	Map only	Employs ground and aerial vehicles. Considers computational limitations.
Yue et al. [40]	3D voxel	RBPF	Matching	Separate	Map only	Considers memory and bandwith limitations.
Jessup et al. [38,39]	Octree	Octree mapping	Known	Separate	Map only	Octree representation requires less memory and bandwidth than the voxel grids.
Schuster et al. [41]	PC PG	Graph SLAM	Known	Shared	Map only	Requires robot marker detections.
Mohanarajah et al. [36]	Key-frame PG	Graph SLAM	Matching	Shared	Map only	Considers the computation limitations by performing background matching in the cloud.
Bonanni et al. [42]	PC PG	Graph SLAM	Matching	Separate	Map only	The fusion is an offline process.

2.2. Map Matching Methods

A completely new problem is introduced in map merging, if the relative positions of the robots are unknown. Without the position information, the map overlaps must be found by map matching. This problem is made harder by the fact, that it is generally unknown if such an overlap even exists. Given that the correctness of solution itself is unknown and often there are several valid hypotheses, most map matching methods assume consistent local maps and focus on finding the best match. When the match is found, the maps can be fused by using any compatible map fusion algorithm.

2.2.1. Metric Grid Map Matching Methods

One prominent group of methods addressing metric grid merging with unknown positions makes an assumption that the merging will be performed when the robots are somewhere in the other robot's explored area. In the method proposed by Ko, Konolige and others [13,14] the robots use the latest sensor readings from other robots to localize them in their own maps. In [13] the feature importance for successful localization is assessed (corner, junction and door features were manually extracted). Ref. [14] addresses the localization correctness confirmation problem by organizing robot meetings at designated points. Similar research to [13,14] is done by Liu et al. [15], where a virtual robot is created to localize itself in other robot's map with Monte Carlo localization. This virtual robot simulates driving around the other robot's map, and the measurements are used to localize it in the robot's own map. This approach, when compared to [13,14] does not require the robots to be in each other's map to find transformation, but the measurement simulation must be possible. The hypotheses proposed by the virtual robot localization are used to arrange a meeting between both robots, similar to [14].

An alternative approach is to treat the merging problem as a search for transformation between two local maps and evaluating the transformation results. Birk and Carpin [10] propose to search the transformation between two occupancy grid maps by rotating and translating them and then evaluating the results. The search is performed by a random walk algorithm and guided by image similarity metric. In theory, this method is able to find an optimal solution even with small overlaps (given infinite time), if the correct parameters are chosen, but in practice it is slow and scales poorly to large maps. Several other authors have offered upgrades to this method [52,53]. Ma et al. [52] uses a genetic algorithm for search instead of a random walk to improve the convergence speed of the algorithm. Li and Nashasibi [53] also use a genetic algorithm to search for transformation, but additionally propose a new improved metric to guide the search that excludes the need for parameter characterizing weight of overlap between two maps.

Although the transformation search methods work well on unstructured environments and small overlaps between the metric grid maps, and in general are the most domain-independent map merging methods, they generally take minutes to find good solutions and do not scale well with the size of the maps due to the large search space. There are methods that use brute force (such as extensive use of GPU [54] or cloud computing [36]) to speed up the search of state space, but most approaches do not rely on the existence of high computing capability and use additional information or features to guide the search [16–22].

In some cases, an additional information must be collected during the mapping, as there is no way to extract it later. One such example is work by Ho and Newman [16] that collects image sequences during mapping and associates them with the occupancy grid maps. Image subsequences are then matched to find the overlap between two robot maps. It is impossible to acquire image subsequence information later, if it is not done during the mapping.

Most approaches, however, extract features from occupancy grid maps without the need for additional information collection during mapping, making them more universal when compared to [16]. Adluru et al. [17] propose a method, where the global map construction is treated as a mapping with a single virtual robot performing particle filter based SLAM, where the sensor readings from individual robots are merged in a global map by using sequential Monte Carlo estimation. Odometry information of the virtual robot is acquired by matching local maps with the current global map, and the matching is guided by shape information extracted from the maps (corner features). Alnounou et al. [18] uses the Hough transform to extract line segments and circles from the occupancy grid maps. The line segments and circles are stored in a feature list, and feature matching is used to find transformation. Feature matching is also used by Blanco et al. [19] (various interest point detectors—SIFT, Harris points, salient points with Kanade–Lucas–Tomasi method and SURF). All these approaches do not require the collection of specific information during the mapping, but the occupancy grids must contain the relevant features (corners [17], lines and circles [18] or interest points [19]) for the map matching to be successful.

Some methods transform the map into different representations that have useful properties. These methods can still be considered feature extraction and matching methods, but they require the map transformation to another format. One such example is Carpin in [20], who uses Hough transform to compute the map rotations in a deterministic way by aligning Hough spectra and searching for maximums. The assumption is made that the robot maps contain features (lines or curves) that can be parameterized. The cross correlation between the spectra returns the rotation hypothesis, and the rotated maps are then projected on translation axis and cross-correlated again to find translations. Saeedi et al. [21] use a similar approach by using Hough transform to separately find the rotation and then the translation between the two maps. Their method improves the previous work by Carpin [20] with a more robust search for translations. Instead of relying on map projections on the translation axis, the Saeedi et al. method use Hough images of already rotated maps, where special geometric shapes called Hough peaks are extracted and used to find translation between the maps. This approach has the advantage of being able to handle a smaller map overlaps. Lee et al. [22] extract sinograms by Radon transform and identifies salient features within sinograms (a structure that contains features, such as directionality of lines). The transformation is then found with particle swarm optimization.

The summary of the reviewed metric map matching methods is shown in Table 5. Abbreviations: RBPF—Rao–Blackwellized particle filter.

Table 5. The summary of the reviewed metric map matching methods.

Method	Map Type	Algorithm	Relative Pos.	Global Map	Shared Data	Hardware
Ko et al. [14]	Metric grid	-	Known	Separate	Map only	-
Konolige et al. [13]	Grid/features	-	Known	Separate	Map only	-
Liu et al. [15]	Metric grid	-	Known	Separate	Map only	-
Birk Carpin [10]	Metric grid	-	Known	Separate	Map only	Requires high computational capabilities.
Ma et al. [52]	Metric grid	-	Known	Separate	Map only	Requires high computational capabilities.
Li Nashasibi [53]	Metric grid	-	Known	Separate	Map only	Requires high computational capabilities.
Ho Newman [16]	Grid/images	-	Known	Separate	Map only	Sensors must support image collection.
Adluru et al. [17]	Metric grid	RBPF	Known	Separate	Map only	-
Alnounou et al. [18]	Metric grid	-	Known	Separate	Map only	-
Blanco et al. [19]	Metric grid	-	Known	Separate	Map only	-
Carpin [20]	Metric grid	-	Known	Separate	Map only	-
Saeedi et al. [21]	Metric grid	-	Known	Separate	Map only	-
Lee et al. [22]	Metric grid	-	Known	Separate	Map only	-

2.2.2. Feature Map Matching Methods

Feature map merging methods, where the robot relative positions remain unknown, generally find the map transformations by matching features. The feature matching approaches are similar to those used in the metric grid map matching with features [16–19] and mostly differ by the used feature types and their characteristics. Some commonly used feature matching algorithms are RANSAC, SVD (singular value decomposition), iterative closest point (ICP) search and improved iterative closest point (ImpICP) search [55].

One example that only deals with the map matching is an approach proposed by Ballesta et al. [55]. The algoritm merges 3D Harris point maps in a two-dimensional transformation space (robots are only able to move in a single plane, therefore vertical transformation dimension can be ignored) with various feature matching algorithms to compare their performance (RANSAC being the most efficient for the studied case). Robots use Rao Blackwellized particle filter for mapping, similar to Ozkucur and Akin in [9], but they only consider the matching of the most probable particle from each map.

Thrun and Liu in [28] provide a feature matching method in the framework of SEIF (Sparse Extended Information Filter) mapping that not only matches the pairs of features in both maps, but also takes into account the lack of features. The feature matching is performed with finding triplets of features and searching for similar local configurations on the other map. The matches of triplets serve as a starting hypothesis for the fusion process.

Another example of the feature map matching is the method by Lakaemper et al. [56], who use shape analysis and similarity to find the common parts of polyline feature maps. The maps are fused using perceptual grouping, which is a custom-made solution for polyline maps.

An interesting problem that is addressed by Dinnissen et al. [57] is the choice of the merging time and method. They use reinforcement learning to create and train a model that helps the robots to determine whether they should merge the maps and which method to use based on their current particle filter states and sensor observations. If successfully trained, such model can determine, when the maps can be merged with the feature matching methods, when the grid matching methods should be used and when the merging is not recommended.

The summary of the reviewed feature map matching methods is shown in Table 6. Abbreviations: RBPF—Rao–Blackwellized particle filter. SEIF—sparse extended information filter.

Table 6. The summary of the reviewed feature map matching methods.

Method	Map Type	Algorithm	Relative Pos.	Global Map	Shared Data	Hardware
Ballesta et al. [55]	Feature map	RBPF	Unknown	Separate	Map only	-
Lakaemper et al. [56]	Feature map	Custom	Unknown	Separate	Map only	-
Thrun Liu. [28]	Feature map	SEIF	Unknown	Separate	Map only	Tolerant of the network latencies.
Dinnissen et al. [57]	Metric/feature	RBPF	Unknown	Separate	Map only	-

2.2.3. Graph-Based Map Matching Methods

Compared to the metric grid maps and feature maps, the topological maps are rarely used independently as the only map representation due to their high abstraction level. Nonetheless, there are several merging methods dedicated especially to topological maps [11,12].

Map merging method developed by Dedeoglu and Sukhatme [11] uses single vertex matches to find possible transformations between two topological graphs. Huang and Beevers [12] base their matching algorithm in the graph theory and use multiple vertex matches to compute the transformation between the two maps.

Methods that specifically match pose graphs or hybrid metric-topological maps are rarely encountered.

The summary of the reviewed graph-based map matching methods is shown in Table 7.

Table 7. The summary of the reviewed graph-based map matching methods.

Method	Map Type	Algorithm	Relative Pos.	Global Map	Shared Data	Hardware
Dedeoglu, Sukhatme [11]	Topological	-	Unknown	Separate	Map only	-
Huang, Beevers [12]	Topological	-	Unknown	Separate	Map only	-

2.2.4. Three-Dimensional (3D) Map Matching Methods

The main difference of 3D map matching and fusion, when compared to the 2D map case, is its high dimensionality, which is why memory [39,40], bandwidth [36,40] and processing [36,37,42] requirements are more often explicitly addressed in 3D map merging research. Otherwise, the existing approaches to 3D map matching are similar to their 2D counterparts, but reduce the computational complexity by either using structural features [40], matching submaps [42], or both [36,41].

One such method, proposed by Yue et al [40], does not require known relative transformation. The algorithm reduces the transformation search space by extracting structural edges (large change in local curvature) from the voxel grid and using edge matching to guide the search for the transformation. Additionally, to the structural information, the local voxel information is used to refine the result.

Besides the extraction of structural features, pose graph-based methods are often used in 3D map matching [36,42]. These methods represent the map as pose graphs connected with deformable constraints and associated with a submap. This representation has two main advantages: (1) it reduces the size of the matched maps (only submaps are matched) and (2) allows nonlinear transformation between two maps as both individual submap matches and pose graphs can be optimized.

One example of 3D point cloud matching is a work by Bonanni et al. [42], who propose a map merging approach specifically designed to deal with distorted pose graph maps. The pose graph representation is chosen, because it allows us to deal with nonlinear transformations between two maps and allows us to develop merging methods that improve the quality of the global map. The submap point clouds are matched with the NICP algorithm [58] taking into account the graph topology to reduce the search space. It is noted by the authors that the point cloud matching time takes up a significant portion of the execution time and should be sped up with some appearance-based matching strategy. The main emphasis in this work is put on submap matching strategies and alignment quality evaluation.

Work by Mohanarajah et al. [36] pays special attention to the bandwidth and computational costs of 3D mapping and map merging. The submaps are represented as pose graph associated key-frames, on which SURF key-point matching is performed by using RANSAC. Key-frames are a subset of frames that summarize the full frame set. This representation is chosen to reduce the bandwidth requirements, and the robot map optimization and map merging are mostly done in cloud.

Dube et al. [59] offer a centralized approach to 3D pose graph map creation. The central node optimizes the map based on incremental sparse pose-graph optimization by using sequential and place recognition constraints. The local submaps are matched by extracting 3D segments with SegMatch algorithm introduced in [60].

It should be noted that the structural feature matching is a key to successful and real-time 3D map matching, as noted in [42]. Besides the already mentioned feature extraction methods [36,40,59] there are many other methods that can be used for the matching purpose, some of which are line extraction [43], plane extraction [61], 3D segment e or 3D model approximation [62].

The summary of the reviewed 3D map matching methods is shown in Table 8. Abbreviations: PC PG—point cloud pose graph.

Table 8. The summary of the reviewed 3D map matching methods.

Method	Map Type	Algorithm	Relative Pos.	Global Map	Shared Data	Hardware
Yue et al [40]	Voxel grid	-	Unknown	Separate	Map only	Addresses bandwith limitations.
Bonanni et al. [42]	PC PG	Graph-based	Unknown	Separate	Map only	Requires significant processing capabilities.
Mohanarajah et al. [36]	Key-point PG	Graph-based	Unknown	Separate	Map only	Addresses bandwith limitations. Utilizes cloud computing.
Dube et al. [59]	PC PG	Custom	Matching	Shared	Map only	Designed to work online with LIDAR sensors.

2.3. *A summary of the Homogeneous Map Merging Methods*

The reviewed map merging methods are summarized in the Table 9 by two parameters: the map representation type and the map merging type (fusion or matching). The main ideas and challenges of each group are presented along with references to the specific methods that address these problems.

Table 9. The summary of the homogeneous map merging methods.

Method Type	Commentary
Metric grid map fusion	The methods deal mainly with the data fusion and map optimization [2,3,17,25–27].
Metric grid map matching	The researchers most often address the problem of the map matching and assume that the data from the other map will be treated as if it was just another measurement [10,16–22,52,53]. The matching is usually performed in one of four ways: (1) using localization techniques [13–15]; (2) as a heuristic search in the transformation space [10,52,53]; (3) through feature extraction and matching [16–19]; or (4) using the map transformations to other representations, e.g., Hough spectrum [20,21].
Feature map fusion	Most methods deal with the fusion of the features in the context of specific mapping algorithm (EKF mapping [4,7,8], Particle filter based mapping [9], SEIF [28]). Special attention is paid to the discovery of duplicate features and avoiding the map divergence.
Feature map matching	Most methods in this category address both the feature map matching and the map fusion problem [28,55,56]. The features are most often matched by using some RANSAC, SVD (singular value decomposition) or iterative closest point (ICP) search algorithm version [55].
Graph-based map fusion	One group of methods deal with the the fusion of graph nodes while avoiding duplicates (topological and topological-metric maps) [5,32]. Pose graph fusion addresses the addition strategies of new constraints between the maps [6,33,34]
Graph-based map matching	The methods use graph matching methods to find the correspondence between the graphs [11,12]. The main difference from feature matching is the observation of the connectivity constraints between the maps.
3D map fusion	Most methods use some version of 3D iterative closest point (ICP) algorithm to refine the transformation [37–39,41]. Volumetric maps are transformed to point clouds to make ICP algorithms applicable [37].
3D map matching	The methods in this group generally search for the transformation, which is then usually refined in a similar way to the map fusion [37–39,41]. The computational complexity is reduced by either using structural features [40,59] matching submaps [42], or both [36,41].

3. Map Merging Influence Factors

Six important factors that influence the map merging were identified during the review of the map merging that should also be considered when addressing the heterogeneous map merging case. These factors are: (1) robot hardware, (2) map representation, (3) mapping algorithms, (4) shared data, (5) relative positions, (6) global maps.

The relations of these six factors are shown in Figure 2, where solid arrows represent mandatory relations and dashed arrows—optional relations. Each individual robot is designed to fulfill a specific purpose (mainly environment mapping in the context of this review), and is equipped with a proper hardware to fulfill this task. The map representation and the mapping algorithm are chosen according to the purpose of the map, but are limited by the robot hardware configurations. Every map merging algorithm requires the map matching part, which is influenced by map representations, shared data, knowledge about relative positions and sometimes the mapping algorithms. If the map fusion is performed, it is influenced by the map representation, mapping algorithm and shared data. The global maps are always produced by the merging process (either fused map/s or just correspondence between two maps).

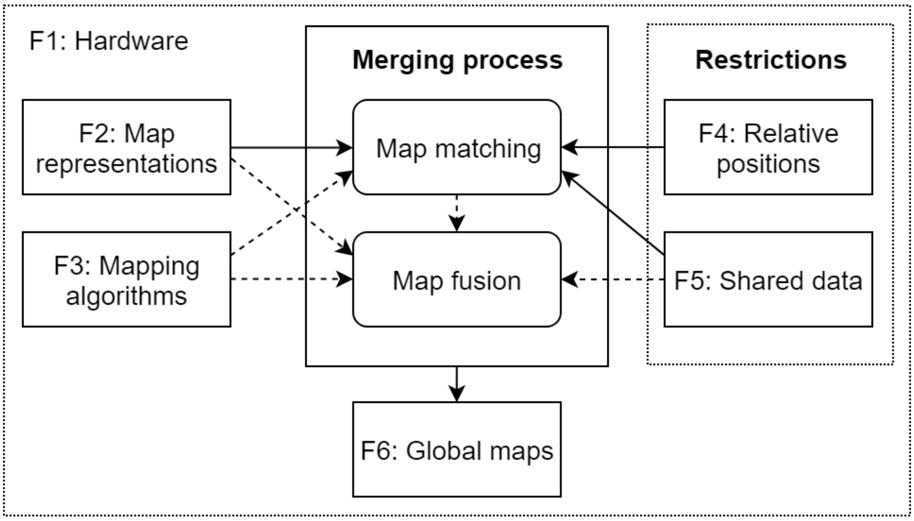

Figure 2. The relationships of the map merging factors.

Every map merging algorithm must take into account the hardware, map representation and mapping algorithm as well as the restrictions (shared data and relative positions) and the desired output (resulting global maps):

1. Robot hardware. Does the robot hardware support map merging? Is it possible to exchange the data, is the processing capability adequate, do sensors support the acquisition of the necessary data (e.g., relative positions of the robots)? These considerations must be taken into account at least indirectly through other factors (through the map representation, relative positioning information, shared data).
2. Map representation. How are the maps represented and can they be matched? Can they be merged?
3. Mapping algorithms. How will the map data from the other robot be integrated in the robot's map?
4. Shared data. Are there any restrictions for shared data between the two robots (map data, full sensor data)?
5. Relative positions. Is the information about the relative positions available? If yes, when and how can it be acquired, how reliable is it?
6. Global maps. If the two maps can be matched, then how will the global map be handled? Will it be merged in one global map, will each robot incorporate the other robot's map data in its own map, will a hybrid map containing both maps be created?

As will be illustrated in subsequent sections, in several cases heterogeneous maps provide significant additional challenges when compared to their homogeneous merging counterparts.

3.1. Map Representation

One of the most obvious factors that differentiates various map merging methods is the map representation, therefore in this review the map merging methods will be divided in four main sections:

1. Metric maps (see Figure 3a) [2,3,6,10,13–20,52–54]. These maps describe the geometric properties of the environment. Occupancy grids are the most common metric grid map type, and they represent the map as arrays, where each cell's value shows the probability that the corresponding area in the environment is free or occupied with obstacle [63].
2. Feature maps (see Figure 3b) [4,7–9,55,56]. In these maps information about the environment is represented as a feature list, where each feature is described with a location and parameters, if they are required for the particular feature type. Features can be points (for example, trees or furniture legs), lines (for example, walls or furniture sides) or other objects. When compared with metric maps, feature maps generally require less computational resources and memory, but they often represent the environment incompletely and do not represent the free space of the environment.
3. Graph-based maps (see Figure 3c) [5,6,11,12,42]. These maps represent the environment as a graph, where the nodes represent the environment locations and the edges are paths or constraints between these locations. For topological maps, the link connecting two locations shows that the robot can move between these two places without traversing any other significant locations. Topological maps lack the geometrical information density of metric maps, but require relatively low memory and significantly simplify the path planning task [64]. Pose graph and hybrid-topological maps are also included in this category.
4. Three-dimensional (3D) maps [36–43]. A common 3D map type is a point cloud map. 3D data can also be represented as discretized volumetric pixel (voxel) grids. However, voxel grids are memory inefficient, and in practice abstractions (Multi-Level Surface maps [65], 2.5-dimensional elevation maps [66]) or octree map compression [38,39,67]) are used.

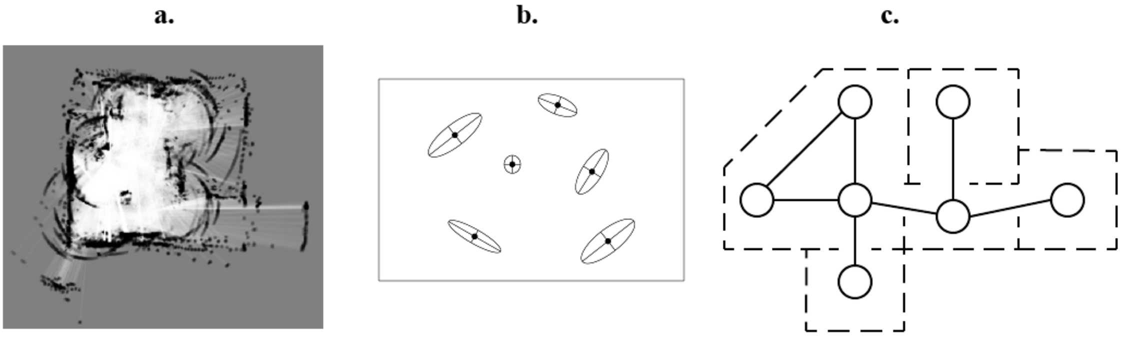

Figure 3. Map type examples: (**a**) occupancy grid map. (**b**) Point feature map. (**c**) Topological map.

Besides the map types listed above many other map types exist (for example, image-based maps [68–70] and manifold maps [71]), but they are used comparatively rarely for robot navigation.

It must be mentioned that often the map types listed above are used together and are not necessarily mutually exclusive, for example, metric maps are used together with feature maps [72,73] or topological maps [32,64,74], or topological maps are combined with feature maps [75]. Any map type may also be supplemented with semantic information [76] or object data [77].

Another important aspect to consider is the purpose of the map. The map types differ significantly not only between the groups listed above, but also within them. Metric grid maps have different resolutions (scale), feature maps represent various features, topological maps contain different locations, 3D point clouds have different sparsity, etc. The map representation is chosen with the purpose of the robot in mind—driverless car navigation [31] has other map representation requirements than the semantic object recognition [77] or the rescue operations [37] It is especially important when merging heterogeneous maps, for while the fusion of homogeneous maps usually produce a map of comparable quality to the originals, it may not always be the case in the heterogeneous map merging.

3.2. Robot Hardware Configuration

Robot hardware influences the mapping algorithms and map representations that the individual robot is capable of using. There are several main hardware categories, which play a role in robot mapping and consequently in map merging:

- Sensor configurations. Sensor configuration determines the environment characteristics the robots are able to detect. Internal and external sensor combination directly influence the types of maps robot is capable of creating, mapping algorithms that can be applied and the accuracy of created maps. Homogeneous map merging approaches generally assume that map differences due to different quality sensors are insignificant—both maps are assumed to have the same quality and have the same weight in merging, but this is not the case for heterogeneous map merging, where different sensors are often the source of map differences. Sensor configuration also impacts whether the robots are able to estimate the position of another robot during an encounter and how accurate this estimate is.
- Communication hardware. It is assumed that all robots involved in mapping are capable of data transfer and reception, otherwise, the map data exchange is fundamentally impossible. Communication channel bandwidth influences the data amount that can be transferred and may limit both homogeneous and heterogeneous map merging possibility [1]. Some existing solutions to the bandwidth limitations are periodic data transfer [54], choosing memory-efficient map representations (for example, octree based representation of 3D maps [38,39,45]) and map compression [40,54].
- Processing capabilities. Processing requirements for different sensor configurations and mapping algorithms significantly differ. Map merging itself can also be computationally expensive and if it is impossible to delegate this task to a more capable team member, a robot with low processing capabilities may be unable to benefit from an improved map. Sharing of processing capabilities has been studied by many researchers, with one approach having robots delegate tasks to more capable team members [68] or cloud [36] and the other creating computing clusters [78] to solve complex tasks. In the recent years solutions have appeared that use the efficient parallel processing capabilities of GPU for multi-robot related tasks, for example, submap matching [36,54]
- Available memory. Memory determines the map size and resolution limitations and also the stored and received data amount. Memory limitations are especially important when considering 3D maps, and 3D octree representation is mainly motivated by memory, bandwidth and processing capability limitations of the robots [38,39,45]. In some cases the available memory allows us to store more data than the robots can process or transfer, and it is possible to acquire higher quality maps after the end of mapping [79].

3.3. Shared Information

Based on stored data, communication channel bandwidth and willingness to share information, different data amount may be shared between robots:

- Maps only. Only the current map is shared with the other robot. Sharing the map requires comparatively little communication channel bandwidth and is therefore one of the more common shared information types [10,12,20,23,80–82].

- All relevant map data. All control and sensor data is shared with the other robot [2,3]. Full data sharing has the benefit of integrating sensor measurements in the map directly and could be especially useful for heterogeneous robots, but is rarely used due to the heavy communication channel load.

3.4. Relative Position Information

Map merging difficulty is influenced greatly by the knowledge of relative robot positioning both for homogeneous and heterogeneous maps. There are two main cases starting from the easiest to most difficult:

- Known positions. The relative positions of robots are initially known or found out during mapping. The mapping can be done by the robots cooperatively updating the global map [4] or each robot can operate as an independent entity and periodically merge their maps [2,3,6–9]. A widely research problem with approximately known positions is the refinement of the transformation [2,3,6–9,37–39,41].
- Unknown positions (relative positions completely unknown, existence of common map part unknown) [10,13–22,40,52–57]. The solution space, where transformation of two maps must be found, is generally huge and hard to find without some feature identification in maps. The problem is that overlap between maps is unknown and may not exist. This case is especially hard for heterogeneous maps due to the possible lack of common and easily identifiable features.

3.5. Mapping Algorithms

The mapping algorithm is an important factor in the map merging due to the different data requirements and produced output. Depending on the used mapping algorithm, some merging methods are influenced significantly because of differences in map representation. In the reviewed works three prominent SLAM approaches were most often used:

- EKF based SLAM. Extended Kalman filter (EKF)-based SLAM used in [4,7,8] represents both robot position and map features as state vectors with associated uncertainty.
- Particle filter based SLAM. Particle filter based SLAM [2,9,17,55] represents robot map with a set of particles, where each particle contains a hypothesis about robot position and a separate map. In the map merging regarding particle filters, a decision must be made how to handle the particle merging. Ballesta et al. [55] merges only the most probable particles, Ozkucur et al. [9] merges the estimated weighted average map of one robot with all the particles of the other robot, and Adluru et al [17] creates a virtual robot, which treats the data from all involved robots as sensor data.
- Graph based SLAM. Graph based SLAM methods [6,36,41,42] represent the map as pose graphs connected with deformable constraints and associated with a submap. The deformable graph like representation allows for nonlinear transformation between two maps even for grid maps, which normally use linear transformations [6], but they are most widely used in 3D point cloud merging [36,41,42].

Additionally, data availability and shared information can significantly influence the map merging result for compatible mapping algorithms. If not only resulting map, but also control and sensor data is available, then, if similar sensors are used and relative positions are known, the other robot gains the opportunity to incorporate this data directly in their maps without the need for map merging [3].

3.6. Resulting Global Maps

In the homogeneous map merging two map fusion results are the most prevalent:

- Shared global map, which the robots update collectively [2–4,7,8,17,36,41,59]
- Separate maps, which can be updated by each robot separately even after the merging. These maps can usually be acquired asymmetrically meaning that each robot has a different map [6,9,10,13,14,18,20].

It makes sense that given different map representations of two robots, the general heterogeneous map merging result should be not one common, but two different maps, where each is represented in the format used by involved robots. However, when heterogeneous maps are fused, another possibility becomes an option:

- Hybrid map incorporates the matching result of both maps not by fusing the data, but by putting another layer atop the existing map (e.g., creating a grid-appearance map [83]).

4. Heterogeneous Map Merging Overview

As stated in the introduction, the heterogeneous map merging field is in a relatively early development stage and few researchers have addressed this problem. The majority of the map merging research focuses on homogeneous robot teams that produce similar maps (some of the most widely cited examples are [10,12,20]). Even if a heterogeneous robot team is used, often an assumption is made that their produced maps do not differ in a significant way and are readily mergeable [37,84,85].

Not all factors or their aspects will always be included in the comparison of the heterogeneous map merging methods (see Table 10 for details). The shared information is omitted in comparison, because relatively few of the reviewed homogeneous map merging methods exchange all collected data (out of all reviewed homogeneous map merging works only some methods [2,3] do this). No reviewed heterogeneous map merging methods assume that all observation/action data is available for both robots.

Table 10. Updated heterogeneous map merging factors.

Factor	Values	Commentary
Map representation	Metric grid maps Feature maps Graph-based maps 3D maps Other maps	There is no single best map representation as each has benefits and shortcomings. Additionally to the general map type, other differences may be present: scale, quality or sparsity.
Positions	Known Unknown	Methods that work with unknown relative positions are generally more universal. If the merging method works for unknown positions, it can also be used for known positions if an appropriate data fusion method is proposed.
Global map	Shared Separate Hybrid	Global map handling determines whether the output is the the same global map for both robots that (shared), different global map for each robot (separate) or the same map for both robots that differs from original map representations (hybrid)
Hardware	LIDAR Camera Other	Only the sensors will be considered in this group (if relevant), as they are one of the most important sources of heterogeneity in the heterogeneous map merging.
Mapping algorithm	EKF SLAM Particle filter based SLAM Graph-based SLAM Other	The mapping algorithms will be considered when they are an important part of the heterogeneity

4.1. Metric Map Merging

The most prominent work in heterogeneous map merging has been performed regarding metric grid map matching (more specifically occupancy grid maps). Topal et al. in [23] propose the first method that can match different scaled heterogeneous metric grid maps not only theoretically, but is specifically intended to do so. To merge different scaled occupancy grids SIFT (scale-invariant feature transform) features are extracted from both maps and used to find a transformation by using nearest neighbor algorithms with minimum Euclidean distance. Several other works have also later addressed different scale occupancy grid merging with different methods [81,82,86].

Ma et al. [86] and Ferrao et al. [82] also use SIFT features to find common key-points in the maps with some differences in how map transformation is found and optimized. Ma et al. [86] uses the random sample consensus (RANSAC) algorithm to find the initial transformation between maps and then optimizes it by solving an objective function based on non-common areas of the maps with trimmed and scaling iterative closest point (TsICP) algorithm. Ferrao et al. [82] addresses the problem similar to [86] with some slight differences in transformation computation.

Park et al. in [81] offer another approach to solve the problem of different resolution occupancy grid map merging. For the merging specific environment features are required: the maps should have at least three separate rectangular space features. Such spaces are common in indoor environments, but rare in outdoor environments.

Shahbandi and Magnusson [87] address the merging of different scale and quality occupancy grids. They use region segmentation and alignment to operate at a high level of abstraction and make the approach more robust to the dissimilarity of maps. This also allows us to treat the scaling parameter as just another transformation parameter additionally to translations and rotation. The main drawback of the method is the computation time, which, as the authors admit, is too high for real-time applications. Similar to the method by Park et al. in [81], it is also limited to mostly indoor environments as it must include distinct regions. The research is expanded in the work by Shahbandi et al. [88], where a method for nonlinear transformation between occupancy grid maps is proposed. Initial alignment provided by a decomposition-based technique is optimized by transforming occupancy grid maps to gradient maps and then using them to find the optimal nonlinear transformation.

Several methods address only different scale grid map merging, but do not account for significantly different levels of noise [23,81,82,86]. There is one method, which is robust against noise and dissimilarities between maps, but it is limited by computation time and environment characteristics [87]. All these methods are capable of yielding separate global maps if the merging is performed twice—once for each map with regards to the other map. When the different scale occupancy grids are matched by any proposed method [23,81,82,86] and the transformation is found, the fusion can be performed by using any metric grid map fusion approach discussed in the homogeneous mapping review section [25–27].

Another research direction in the heterogeneous map matching is the matching of the robot map with some kind of prior map (building plan, sketch, CAD plan) [89,90].

Mielle et al. [90] offer an approach for the matching of sketch maps and occupancy grid maps. They explore the idea that due to their representation of spatial configuration the human-drawn sketches are useful as a prior map for mapping despite their poor accuracy. To find the match between a sketch and a robot map the Voronoi graphs are extracted from both structures and matched with the error-tolerant Neuhaus and Bunke's algorithm for matching planar graphs [91]. This approach cannot be directly applied for real-life SLAM, and Mielle et al. have continued the research direction by developing a new approach [92] that implements a version of graph-based SLAM, which supports the incorporation of the information from an approximate prior. Instead of human-drawn sketches, emergency maps are used in this research. Normal distribution transform (NDT) is used as a map representation, and maps are matched by extracting and using corner features. Unlike the original research [90], this method allows us to not only match the maps, but also to integrate the prior's data in the robot's map.

Boniardi et al. [89] uses the graph-based SLAM to build a pose graph map that is consistent with a prior: CAD floor plan. The CAD map lacks important information about the environment (the room contents), but is useful as a prior for the building wall configurations. The initial location of the robot in the CAD map is assumed to be known.

Another prominent direction in the heterogeneous metric map merging is the matching of maps from different sensors for the localization [93–95] or map building purposes [96].

Caselitz et al. [95] provide a method for the localization of the robot equipped with a monocular camera in a dense 3D point cloud map constructed with the LIDAR. The monocular camera provides a sparse 3D point set with the ORB SLAM algorithm, and this point set is then aligned with the ICP algorithm. A good guess of an initial estimation is assumed in this work. It must be noted that this method does not provide a map, but only performs continuous localization in an existing map representation.

Gawel et al. [93] extends upon the work by Caselitz et al. [95] by removing the constraint of known initial correspondences. To find the match between the LIDAR point cloud map and the sparse vision keypoint set, they first reduce the density of the LIDAR map to be comparable with the sparse vision point map. They then employ three different structural descriptors (Boxli, 3D Gestalt and NBLD) to find the match between the point clouds. In their further work [94] the registration is performed by clustering geometric keypoint descriptor matches between the map segments. It is assumed that the IMU sensor is available and with it the z-direction is known.

Lu et al. [96] use a 2D prior map constructed by a high precision LIDAR mapping system to improve a low-quality monocular vision 3D point cloud map. The 2D prior map is used to correct the vertical planes constructed by the visual SLAM. 3D point cloud planes are periodically detected and matched with the prior, which is assumed to be more accurate. In the case of a match the mapping error is minimized according to the reference map.

A case of metric grid vs feature map fusion is presented in Husain et al. [79], where the merging of 2D occupancy grid maps and 3D point clouds to create a more complete environment representation is addressed. The robot relative positions are known, and the authors note the need for postprocessing due to limited bandwidth and processing capabilities of individual robots, and the robots only use coarse 2D maps during the mapping.

The current state of the art in heterogeneous metric grid map merging can be seen in Table 11. Abbreviations: NDT—normal distribution transform; PC—point cloud.

Table 11. Heterogeneous metric map merging approaches.

Map Type	Mapping Alg.	Relative Pos.	Global Map	Hardware	Methods	Comments
Grid (scale)	-	Unknown	Separate	-	[23,81,82,86]	All methods deal with different scale occupancy grid merging
Grid (scale, qual.)	-	Unknown	Separate	-	[87,88]	Methods can merge different scale and quality maps, if the environment has distinct regions. [87] finds nonlinear transformation.
NDT/prior	-	Unknown	Separate	-	[92]	Method fuses Normal Distribution Transform (NDT) map with prior emergency maps by using graph-based SLAM and corner feature matching. Only robot uses the resulting map.
Pose graph/CAD	Graph-based	Known	Separate	-	[89]	Method uses the CAD map prior to correct the pose graph map constructed by the robot.
Dense/sparse 3D PC	-	Known	-	LIDAR/Mono-cam.	[95]	The dense LIDAR 3D point cloud map is used for the localization with a monocular camera that produces sparse point set.
Dense/sparse 3D PC	-	Unknown	-	LIDAR/Mono-cam.	[93,94]	These methods improve the work by [95] by addressing the dense and sparse point cloud matching with unknown initial correspondences.
3D PC/prior	-	Known	Separate	LIDAR/Mono-cam.	[96]	The LIDAR constructed prior map is used as a reference map for low quality vision based mapping improvement.
3D PC/Grid	-	Known	Hybrid	Different LIDAR	[79]	The method merges 2D occupancy grid with 3D point cloud in postprocessing.

4.2. Metric Grid vs. Feature Map Merging

Map types used in robotics can be fundamentally different in their representations, but some methods exist to transform one map type into another. For example, it is possible to get feature maps from metric maps by applying feature (line, point, corner, etc.) detection algorithms [17,18]. Feature maps can be transformed into metric maps by creating an occupancy grid and marking cells containing features as occupied [97].

However, such transformation methods do not exist for all map types and in some cases, the reliable merging of two maps is fundamentally impossible. An example for such a case are two robots, which sense the environment with different sensors: one robot uses a camera to detect and map colored landmarks, but the other robot uses LIDAR, which cannot detect colors and therefore features detected by other robot are useless to it. It must also be noted that the merging of different type and format maps does not necessarily work in both directions. For example, it may be possible to incorporate information from the feature map in the occupancy grid, but not the other way around.

The most relevant research concerning metric grid vs feature map merging relates to the feature extraction from metric grids (some examples are [17,18,98]. These works do not directly deal with heterogeneous map merging, but feature extraction from metric maps can help find the correspondence between metric and feature maps if the features are similar. Although not exactly map merging, finding the correspondence (matching) of two maps is the map merging's first step, and the works that deal with feature extraction from metric maps can be considered relevant.

Li and Olson in [98] offer to use scale-independent corner feature detectors to find a transformation between two scans. Corner detection in occupancy grid maps can serve as a starting point in the corner feature map—occupancy grid merging. Corner extraction from grid maps is also used for map merging by Adluru et al. [17].

Feature extraction from occupancy grid maps idea is also used by Alnounou et al. in [18] with the difference that hybrid occupancy grid-feature maps are created during the mapping process. Instead of corner features, the line-segments, circular arcs, and curve features are extracted.

The state of the art in heterogeneous metric grid vs. feature map merging is shown in Table 12. The grid vs. feature map merging methods are currently limited with feature extraction from occupancy grid maps, but no research on incorporating these features in feature maps has been done.

Table 12. Heterogeneous metric map merging approaches.

Map Type	Mapping Alg.	Relative Pos.	Global Map	Hardware	Methods	Comments
Grid vs feature	-	Unknown	-	-	[17,18,98]	Methods deal with feature extraction from occupancy grids, but not feature and grid map merging. Cited methods are examples dealing with feature extraction; not an exhaustive list.

4.3. Metric Grid vs. Topological Map Merging

This section addresses the methods that establish the matches between the topological and metric maps. There are many methods that extract topological maps from occupancy grids (some examples are [99–102], but the outputs are very different.

Fabrizi and Saffiotti [99] extract the topology based on the shape of free space in grid maps focusing on large open spaces connected by narrow passages. Joo et al. [100] construct topological maps by detecting virtual doors with the help of corner features. These virtual doors then serve as edges between nodes that represent rooms. Schwertfeger and Birk [101] extract topology graphs derived from Voronoi diagrams that are themselves extracted from 2D grid maps. These graphs are used to assess the map quality by matching it and comparing it with the ground truth. Kakuma et al. [102] also extract topological graph for matching with the ground truth, but acquires it through region segmentation.

Table 13 shows the list of methods (not exhaustive) that deal with topological map extraction from metric grid maps. There are no known methods to the author of this review that would deal with generating grids from topological maps in robotic mapping or map merging context.

Table 13. Heterogeneous metric map merging approaches.

Map Type	Mapping Alg.	Relative Pos.	Global Map	Hardware	Methods	Comments
Grid vs topological	-	Unknown	-	-	[99–102]	Methods deal with extraction of topological structures from occupancy grids, but not topological and grid map merging. Cited methods are not an exhaustive list of topological structure extraction methods.

4.4. Other Maps

A prominent research area, related to heterogenous map merging, deals with cross-view image matching, where aerial view images are matched with data collected on the ground (some examples are [103–105]). Yamamoto and Nakagawa [103] consider a problem where 3D LIDAR data and satellite image data must be merged to improve building classification. Although the demonstrated results are satisfactory, this research has limited use in multi-robot map merging. The transformation is manually determined and satellite data is rarely used as a data source for robot mapping due to the lack of details necessary for robot's size. Work by [104] uses a deep convolutional neural network to detect buildings in aerial and ground images. They then retrieve the k nearest neighbors from the reference buildings using a Siamese network and use an efficient multiple nearest neighbor matching method based on dominant sets to find the nearest neighbors of buildings and match them. Work by Fanta-Jende et al. [105] addresses matching of mobile mapping data and aerial images by searching for mutual planes in both images and homogenizing images to achieve pixel-level accuracy. Some methods address the cross-view matching by using semantic information [106,107]. It must be noted that these methods do not comprise an extensive review of all cross-view image matching methods, but are examples of this promising research direction, which is related to heterogeneous map merging, but is currently not used in actual multi-robot systems.

Hofmeister et al. in [68] demonstrates the creation of heterogeneous maps with robot team, where a 'parent' robot is equipped with LIDAR and 'child' robots use cameras. When the occupancy grid map with 'parent' robot is acquired, the other robots create image maps suitable for independent localization under the guidance of parent robot. This research, however, only demonstrates cooperative creation of heterogeneous maps, but did not address the merging issue, if the maps are already created.

To the best of author's knowledge the article by Erinc et al. [83] presents the only research addressing both matching and fusion of fundamentally different map types. In the solution proposed by [83], additionally to the normal mapping, each robot is required to record the wireless signal strength of all access points. This wireless signal strength model is then later used to find overlap between different map types. The solution is tested by merging an occupancy grid and appearance-based map (undirected weighted graph, in which every vertex represents a camera image at a certain position), however, authors claim that any map types can be merged in this way [83].

The overview of methods merging various map types is depicted in Table 14.

Table 14. Heterogeneous metric map merging approaches.

Map Type	Mapping Alg.	Relative Pos.	Global Map	Hardware	Methods	Comments
Aerial/Ground	-	Various	Hybrid	Aerial/Ground	[103–107]	Currently limited use in multi-robot systems for real time mapping.
Grid/Image map	Cooperative	Known	LIDAR/Camera	Separate	[68]	Requires cooperative mapping, image maps can't be independently built.
Grid/Appearance	-	Unknown	-	Hybrid	[83]	Requires creation of wireless signal strength maps for all robots. Only considers finding correspondences, but not integration of map data.

5. Discussion and Challenges

The overview of the state of the art in heterogeneous map merging is summarized in Table 15. Various map matching and fusion algorithms have been created over the years for homogeneous maps, but the field of heterogeneous map merging is still full of challenges. Most of the progress has been made in the different scale occupancy grid merging. Another research direction in the heterogeneous map matching is the matching of the robot map with some kind of prior map (building plan, sketch, CAD plan) [89,90]. One more direction in the heterogeneous metric map merging is the matching of maps from different sensors for the localization [93–95] or map building purposes [96]. Some solutions also exist for fundamentally different map merging, but there are restrictions—either the relative transformation of maps must be known [79] or other information is required (WiFi signal strength map in [83]). Some solutions exist, that may support the metric grid vs feature map merging [17,18,98] and metric grid vs topological map merging [99–102], but those are developed for other purposes and their use for heterogeneous map merging purposes remains unknown.

Table 5 shows the overview of the heterogeneous map merging approaches. A map merging method must consider at least one heterogeneity factor (format, sensors or scale) to be considered heterogenous.

Based on the review and discovered factors and challenges, two main heterogeneous map merging steps can be defined:

1. Exchange of meta information. This step is necessary to determine whether the map merging is possible and which algorithm should be used. It must be noted that merging of different type and format maps may not necessarily work in both directions. For example, it can be possible to incorporate information from the feature map in the occupancy grid, but not the other way around. Meta information should include all important factors required to perform successful heterogeneous map merging or reject the merging attempt due to incompatibilities or non-existant algorithms.

    (a) Map type and any significant specific information that can be objectively described. Such information is, for example, the scale for occupancy grids, feature types for feature maps, stored topological information, etc. Additional information about extracted features, object data, semantic data should also be included.

    (b) Data that each robot is capable (and willing) to share and receive. The data can be just the map, or also include trajectory and raw sensor data. Hardware and communication channel limitations must be taken into account when determining shared information.

    (c) Relative position information. Map merging difficulty both for homogeneous and heterogeneous cases differs significantly based on the knowledge about the relative positioning of the robots.

2. Merging of maps. This is the main challenge, where specific heterogeneous map merging algorithms must be developed. Unfortunately, there is no universal solution as evidenced by the review of existing homogeneous and heterogeneous map merging methods, therefore case-specific algorithms must be developed. The merging is asymmetric, as each robot seeks to incorporate the data from another robot in its map.

Table 15. Heterogeneous metric map merging approaches.

Methods	Heterogeneity	Comments
[23,81,82,86,87]	Different scale occupancy grids	All methods focus on the map matching with unknown positions and are able to produce separate maps. [87] also considers different quality maps.
[87]	Different scale and quality occupancy grids	The approach focuses on the map matching with unknown positions and is able to produce separate maps.
[89,92,96]	Robot map versus a prior map.	The methods deal with matching the prior with a robot map. The result is intended to improve the robot map.
[93–96]	Different sensors	The methods address the maps constructed with different sensors. [95,96] assume known positions and perform localization [95] or map improvement [96]; [93,94] work on finding the transformation between the maps.
[79]	2D and 3D maps	The method fuses 2D and 3D map in one hybrid map, when the relative positions are known and all data is collected.
[17,18,98]	Feature and grid maps	Methods deal with the feature extraction from occupancy grids, but not feature and grid map merging.
[99–102]	Topological and grid maps	Methods deal with the extraction of topological structures from occupancy grids, but not topological and grid map merging.
[103–107]	Aerial and ground view matching	Methods address the cross-view localization problem for aerial and ground views.
[68]	Grid and image maps	The approach addresses cooperative heterogeneous map creation for robots with various capabilities.
[83]	Grid and appearance maps	The approach matches the maps that have little commonalities with the help of wireless signal strength maps. In the end both maps are fused in one hybrid map.

IEEE 1873–2015 [108] is a significant step towards a common standard regarding robot map data exchange and supports the representation and exchange of 2D grid, geometric (feature) and topological maps in XML format along with the corresponding metadata. It also allows us to represent the environment with all supported maps in combination in a coherent way. Notably, map positions can be specified relative to the coordinate system of another local map, which is a very useful feature when merging maps. Although several options are still missing for this standard to cover all metadata requirements listed above, in a few years this representation could cover most of the needs in heterogeneous map merging data exchange:

- Representation of all map types, most notably any type of 3D maps (noted in [108] as possible future development).
- Semantic information (noted in [108] as possible future development).

- Data other than map (trajectory, raw sensor data). This is a minor issue, as based on the most recent methods, very few map merging methods require this type of data.

6. Conclusions

This article provides a state of the art of the homogeneous and heterogeneous map merging field and the main challenges that must be overcome to successfully implement the capability to merge different maps in a multi-robot team. Six factors are identified that influence the outcome of map merging: (1) robotic platform hardware configurations, (2) map representation types, (3) mapping algorithms, (4) shared information between robots, (5) relative positioning information, (6) sharing level of resulting global maps.

The influence of most important factors is analyzed in the context of heterogeneous map merging and two main steps for problem-solving are defined for the general merging case—(1) meta-information exchange and (2) asymmetric merging of maps based on algorithm tailored for the specific case.

It is concluded that heterogeneous map merging still has numerous challenges that must be addressed:

1. Heterogeneous map merging algorithms for specific cases. So far only some algorithms to merge different scale occupancy grid maps or heterogenous maps with specific restrictions exist, and even these solutions do not address asymmetric merging, where a separate global map is produced for each involved robotic platform.
2. The incorrect merging chances are higher with heterogeneous maps, therefore it is necessary to research mechanisms to reduce the risk of map corruptions due to mistakes. Some possible solutions include multi-level map storage solutions or meeting strategies to confirm merging decisions.
3. The merging of different quality maps is currently severely lacking even for same-type maps. To facilitate the propagation of the higher quality maps, map quality assessment algorithms are necessary.

Funding: This work has been supported by the European Regional Development Fund within the Activity 1.1.1.2 "Post-doctoral Research Aid" of the Specific Aid Objective 1.1.1 "To increase the research and innovative capacity of scientific institutions of Latvia and the ability to attract external financing, investing in human resources and infrastructure" of the Operational Programme "Growth and Employment" (No. 1.1.1.2/VIAA/1/16/030).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saeedi, S.; Trentini, M.; Seto, M.; Li, H. Multiple-Robot Simultaneous Localization and Mapping: A Review. *J. Field Robot.* **2016**, *33*, 3–46. [CrossRef]
2. Thrun, S. A probabilistic on-line mapping algorithm for teams of mobile robots. *Int. J. Robot. Res.* **2001**, *20*, 335–363. [CrossRef]
3. Howard, A. Multi-robot simultaneous localization and mapping using particle filters. *Int. J. Robot. Res.* **2006**, *25*, 1243–1256. [CrossRef]
4. Fenwick, J.W.; Newman, P.M.; Leonard, J.J. Cooperative concurrent mapping and localization. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Washington, DC, USA, 11–15 May 2002; pp. 1810–1817.
5. Dudek, G.; Jenkin, M.; Milios, E.; Wilkes, D. Topological exploration with multiple robots. In Proceedings of the 7th International Symposium on Robotics with Application (ISORA), Anchorage, AK, USA, 10–14 May 1998; Volume 3.
6. Bonanni, T.M.; Grisetti, G.; Iocchi, L. Merging partially consistent maps. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*; Springer: Cham, Switzerland, 2014; pp. 352–363.
7. Rodriguez-Losada, D.; Matia, F.; Jimenez, A. Local maps fusion for real time multirobot indoor simultaneous localization and mapping. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; Volume 2, pp. 1308–1313.

8. Zhou, X.S.; Roumeliotis, S.I. Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 1785–1792.
9. Ozkucur, N.E.; Akın, H.L. Cooperative multi-robot map merging using fast-SLAM. In *Robot Soccer World Cup*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 449–460.
10. Birk, A.; Carpin, S. Merging occupancy grid maps from multiple robots. *Proc. IEEE* **2006**, *94*, 1384–1397. [CrossRef]
11. Dedeoglu, G.; Sukhatme, G.S. Landmark-based matching algorithm for cooperative mapping by autonomous robots. *Distrib. Auton. Robot. Syst.* **2000**, *4*, 251–260.
12. Huang, W.H.; Beevers, K.R. Topological map merging. *Int. J. Robot. Res.* **2005**, *24*, 601–613. [CrossRef]
13. Konolige, K.; Fox, D.; Limketkai, B.; Ko, J.; Stewart, B. Map merging for distributed robot navigation. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 1, pp. 212–217.
14. Ko, J.; Stewart, B.; Fox, D.; Konolige, K.; Limketkai, B. A practical, decision-theoretic approach to multi-robot mapping and exploration. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 4, pp. 3232–3238.
15. Liu, Y.; Fan, X.; Zhang, H. A fast map merging algorithm in the field of multirobot SLAM. *Sci. World J.* **2013**, *2013*, 169635. [CrossRef] [PubMed]
16. Ho, K.; Newman, P. Multiple map intersection detection using visual appearance. In Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore, 14–16 December 2005.
17. Adluru, N.; Latecki, L.J.; Sobel, M.; Lakaemper, R. Merging maps of multiple robots. In Proceedings of the 2008 19th International Conference on Pattern Recognition, Tampa, FL, USA, 8–11 December 2008; pp. 1–4.
18. Alnounou, Y.; Paulik, M.J.; Krishnan, M.; Hudas, G.; Overholt, J. Occupancy Grid Map Merging Using Feature Maps. In Proceedings of IASTED Technology Conferences, Cambridge, MA, USA, 1–3 November 2010.
19. Blanco, J.L.; González-Jiménez, J.; Fernández-Madrigal, J.A. A robust, multi-hypothesis approach to matching occupancy grid maps. *Robotica* **2013**, *31*, 687–701. [CrossRef]
20. Carpin, S. Fast and accurate map merging for multi-robot systems. *Auton. Robots* **2008**, *25*, 305–316. [CrossRef]
21. Saeedi, S.; Paull, L.; Trentini, M.; Seto, M.; Li, H. Map merging using Hough peak matching. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugalc, 7–12 October 2012; pp. 4683–4688.
22. Lee, H.C.; Roh, B.S.; Lee, B.H. Multi-hypothesis map merging with sinogram-based PSO for multi-robot systems. *Electron. Lett.* **2016**, *52*, 1213–1214. [CrossRef]
23. Topal, S.; Erkmen, D.; Erkmen, A.M. A novel map merging methodology for multi-robot systems. In Proceedings of the World Congress on Engineering and Computer Science 2010, San Francisco, CA, USA, 20–22 October 2010; Volume I, pp. 383–387.
24. Hermann, M.; Pentek, T.; Otto, B. Design principles for industrie 4.0 scenarios. In Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, USA, 5–8 January 2016; pp. 3928–3937.
25. Carlone, L.; Ng, M.K.; Du, J.; Bona, B.; Indri, M. Rao-Blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 243–249.
26. Burgard, W.; Moors, M.; Fox, D.; Simmons, R.; Thrun, S. Collaborative multi-robot exploration. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24–28 April 2000; pp. 476–481.
27. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
28. Thrun, S.; Liu, Y. Multi-robot SLAM with sparse extended information filers. In *Robotics Research. The Eleventh International Symposium*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 254–266.
29. Dayoub, F.; Cielniak, G.; Duckett, T. Long-term experiments with an adaptive spherical view representation for navigation in changing environments. *Robot. Auton. Syst.* **2011**, *59*, 285–295. [CrossRef]
30. Konolige, K.; Bowman, J. Towards lifelong visual maps. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 1156–1163.

31. Mühlfellner, P.; Bürki, M.; Bosse, M.; Derendarz, W.; Philippsen, R.; Furgale, P. Summary maps for lifelong visual localization. *J. Field Robot.* **2016**, *33*, 561–590. [CrossRef]
32. Chang, H.J.; Lee, C.G.; Hu, Y.C.; Lu, Y.H. Multi-robot SLAM with topological/metric maps. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 1467–1472.
33. Cunningham, A.; Paluri, M.; Dellaert, F. DDF-SAM: Fully distributed SLAM using constrained factor graphs. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3025–3030.
34. Cunningham, A.; Indelman, V.; Dellaert, F. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 5220–5227.
35. Thrun, S.; Montemerlo, M. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *Int. J. Robot. Res.* **2014**, *25*, 403–429. [CrossRef]
36. Mohanarajah, G.; Usenko, V.; Singh, M.; D'Andrea, R.; Waibel, M. Cloud-based collaborative 3D mapping in real-time with low-cost robots. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 239–244. [CrossRef]
37. Michael, N.; Shen, S.; Mohta, K.; Kumar, V.; Nagatani, K.; Okada, Y.; Kiribayashi, S.; Otake, K.; Yoshida, K.; Ohno, K.; et al. Collaborative mapping of an earthquake damaged building via ground and aerial robots. *J. Field Robot.* **2012**, *29*, 832–841. [CrossRef]
38. Jessup, J.; Givigi, S.N.; Beaulieu, A. Merging of octree based 3d occupancy grid maps. In Proceedings of the 2014 IEEE International Systems Conference Proceedings, Ottawa, ON, Canada, 31 March–3 April 2014; pp. 371–377.
39. Jessup, J.; Givigi, S.N.; Beaulieu, A. Robust and efficient multirobot 3-d mapping merging with octree-based occupancy grids. *IEEE Syst. J.* **2017**, *11*, 1723–1732. [CrossRef]
40. Yue, Y.; Senarathne, P.N.; Yang, C.; Zhang, J.; Wen, M.; Wang, D. Hierarchical Probabilistic Fusion Framework for Matching and Merging of 3-D Occupancy Maps. *IEEE Sens. J.* **2018**, *18*, 8933–8949. [CrossRef]
41. Schuster, M.J.; Br, C.; Hirschmüller, H.; Suppa, M.; Beetz, M. Multi-robot 6D graph SLAM connecting decoupled local reference filters. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5093–5100.
42. Bonanni, T.M.; Della Corte, B.; Grisetti, G. 3-D map merging on pose graphs. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1031–1038. [CrossRef]
43. He, S.; Qin, X.; Zhang, Z.; Jagers, M. Incremental 3D Line Segment Extraction from Semi-dense SLAM. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1658–1663.
44. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. *Sens. Fusion IV Control Paradig. Data Struct.* **1992**, *1611*, 586–607. [CrossRef]
45. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robots* **2013**, *34*, 189–206. [CrossRef]
46. Ohno, K.; Tadokoro, S.; Nagatani, K.; Koyanagi, E.; Yoshida, T. Trials of 3-D map construction using the tele-operated tracked vehicle Kenaf at Disaster City. In Proceedings of the 2010 IEEE International Conference on robotics and automation, Anchorage, AK, USA, 3–7 May 2010; pp. 2864–2870.
47. Dryanovski, I.; Morris, W.; Xiao, J. Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 1553–1559.
48. Luo, W.; Zhang, H. Visual analysis of large-scale lidar point clouds. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2487–2492.
49. Wolff, K.; Kim, C.; Zimmer, H.; Schroers, C.; Botsch, M.; Sorkine-Hornung, O.; Sorkine-Hornung, A. Point cloud noise and outlier removal for image-based 3D reconstruction. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 118–127.
50. Chen, S.; Tian, D.; Feng, C.; Vetro, A.; Kovačević, J. Fast resampling of three-dimensional point clouds via graphs. *IEEE Trans. Signal Process.* **2017**, *66*, 666–681. [CrossRef]
51. Tombari, F.; Salti, S.; Di Stefano, L. A combined texture-shape descriptor for enhanced 3D feature matching. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 809–812.

52. Ma, X.; Guo, R.; Li, Y.; Chen, W. Adaptive genetic algorithm for occupancy grid maps merging. In Proceedings of the 2008 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008; pp. 5716–5720.
53. Li, H.; Nashashibi, F. A new method for occupancy grid maps merging: Application to multi-vehicle cooperative local mapping and moving object detection in outdoor environment. In Proceedings of the 2012 12th International Conference on Control Automation Robotics and Vision, Guangzhou, China, 5–7 December 2012; pp. 632–637.
54. Reid, R.; Bräunl, T. Large-scale Multi-robot Mapping in MAGIC 2010. In Proceedings of the 2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM), Qingdao, China, 17–19 September 2011; pp. 239–244.
55. Ballesta, M.; Gil, A.; Reinoso, O.; Juliá, M.; Jiménez, L.M. Alignment of visual maps in multirobot fastslam. In Proceedings of the 8th WSEAS International Conference on Computational Intelligence, Man-machine Systems and Cybernetics (CIMMACS), Puerto De La Cruz, Spain, 14–16 December 2009; pp. 92–97.
56. Lakaemper, R.; Latecki, L.J.; Wolter, D. Incremental multi-robot mapping. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 3846–3851.
57. Dinnissen, P.; Givigi, S.N.; Schwartz, H.M. Map merging of multi-robot slam using reinforcement learning. In Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, Korea, 14–17 October 2012; pp. 53–60.
58. Serafin, J.; Grisetti, G. NICP: Dense normal based point cloud registration. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 742–749.
59. Dubé, R.; Gawel, A.; Sommer, H.; Nieto, J.; Siegwart, R.; Cadena, C. An online multi-robot slam system for 3d lidars. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1004–1011.
60. Dubé, R.; Dugas, D.; Stumm, E.; Nieto, J.; Siegwart, R.; Cadena, C. Segmatch: Segment based place recognition in 3d point clouds. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5266–5272.
61. Pathak, K.; Birk, A.; Vaskevicius, N.; Pfingsthorn, M.; Schwertfeger, S.; Poppinga, J. Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation. *J. Field Robot.* **2010**, *27*, 52–84. [CrossRef]
62. Hahnel, D.; Burgard, W.; Thrun, S. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robot. Auton. Syst.* **2003**, *44*, 15–27. [CrossRef]
63. Elfes, A. Occupancy grids: A stochastic spatial representation for active robot perception. *Proc. Sixth Conf. Uncertain. AI* **1990**, *2929*, 6.
64. Pfingstorn, M.; Slamet, B.; Visser, A. A scalable hybrid multi-robot SLAM method for highly detailed maps. In *Robot Soccer World Cup*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 457–464.
65. Triebel, R.; Pfaff, P.; Burgard, W. Multi-level surface maps for outdoor terrain mapping and loop closing. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 2276–2282.
66. Pfaff, P.; Triebel, R.; Burgard, W. An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *Int. J. Robot. Res.* **2007**, *26*, 217–230. [CrossRef]
67. Wurm, K.M.; Hennes, D.; Holz, D.; Rusu, R.B.; Stachniss, C.; Konolige, K.; Burgard, W. Hierarchies of octrees for efficient 3d mapping. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4249–4255.
68. Hofmeister, M.; Kronfeld, M.; Zell, A. Cooperative visual mapping in a heterogeneous team of mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 1491–1496.
69. Erinc, G.; Carpin, S. Anytime merging of appearance-based maps. *Auton. Robots* **2014**, *36*, 241–256. [CrossRef]
70. Elibol, A.; Kim, J.; Gracias, N.; Garcia, R. Efficient image mosaicing for multi-robot visual underwater mapping. *Pattern Recognit. Lett.* **2014**, *46*, 20–26. [CrossRef]

71. Howard, A. Multi-robot mapping using manifold representations. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; Volume 4, pp. 4198–4203.
72. Nieto, J.I.; Guivant, J.E.; Nebot, E.M. The hybrid metric maps (HYMMs): A novel map representation for DenseSLAM. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; Volume 1, pp. 391–396.
73. Wurm, K.M.; Stachniss, C.; Grisetti, G. Bridging the gap between feature-and grid-based SLAM. *Robot. Auton. Syst.* **2010**, *58*, 140–148. [CrossRef]
74. Schmuck, P.; Scherer, S.A.; Zell, A. Hybrid metric-topological 3D occupancy grid maps for large-scale mapping. *IFAC-PapersOnLine* **2016**, *49*, 230–235. [CrossRef]
75. Ko, B.Y.; Song, J.B.; Lee, S. Real-time building of a thinning-based topological map with metric features. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; Volume 2, pp. 1524–1529.
76. Kostavelis, I.; Gasteratos, A. Semantic mapping for mobile robotics tasks: A survey. *Robot. Auton. Syst.* **2015**, *66*, 86–103. [CrossRef]
77. Salas-Moreno, R.F.; Newcombe, R.A.; Strasdat, H.; Kelly, P.H.; Davison, A.J. Slam++: Simultaneous localisation and mapping at the level of objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1352–1359.
78. Marjovi, A.; Choobdar, S.; Marques, L. Robotic clusters: Multi-robot systems as computer clusters: A topological map merging demonstration. *Robot. Auton. Syst.* **2012**, *60*, 1191–1204. [CrossRef]
79. Husain, A.; Jones, H.; Kannan, B.; Wong, U.; Pimentel, T.; Tang, S.; Daftry, S.; Huber, S.; Whittaker, W.L. Mapping planetary caves with an autonomous, heterogeneous robot team. In Proceedings of the 2013 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2013; pp. 1–13.
80. Saeedi, S.; Paull, L.; Trentini, M.; Seto, M.; Li, H. Group mapping: A topological approach to map merging for multiple robots. *IEEE Robot. Autom. Mag.* **2014**, *21*, 60–72. [CrossRef]
81. Park, J.; Sinclair, A.J.; Sherrill, R.E.; Doucette, E.A.; Curtis, J.W. Map merging of rotated, corrupted, and different scale maps using rectangular features. In Proceedings of the 2016 IEEE/ION Position, Location and Navigation Symposium (PLANS), Savannah, GA, USA, 11–14 April 2016; pp. 535–543.
82. Ferrao, V.T.; Vinhal, C.D.N.; da Cruz, G. An Occupancy Grid Map Merging Algorithm Invariant to Scale, Rotation and Translation. In Proceedings of the 2017 Brazilian Conference on Intelligent Systems (BRACIS), Uberlandia, Brazil, 2–5 October 2017; pp. 246–251.
83. Erinc, G.; Balaguer, B.; Carpin, S. Heterogeneous map merging using WiFi signals. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 5258–5264.
84. Ch, P.; Carnegie, D.A. Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots. *Robot. Auton. Syst.* **2013**, *61*, 565–579.
85. Fankhauser, P.; Bloesch, M.; Krusi, P.; Diethelm, R.; Wermelinger, M.; Schneider, T.; Dymczyk, M.; Hutter, M.; Siegwart, R. Collaborative navigation for flying and walking robots. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 2859–2866.
86. Ma, L.; Zhu, J.; Zhu, L.; Du, S.; Cui, J. Merging grid maps of different resolutions by scaling registration. *Robotica* **2016**, *34* , 2516–2531. [CrossRef]
87. Shahb, S.G.; Magnusson, M. 2D map alignment with region decomposition. *Auton. Robots* **2019**, *43*, 1117–1136.
88. Shahb, S.G.; Magnusson, M.; Iagnemma, K. Nonlinear Optimization of Multimodal Two-Dimensional Map Alignment With Application to Prior Knowledge Transfer. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2040–2047.
89. Boniardi, F.; Caselitz, T.; Kümmerle, R.; Burgard, W. A pose graph-based localization system for long-term navigation in CAD floor plans. *Robot. Auton. Syst.* **2019**, *112*, 84–97. [CrossRef]
90. Mielle, M.; Magnusson, M.; Lilienthal, A.J. Using sketch-maps for robot navigation: Interpretation and matching. In Proceedings of the 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Lausanne, Switzerland, 23–27 October 2016; pp. 252–257.

91. Neuhaus, M.; Bunke, H. An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification. In Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Lisbon, Portugal, 18–20 August 2004; pp. 180–189.
92. Mielle, M.; Magnusson, M.; Andreasson, H.; Lilienthal, A.J. SLAM auto-complete: Completing a robot map using an emergency map. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 35–40.
93. Gawel, A.; Cieslewski, T.; Dubé, R.; Bosse, M.; Siegwart, R.; Nieto, J. Structure-based vision-laser matching. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 182–188.
94. Gawel, A.; Dubé, R.; Surmann, H.; Nieto, J.; Siegwart, R.; Cadena, C. 3d registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 27–34.
95. Caselitz, T.; Steder, B.; Ruhnke, M.; Burgard, W. Matching geometry for long-term monocular camera localization. In Proceedings of the ICRA Workshop: AI for Long-Term Autonomy, Stockholm, Sweden, 16–21 May 2016.
96. Lu, Y.; Lee, J.; Yeh, S.H.; Cheng, H.M.; Chen, B.; Song, D. Sharing heterogeneous spatial knowledge: Map fusion between asynchronous monocular vision and lidar or other prior inputs. In Proceedings of the 18th International Symposium on Robotics Research (ISRR), Puerto Varas, Chile, 11–14 December 2017; Volume 158.
97. Andersone, I. Probabilistic Mapping with Ultrasonic Distance Sensors. *Procedia Comput. Sci.* **2017**, *104*, 362–368.
98. Li, Y.; Olson, E.B. A general purpose feature extractor for light detection and ranging data. *Sensors* **2010**, *10*, 10356–10375. [CrossRef]
99. Fabrizi, E.; Saffiotti, A. Extracting topology-based maps from gridmaps. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 3, pp. 2972–2978.
100. Joo, K.; Lee, T.K.; Baek, S.; Oh, S.Y. Generating topological map from occupancy grid-map using virtual door detection. In Proceedings of the IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010; pp. 1–6.
101. Schwertfeger, S.; Birk, A. Evaluation of map quality by matching and scoring high-level, topological map structures. In Proceedings of the 2013 IEEE international conference on robotics and automation, Karlsruhe, Germany, 6–10 May 2013; pp. 2221–2226.
102. Kakuma, D.; Tsuichihara, S.; Ricardez, G.A.G.; Takamatsu, J.; Ogasawara, T. Alignment of occupancy grid and floor maps using graph matching. In Proceedings of the 2017 IEEE 11th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 30 January–1 February 2017; pp. 57–60.
103. Yamamoto, T.; Nakagawa, M. Merging Airborne LIDAR Data and Satellite SAR Data for Building Classification. *Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci.* **2015**, *40*, 227. [CrossRef]
104. Tian, Y.; Chen, C.; Shah, M. Cross-view image matching for geo-localization in urban environments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3608–3616.
105. Fanta-Jende, P.; Nex, F.; Vosselman, G.; Gerke, M. Co-registration of panoramic mobile mapping images and oblique aerial images. *Photogram. Record* **2019**, *34*, 148–173. [CrossRef]
106. Gawel, A.; Del Don, C.; Siegwart, R.; Nieto, J.; Cadena, C. X-view: Graph-based semantic multi-view localization. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1687–1694. [CrossRef]
107. Castaldo, F.; Zamir, A.; Angst, R.; Palmieri, F.; Savarese, S. Semantic cross-view matching. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 7–13 December 2015; pp. 9–17.
108. Amigoni, F.; Yu, W.; Andre, T.; Holz, D.; Magnusson, M.; Matteucci, M.; Moon, H.; Yokotsuka, M.; Biggs, G.; Madhavan, R. A Standard for Map Data Representation: IEEE 1873-2015 Facilitates Interoperability between Robots. *IEEE Robot. Autom. Mag.* **2018**, *25*, 65–76. [CrossRef]