sustainability

MDPI

Article

SLAM for Humanoid Multi-Robot Active Cooperation Based on Relative Observation

Zhaoyi Pei [1], Songhao Piao [1,*], Mohammed El Habib Souidi [2], Muhammad Zuhair Qadir [1] and Guo Li [1]

[1] Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China; peizhaoyi@stu.hit.edu.cn (Z.P.); mzuhairqadir@gmail.com (M.Z.Q.); 14B903020@hit.edu.cn (G.L.)
[2] Department of Computer Science, University of Khenchela, Khenchela 40000, Algeria; mohamed.souidi@hit.edu.cn
* Correspondence:piaosh@hit.edu.cn

Received: 26 June 2018; Accepted: 10 August 2018; Published: 20 August 2018

Abstract: The simultaneous localization and mapping (SLAM) of robot in the complex environment is a fundamental research topic for service robots. This paper presents a new humanoid multi-robot SLAM mechanism that allows robots to collaborate and localize each other in their own SLAM process. Each robot has two switchable modes: independent mode and collaborative mode. Each robot can respond to the requests of other robots and participate in chained localization of the target robot under the leadership of the organiser. We aslo discuss how to find the solution of optimal strategy for chained localization. This mechanism can improve the performance of bundle adjustment at the global level, especially when the image features are few or the results of closed loop are not ideal. The simulation results show that this method has a great effect on improving the accuracy of multi-robot localization and the efficiency of 3D mapping.

Keywords: SLAM; multi-robot system (MRS); humanoid robot; cooperative localization

1. Introduction

Artificial intelligence has been widely applied to environmental science including environmental management [1] and sustainability [2]. As the main platform of artificial intelligence applications, intelligent robots have always been research emphasis of sustainability. Robots can save a lot of manpower costs, and robots can also adapt to complex and dangerous environment such as alien planet [3] and disaster or hazardous area [4,5]. Humanoid robots are referred as the mechanical device having structure and motion ability similar to the human beings. Relative to the tracked and wheeled, multi-legged mobile robots, the humanoid robots has the advantages of strong adaptability, doing large scope of work, rich action form and high efficiency of energy optimization. It has been an active area of research in the field of robotics. Now humanoids have become the preferred platform for artificial intelligence applications and human services. In addition, humanoid robots have a similar appearance and behavior patterns as compared with the humans as matter of fact they are considered as the first choice for service robots which are more compatible with humans than other types of robots.

Simultaneous localization and mapping (SLAM) can be described as: a robot in an unknown environment from an unknown location start to move and localize itself in the process of moving according to the pose estimation and map that has been generated, the incremental map is constructed at the same time. The autonomous localization and navigation of the robot can be realized based on built map and sensor data. A hard nut to crack is the error of SLAM will be accumulated with the movement of the robot. One of the main methods to eliminate the error is closed-loop detection. But the robot can detect the closed loop only when it has reached the postion it had been before

which will make efficiency of exploration suddenly drop in the face of larger and more complex environments. Therefore, SLAM for multi-robot system has attracted more attention from the research community. Multi-robot system (MRS) is a set of multiple agents, members in MRS coordinate and serve each other to complete a task together which can not be accomplished without cooperation. Multi-robot cooperative SLAM has always been the focus of robotics research. Detailed maps and precise localization are the basis for MRS to accomplish other tasks. Cooperation among robots is the core of MRS for SLAM expecially for humanoid robots whose localization ability is not strong enough due to lack of accurate and sufficient motion sensor information (For example, a wheeled robot has a odometer but a humanoid robot doesn't), but the emphsis in most of the studies about multi-robot cooperation for humanoid SLAM, is to construct the independent robot map fushion results. Closed-loop detection of robot is relatively independent and can only eliminate the position error of itself, if the robot can locate each other according to their own situation, efficiency of multi humanoid robot SLAM will be greatly improved.

In this paper, the localization self-confidence and localization tying for a humenoid MRS are proposed. Each robot has two kinds of switching mode, independent mode and cooperative mode, the robot can proposed the request of localization assistance according to the its localization self-confidence, other robots response to the request according to their localization assistant degree, besides, we propose a method of multi-robot localization called chained localization. Finally, a method to obtain the optimal strategy of chained localization is aslo discussed. The configuration and additional properties of a biped robot in order to achieve this collaborate localization mechanism is introduced in Section 4. A vision based localization function between two robots is informed in Section 5. We discuss the algorithm of how to find optimal solution of cooperation in Section 6. The total sequences of our mechanism and the simulation experiments are shown in Sections 7 and 8.

2. Related Work

Relative observation between robots to solve cooperative localization has always been a hot topic in MRS research. Some geometric based algorithms can be used to calculate the relative positions between two robot satisfying certain constraints [6,7]. AT Rashid, M Frasca proposed a new algorithm called cluster matching where clusters of nodes scanned by distance IR sensor are deployed to estimate location and orientation of the robots which can't be tracked [8]. Nikolay Atanasov, Jerome Le Ny focused on getting optimal control strategy by minimizing the entropy of the estimation task [9]. MW Mehrez and Gki Mann designed a framework in which robots are located and commanded to follow time varying trajectories with respect to another robot reference frame to get optimal relative localization [10]. D Fox, W Burgard employed probabilistic methods to measure each robot's belief whenever one robot detects another so that high-cost sensors among groups can be shared across multiple robots [11]. Od Silva and Gki Mann proposed a target tracking inspired design which can accommodate communication at a low predefined [12]. Some filtering algorithms such as evolutionary assisted particle filter (EAPF) [13] and information filter (IF) [14] were also used to deal with the problem of multi-sensor information fusion in cooperative localization of multiple robots. Some work also addressed wireless communication [15] and salient landmarks [16].

Some research focus on the integration and representation of maps such as pose graph [17] and 6D graph [18]. Mw Kai, C Stachniss presented an MRS for SLAM with multiple representations of the maps which can be switched according to the current observation [19]. The critical innovation point of reference [20] is the design of mechanism which allows multiple threads to concurrently read and modify the same map for multiple micro aerial vehicles. Lecture [21] describes strategy using data of 2D LIDAR sensors to build a joint map instead of occasional merging of smaller maps.

Hybrid multi robot systems can combine different abilities and different structures of robots in the same type of tasks. Michael Milford, Gordon Wyeth present a hybrid reactive control system to perform SLAM [22]. Lecture [23] present a two-level mapping method for MRS which allows robots set up the constraints of local and global maps when they enter a new camera visual field.

3. Discussion

The works referred to in the previous section demonstrate state of the art of SLAM for MRS and can be summarized as the solution of the following three open research questions for Multi-robots SLAM.

Task allocation: The task allocation of a multi-robot SLAM is to assign the robots to the suitable environment for their exploration, that is, to reorganize the team of robots under the condition that certain environment is known. For instance, assigning unmanned aerial vehicle (UAV) to places which are difficult to navigate for unmanned vehicles to replace unmanned vehicles can be seen as an optimization of task allocation. This kind of optimization can be regarded as the optimization of local mapping and localization (single robot SLAM) at the higher level.

Relative localization: Relative localization means that the robots determines the relative poses through mutual observation. Observation can be active or casual. This relative localization is an important issue for combination two maps built with data from two robots.

Map representation and Consistency of map: The ultimate goal of SLAM for MRS is to generate a globally consistent map, which consists of local maps generated by single robots. Research on map representation is to use new forms of maps to better represent local and global maps built by MRS. One of the main considerations in this study is to ensure the consistency of the map, which is to guarantee that the local maps generated by single robot are accurate when they are uploaded to the global map. If local maps generate by robots overlaps, the robots are able to correct the map and their trajectories by combining their observations to decrease measurement errors.

The above three issue are also the problem that need to overcome for applying single robot SLAM algorithm to multi-robot SLAM. There are strong correlation between these three points, that is to say, optimizing one aspect will inevitably affect the overall effect of multi-robot SLAM and the operation of the other two aspects. For example, if the local maps uploaded by two robots are overlaped, the overlap part can be used to correct the estimation of relative localization, otherwise the result relative localization can be used to generate a global map containing the local maps of the robots.

When the robots in multi-agent system localize themself using some of existing algorithms, they are more dependent on static landmarks and some auxiliary localization such as GPS and IR. When it comes to relative observation algorithms, robots will observe other robots on condition that their localization accuracy is unknown, which will lead to further accumulation of errors (If the robot A gets the pose under the relative observation of B, and the error of B's self localization is very large, then the error of A's self localization will be great too in a large probability). This paper mainly adresses the active relative observation and tries to solve and propose a mechanism that can cooperative localize based on the accuracy of the self localization of each robot without the assistance of a global localization sensor and landmarks.

4. Configuration and Properties of Humanoid Robots

4.1. Localization Self-Confidence

Localization self-confidence (LSC) can be used to estimate the accuracy of a robot's localization to its own pose. LSC will continue to decay as the robot itself moves because the cumulative error of the localization will increase continuously if no closed looping is detected. The LSC will be reset if the closed loop is detected. LSC is updated such as Algorithm 1 without no assistance of other robots. while η is decay factor. We assume that every step of the robot takes the same localization error. So η is a constant. This is the main attribute of a robot according to which the robot decides whether it needs assistance or not and whether it can assistant others or not. Information provided by robots with a higher LSC is believed to have higher credibility.

Algorithm 1 Update of localization self-confidence

```
robot.lsc ← 1
while robot works do
    if robot moves then
        robot.lsc = η ∗ robot.lsc
    end if
    if robot detects closed looping then
        robot.lsc ← 1
    end if
end while
```

4.2. Chained Localization

Localization chain $\{A_1, A_2....A_n\}$ means A_i (i from 1 to $n-1$) calculates the pose of A_{i+1} according to its pose and the observation of A_{i+1}. A_{i+1} will update its pose as the calculated value of $A_i's$, and continue to calculate the pose of A_{i+2} with its new pose. This way of localization is called chained localization which is the mechanism of collaborative SLAM among robots proposed in this article. The running process is shown as follows:

$$P_i = localization(...localization(localization(P_1, V_{1,2}), V_{2,3}), V_{i-1,i}) \tag{1}$$

where localization is the algorithm to get pose of robot i (P_i) according to pose of robot i (P_{i-1}) and observation of robot j ($V_{i,i-1}$). Figure 1 is diagrammatic of chained localization.

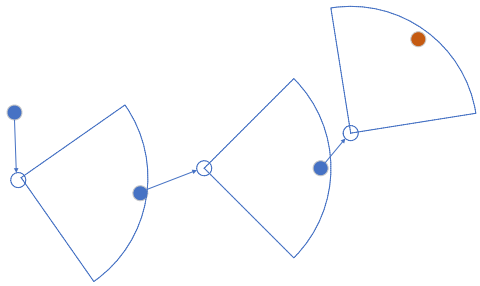

Figure 1. Chained localization.

4.3. Localization Self-Confidence Updating Method for Chained Localization

After the successful implementation of the chained localization, the LSC of the robot i (from 2 to n) participating in the chained localization is updated as follows:

$$LSC_i = (\gamma_{i-1,i}.....\gamma_{2,3}((\gamma_{1,2}(lsc_1 + k * dist_{1,2})) + k * dist_{2,3}) + k * dist_{i-1,i}) \tag{2}$$

γ_{ij} is observation error coefficient from i to j of which the range is 0 to 1. The observation error of robot i to robot j obeys normal distribution. So when calculating localization self-confidence, the robot should take into account the effects of these errors and let LSC be multiplied by this discount factor. γ_{ij} can be seen as an inherent attribute codetemined by robot i and robot j. $k * dist_{ij}$ indicates the loss of LSC due to robot i covers $dist_{ij}$ in order to observe robot j. In this paper, k and γ are set as constant.

4.4. Localization Tying

Localization tying shows the fitness of one robot to assistant another robot for localization. $tying_{ij}$ means localization tying of robot i to robot j. $tying_{ij}$ is calculated as follows:

$$tying_{ij} = \frac{lsc_i - lsc_j}{dist_{ij}} \tag{3}$$

while $dist_{ij}$ represents the distance that robot i should cover if it wants to assistant robot j. If this path is not found, i will be deleted from the list of assistant robots.

4.5. Path-Finding Algorithm

D* algorithm [24] is applied for 3D path-finding which is mainly used for robot exploration. It is aslo the path finding algorithm used by the Mars probe. D* algorithm is a heuristic shortest path algorithm which can be applied to the dynamic environment. It can reduce the range and cost of the search path by defining the estimation function as the estimation of the length of the path. First, it starts to search the nearby reachable points from the starting point and saves their estimation functions and search outwards based on searched points. In the process of search, reachable points with minimum estimated value is selected greedily out until the end point is reached.

4.6. Organiser

Organiser is the mainstay of this MAS as it will handle the assistant requests of all robots and giving the optimal cooperation strategy.

5. Assistant Localization between Two Red Robots

In this section, we specifically discuss how a robot can localize another robot in chained localization. The robot that is localized is called the target robot and the robot which localizes it is called assistant robot. Each robot uses RGB-D camera to collect visual information. First, A part of the 3D feature points of the target robot can be searched in the RGB-D images collected by the robot. Next, these feature points are matched with the feature points on the target robot's 3D model like Figure 2.

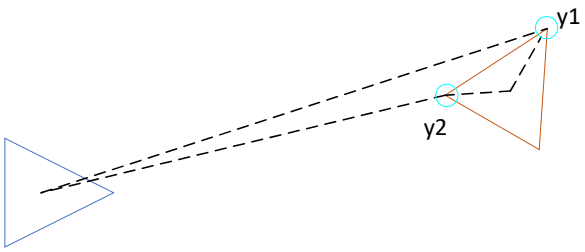

Figure 2. Assistant localization between two robots.

5.1. Obtaining the World Coordinates of Features Point on Target Robots

Set the coordinate of features point y under assistant robot's coordinates frame as $P = [X, Y, Z]^T$ and coordinates on the physical imaging plane as $P' = [X', Y', Z']^T$. The relationship between P and P' is shown in Equation (4).

$$\begin{cases} X' = f\frac{X}{Z} \\ Y' = f\frac{Y}{Z} \end{cases} \tag{4}$$

while f is the focal length. The pixel coordinates of P are set as $[u, v]^T$. The relationship between $[u, v]^T$ and P' is shown in Equation (5).

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases} \tag{5}$$

α, β are the scaling coefficient on the u,v axis. $[c_x, c_y]^T$ is the translation vector of the origin. Replace $\alpha x'$ with f_x and $\beta Y'$ with f_y, This equation can be expressed in homogeneous coordinates as Equation (6)

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = KP \tag{6}$$

where K is equal to $\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$ which is also known as the internal reference matrix of the camera. P can be considered as a transformation of P_w under the current robot pose. The pose can be represented by a rotation matrix R and a translation vector t. So the Equation (6) can be converted to Equation (7) where T is equal to $\begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix}$

$$ZP_{uv} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K(RP_w + t) = KTP_w \tag{7}$$

$$P_w = K^{-1}T^{-1}ZP_{uv} \tag{8}$$

5.2. Getting the Pose of the Target Robots

What we want to do is to get R and t in Equation (7) Solving the pose of the target robot can be transformed into finding a solution to the iterative closest point (ICP) problem. Set the coordinate of features point y_i of assistant robot's 3D model as P_i and world coordinate as p_i^w. The error term for the point pair (P_i,P_i^w) is

$$e_i = p_i - (RP_i^w + t) \tag{9}$$

Then, the least squares problem is constructed, and R, t, which minimizes the sum of squared error shall be obtained.

$$min_{R,t} = \frac{1}{2}\Sigma||p_i - (Rp_i^w + t)||^2 \tag{10}$$

The following is the derivation of its solution:
First, define the centroid of two sets of points:

$$p = \frac{1}{n}\Sigma(p_i), P^w = \frac{1}{n}\Sigma(p_i^w) \tag{11}$$

$$\begin{array}{rcl} \frac{1}{2}\Sigma||p_i - (Rp_i^w + t)||^2 & = & \frac{1}{2}\Sigma||p_i - Rp_i^w - t - p + Rp^w + p + Rp^w||^2 \\ & = & \frac{1}{2}\Sigma||p_i - p - R(p_i^w - p^w) + (p - Rp^w - t)||^2 \\ & = & \frac{1}{2}\Sigma||p_i - p - R(p_i^w - p^w)||^2 + ||(p - Rp^w - t)||^2 \\ & & +2(p_i - p - R(p_i^w - p^w))^T(p - Rp^w - t) \end{array} \tag{12}$$

The objective function after the simplification is:

$$min_{R,t} = \frac{1}{2}\Sigma||p_i - p - R(p_i^w - p^w)||^2 + ||(p - Rp^w - t)||^2 \tag{13}$$

We only need to get R first and then make $p - Rp^w - t$ equal to zero to get t. Further simplification is as followed:

$$q_i = p_i - p, q_i^w = p_i^w - p^w \tag{14}$$

The objective function can be further reduced to:

$$R^* = arg\max_R \frac{1}{2}\Sigma_{i=1}^n ||q_i - Rq_i^w||^2 \tag{15}$$

Expand the error term for R, the result is shown as Equation (16):

$$\frac{1}{2}\Sigma_{i=1}^n ||q_i - Rq_i^w||^2 = \frac{1}{2}(\Sigma q_i^T q_i + q_i^{wT} R^T R q_i^w - 2q_i^T R q_i^w) \tag{16}$$

R is an antisymmetric matrix, so the result is only related to third items $-q_i^T R q_i^w$. The optimization objective function is changed to:

$$\Sigma_{i=1}^n(-q_i^T R q_i^w) = \Sigma_{i=1}^n - tr(Rq_i^w q_i^T) = -tr(R\Sigma_{i=1}^n q_i^w q_i^T) \tag{17}$$

In order to solve R, first define the matrix

$$W = \Sigma_{i=1}^n q_i q_i^{wT}$$

SVD decomposition of W is obtained:

$$W = U\Sigma V^T \tag{18}$$

When w is full rank, R is as

$$R = UV^T \tag{19}$$

Finally, we can calculate out t based on R:

$$t = p - Rp^w \tag{20}$$

6. Approximate Solution of Optimal Strategy for Chained Localization

In this section, we will discuss how to approach the optimal strategy for chained localization. The optimal strategy for chained localization is a sequence $\{a_1, a_2, ..., a^t\}$ whose cumulative reward is larger than other sequences whose ending is a^t, a^t is the robot who needs assistance. Cumulative reward is used to measure the change of localization self-confidence of all robot after chained localization. Cumulative reward is defined as:

$$V_\pi = \gamma_{12}(r_{12} + \gamma_{23}(r_{23} + \gamma_{34}(r_{34} + ...\gamma_{n-1,n} * r_{n-1,n})))|\pi = \{a_1, a_2.....a_n\} \tag{21}$$

LSC_i was determined by all the robots before i as Section 4.3 shown, so before the sequences prior to i is determined $r_{i,i+1}$ is uncertain. Two algorithms are proposed to get optimal solution based on reinforcement learning and greedy algorithm.

6.1. Algorithms Based on Reinforcement Learning

The process can be considered as to find the path whose maximum cumulative reward from the first robot to the target robot. If the update of the assistance tying matrix Γ is not considered, this process can be regarded as being provided with Markov property. Let a_{ij} represent the action that robot i observe robot j, that means the state is changed from S_i to S_j. Localization tying r_{ij} can be rewarded as immediate rewards. γ_{ij} is the discount factor. Therefore, the problem of obtaining the optimal strategy can be solved by the algorithm of reinforcement learning. This algorithm is shown as

Algorithm 2. The state representing target robot is selected as absorbing state. From step 3 to step 12, each $Q(s,a)$ is continuously changing until all of them become optimal.

The premise of applying this algorithm is to assume that the process is Markov process. But according to the analysis in Section 6, this process is not strictly following the Markov property. This method is an approximate algorithm. After each iteration, the smaller the update matrix is, the closer the strategy is to the optimal strategy. We discuss a method based on greedy algorithm without considering this approximation in the next subsection.

Algorithm 2 Algorithms based on Reinforcement Learning

input: R: The immediate payoff matrix, Γ: Observation error coefficient
input: S: Sets of all state, s_f: The state meaning the first assistant
input: P: State transition matrix robot, s_t: The robot need assistance
output: $\pi*_f$: Optimal assistance chain from s_s to s_t

1: **function** GETOPTIMALSTRATEGY(R, Γ, S, s_s, s_t)
2: **for each** $s \in S$ $a \in A$(Actions that can be taken in state s) **do**
3: $Q(s,a) = 0$
4: **end for**
5: **repeat**
6: convergence $\leftarrow 1$
7: **for each** $s \in S$ **do**
8: **for each** $a \in A$ **do**
9: $Q'(s,a) = r_{s,p(s,a)} + \gamma_{s,p(s,a)}(max_{a'}(Q(P(s,a),a')))$
10: **if then**$Q'(s,a) \neq Q(s,a)$
11: convergence $\leftarrow 0$
12: **end if**
13: $Q(s,a) = Q'(s,a)$
14: **end for**
15: **end for**
16: **until**
17: **end function**

6.2. Algorithms Based on Greedy Algorithm

The main idea of this algorithm is to search for better strategies based on the current optimal strategy. A table containing the source state's (the first state we choose for the optimal strategy) reachable states, as well as the optimal path to each state are maintained. The state j whose optimal strategy is maximum is selected and its next reachable state i is searched, if the state i is not in the optimal policy of the state j and is not in the table, the state i is added to the table. If the state i is already in the table, the reward of the policy that contains the j and that of the best policy of i are compared. If the policy containing the j is larger, the optimal policy of the j is set as the optimal strategy of j plus i which means $\pi_i \leftarrow \pi_j + j$. The concrete steps of the algorithm are shown by the Algorithm 3.

Algorithm 3 is explained in the following manner: First all the variables are initialized and (7 to 13) add the source state to the open table (14). Then get the current optimal strategy and search the better strategy based on it (16–50). Finally, output the strategy π_t.

Algorithm 3 Algorithms based on Greedy Algorithm

input: T_o: The initial matrix of localization tying , Γ: Observation error coefficient
input: A: Sets of allg agent, a_s: The first assistant robot, a_t: The robot need assistance
output: π_t: Optimal assistance chain from a_s to a_t

function GETOPTIMALSTRATEGY(R, A, a_s, a_t)
    Π: π_i is a list containing optimal Strategy from a_S to a_t
    V: v_i is Cumulative reward when the best strategy π_i is taken
    T: T_i is the matrix of localization typing when the best strategy π_i is taken ($i > 0$)
    open: An list containing the *a*s which can be visited next
    $open \leftarrow []$
    i$\leftarrow$ 0
    repeat
        $convergence = 1$
        V_i = 0
        Π_i = []
        R_i = R
        i$\leftarrow$ i + 1
    until $i = n$
    Add (*open*,*s*)
    repeat
        a_j = open.head
        i$\leftarrow$ 0
        repeat
            if $R_j.r_{ji} \neq 0$ and $i \notin \pi_j$ **then**
                Add (*open*,*i*)
                $convergence \leftarrow 0$
                if $\pi_i = []$ **then**
                    $v_i = \gamma_{ji}(R_j.r_{ji})$
                    R_i.update (R_j)
                    Add (π_j,j)
                else
                    $\pi*$ = Insert (π_j,*i*,$\pi_j.tail$)
                    $v*$ = Value ($\pi*$,R_j)
                    if $v* > v_i$ **then**
                        $convergence \leftarrow 0$
                        $\pi_i = \pi*$
                        R_i.update (R_j)
                        $k \leftarrow 0$
                        repeat
                            if $j \in \pi_k$ **then**
                                $V_i \leftarrow 0$
                                $\Pi_i \leftarrow []$
                                $R_i \leftarrow R$
                            end if
                        until $k = n$
                    end if
                end if
            end if
        until $i = n$ or convergence = 0
        if convergence = 1 **then**
            Delete (*open*,*j*)
        end if
        open.sort()
    until convergence = 1
end function

7. Sequences of Collaborative Algorithm

In this section, the whole sequences of collaborative algorithm and the communication process between organiser and robots will be introduced.

Each robot has two switchable modes: independent mode and collaborative mode. Robot can explore and SLAM freely in independent mode, but once it needs to assist other or be assisted, it will switch to the collaborative mode at the request of organiser until the collaboration is completed and can be switched back. To achieve this goal, each robot has to run two threads at the same time: motion thread and listen thread. The whole sequences of algorithm running on all of robots' clients is shown as Algorithm 4 and it is explained in following manner: First initialize all the parameters (02–06). Then the robots can judge whether they can move freely or not. In the process of exploration, the visual odometer and bundle adjustment are running at the same time. The pose map is uploaded to organiser at set intervals (09). The robot needs to judge whether assistance is needed or not (10). If the assistance is not necessary it moves and updates the parameters ifself (12–20), otherwise it will ask organiser for help and block up there to wait for a response (22–23). If it gets a command to assist others, he will arrive at the designated position and observe the specified target, and finally return the observation or information "mission failture" to organiser (26–30). When it comes to the listen thread, robots will determine its status based on information obtained from organiser. If one robot needs to assistant others, it will get the path and target (36–39). If the command is "queuing", the robot will wait in place, in order to ensure that there is no unlimited waiting, the robot will cancel the request and continue to move freely after more than a certain time, but the level of his need for help will increase (40–48). If the others are "being assistanted", it will be waiting for other robot assistance in place (49–50). Robot can aslo obtain pose and lsc calculated by organiser as the prior information visual odometer and bundle adjustment (51–53). After the chained localization is execued, robots must upload its own parameters as required (54–55).

The main function of organiser is to handle the request queue, to get the optimal policy for each request, and to monitor the execution process of the policy. Therefore, two threads are needed to implement the function: listen request thread. The sequences of this algorithm is shown as Algorithm 5 and it is explained in following manner: Organiser determines whether the current request needs to wait or not and notify the relevant robot according to the queue's condition (5–11). If the requirement is delete specified request, then request is deleted (12–13). When processing requests, first, all requests in the queue are sorted according to the assistant level and get the first request (16–17). Then organiser ask all robots except target robot for the information and calculate the optimal algorithm (18–30). For each robot in the optimal strategy, the organiser send their goals and move paths (31–36). After all robot replying, the organiser send the parameters that need updating for each robot (37–40). The communication process between the organiser and the robot is shown in Figure 3 where parameters are *level* and *lsc*.

Algorithm 4 Sequences of robots

```
for each a ∈ A do
    a.lsc ← 0
    a.Waitstep ← 0
    a.waittime ← 0
    a.x ← 0
    a.level ← 0
    while a.isrunning = True do
        Motion thread:
        if a.randomMoving = True then
            if a.lsc > needAssistance or x > Waitstep then
                a.freeExplore()
                a.lsc ← η ∗ a.lsc
                if a.Waitstep ≠ 0 then
                    a.x ← a.x + 1
                end if
                if robot detects closed looping then
                    a.lsc ← 1, a.level ← 0
                    a.x ← 0
                    a.Waitstep ← 0
                end if
            else
                sendrequest (a.lsc, a.postion, a.level)
                waitresponse()
            end if
        else
            if a.moving (a.path) then
                a.sendmessage (a,a.observe(a.target))
                a.path ← Null
                a.target ← Null
            else
                a.sendmessage (a,Fail)
            end if
        end if
        Listen tread:
        response = getResponse()
        if response = assistantOthers then
            a.path ← response.path
            a.target ← response.path
            a.randomMoving ← False
        else if response = queuing then
            a.starttiming()
            a.randomMoving← False
            if a.timing() > a.waittime then
                sendrequest(delete,a)
                a.randomMoving ← True
                a.level ← a.level + 1
                a.waitstep = Waitstep
            end if
        else if response = waitforAssistant then
            a.randomMoving = False
        else if response = updatePose then
            a.pose = response.pose
            a.lsc = response.lsc
        else if response = askInformation then
            sendmessage(a)
        end if
    end while
end for
```

Algorithm 5 Sequences of organiser

```
organiser.queue = []
while organiser.isrunning = True do
    listen request thread:
    request = getRequest()
    if request.type = askAssistant then
        if queue.empty = False then
            queue.add (requst.a)
            sendmessage (a,queuing)
        else
            sendmessage (a,waitforAssistant)
        end if
    else if request.type = deleteRequest then
        queue.delete (request.a)
    end if
    handle request thread:
    queue.sort (level)
    a = queue.head()
    for each a' ∈ A − a do
        sendmessage (askInformation)
    end for
    waitforResponse()
    π∗.path ← Null
    π∗.Value ← 0
    for each a' ∈ A do
        if a'.lsc > a.lsc + θ then
            if  thenπ∗.value < getOptimalStrategy(R, A, Π, a, a').value
                π∗ ← getOptimalStrategy(R, A, Π, a, a')
            end if
        end if
    end for
    for each a ∈ π∗ do
        sendmessage (a,a.target,a.path)
        if a ∈ queue then
            queue.delete(a)
        end if
    end for
    waitforResponse()
    for each a ∈ π∗ do
        sendmessage(a,a.pose,a.lsc)
    end for
end while
```

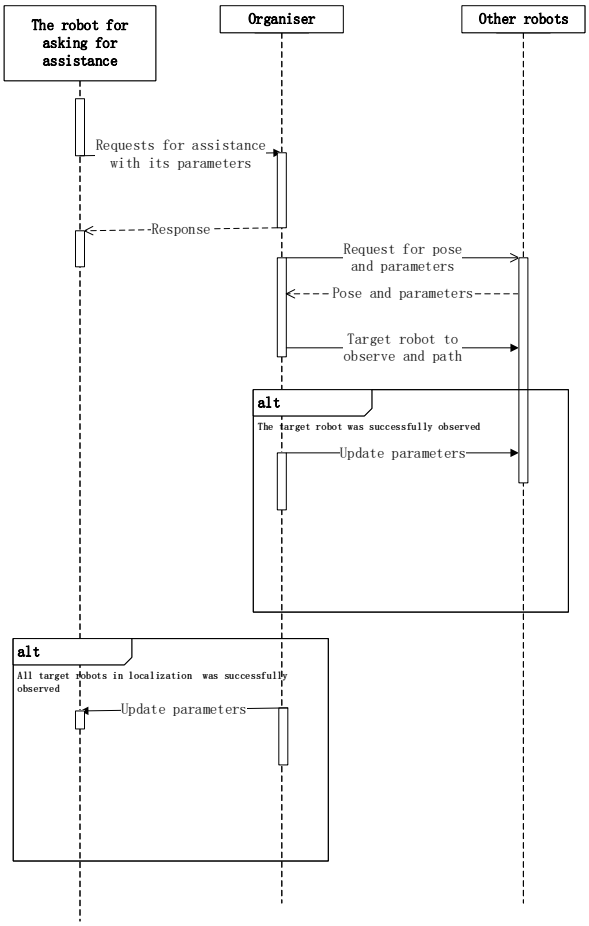

Figure 3. Sequences of communication between organiser and robots.

8. Experiment

The goal of the simulation experiments described in this section is to evaluate the performance of the proposed cooperative SLAM algorithm under different parameters. The experiment was implemented on the robot operating system (ROS) platform whose distributed design is very suitable for simulating MRS. Gazebo is a ROS package used to simulate the real-world, including indoor environment and outdoor environment. The experimental environment is ROS system running on distributed computing cluster. 'Gazebo' and organiser are deployed to one node, and the simulation program of six robots are deployed evenly on the other three nodes. Network delays can be used to simulate communication delays. We build one simulation robots using the 'darwin-op' model with a deep camera in its head and the control API of 'darwin-op'. The initial pose of each robot is known and "orbslam2" [25] which is an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras runs independently on each robot. The experimental world is shown as Figure 4:

The background of our experiment is that multiple humanoid robots explore unknown maps. So the main indicators to measure the implementation of the task are robots position RMSE, orientation RMSE and main landmark position RMSE. The effect of cooperative localization between robots can be reflected under such experimental background.

The experiments have been done with $\eta = 0.996$, $\gamma = 0.93$, $\theta = 0.35$ (robot will ask for assistance when LSC is lower than it). The time step is set as 5 s.

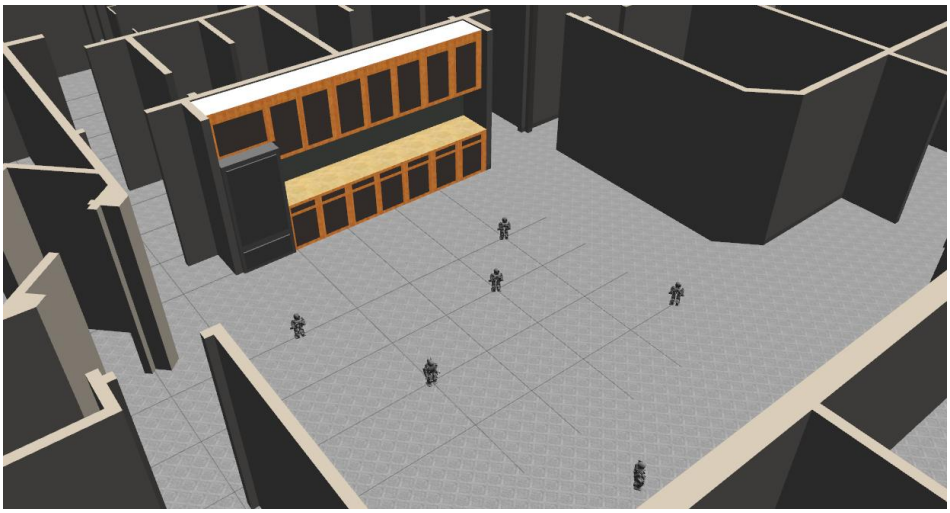

Figure 4. Experimental world.

Experiment 1. This experiment verify the effectiveness of the proposed algorithm compared with relative method. The Table 1 shows the result of three different ways which are explained below after 30 experiments for each case:

- Case 1: Cluster matching algorithm in the environment without distance IR sensor referred to in literature [8]. But we made some changes to that algorithm to consistent with our experiment background. We assume that there is no IR sensor in the environment, and the target robot that needs help is an "invisible robot".The cluster matching algorithm proposed in literature 3 is used to get the cluster head and the neighbors of the invisible robot which is also the assistant robots.
- Case 2: The proposed method with optimal algorithm based on reinforcement learning.
- Case 3: The proposed method with optimal algorithm based on greedy algorithm.

Table 1. Total comparison result.

	Robots Position RMSE (m)	Orientation Position RMSE (deg)	Landmark Position (m)
Case 1	0.3246	7.5352	0.3386
Case 2	0.2573	5.3498	0.2477
Case 3	0.2392	4.8344	0.2238

The proposed algorithm reduce robots position RMSE (m) by 25.30% Orientation position RMSE (deg) by 35.84% and Landmark position RMSE (m) by 33.90% compared with cluster matching algorithm in the best case. The development of these three indicators over time are shown in Figures 5–7 respectively. The cluster matchings algorithm does not take into account the accuracy of the pose of the observer itself, especially without the help of IR sensor. However the proposed algorithm determines the strategy of cooperation according to LSC of each robot which can make the advantage of collaboration better.

Experiment 2. This experiment shows the effects of different parameters and different optimal strategies on proportion of coordination. Active cooperative localization improves the efficiency of correction of robots' pose and landmarks at the expense of the efficiency of MRS in exploring new landmarks. The cost of MRS in active collaborative localization can be measured by the distance of the robots covered in the process. The more journey it is, the greater the possibility of the robot to find new landmarks is if they covered that distance in free walk, and the greater the cost is. Therefore, The variable used to compare is percentage of the distance robots cover for collaboration. The Figure 8 shows the result of three different ways which are explained:

- Case 1: The proposed method with optimal algorithm based on greedy algorithm.
- Case 2: The proposed method with optimal algorithm based on reinforcement learning.

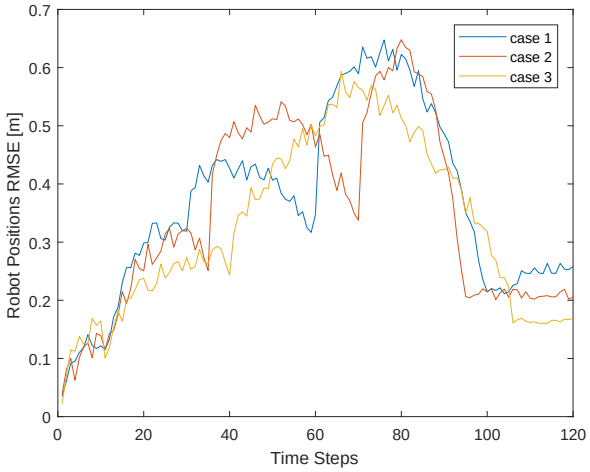

Figure 5. Robots position RMSE.

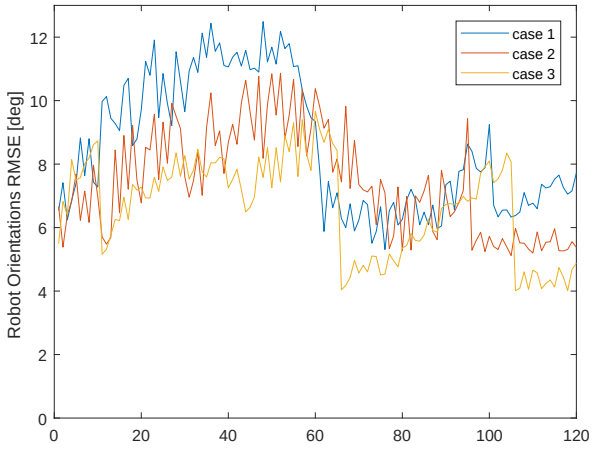

Figure 6. Orientation position RMSE (deg).

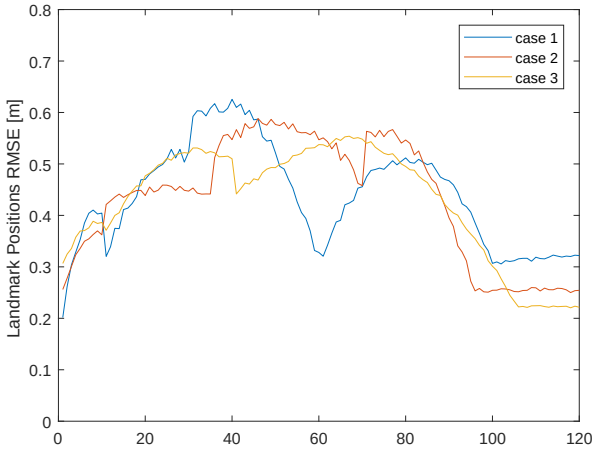

Figure 7. Landmark position RMSE (m).

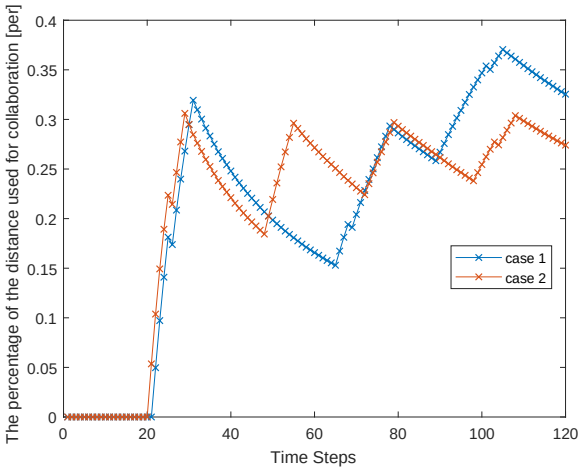

Figure 8. Percentage of the distance robots cover for collaboration with different optimal algorithm.

Different θ will also affect the process of collaboration. Figure 9 shows the consumption of collaboration where case 1 case 2 case 3 respectively take θ as 0.2, 0.35, 0.6. The experimental results show that when the greedy algorithm and the theta value are larger, the consumption of collaboration is larger, so although the effect of collaborative positioning is improved, it will affect the exploration and construction of MAS.

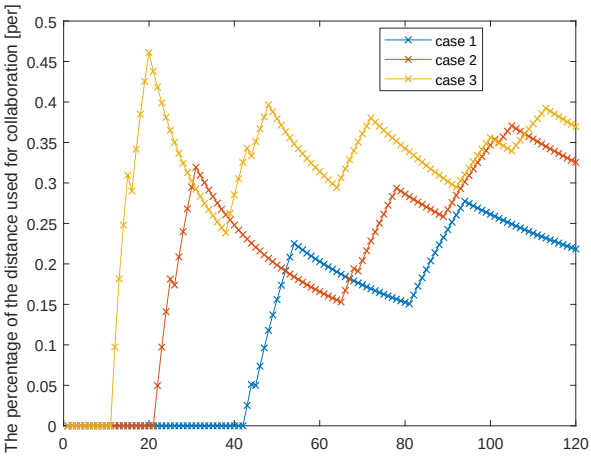

Figure 9. Percentage of the distance robots cover for collaboration with different optimal algorithm.

9. Conlusions

This paper proposed a relative localization method called chained localization for humanoid multi-robot SLAM. Robots in the proposed MAS are provided with an extra attribute: LSC which can be updated when they are exploring or the closed loop is detected or the collaborative localization is completed. Besides, We also propose two algorithms for obtaining optimal strategies for chained localization based on reinforcement learning and greedy algorithm. Compared with other algorithms, each robot can share its accumulated localization advantage which can reduce dependence on closed loop detection and static landmark. The proposed mechanism and algorithms is suitable for application to large scale environment without static landmark, global localization sensors. However, This algorithm can only be used when the initial absolute pose of the robot is known, because the global map and the local map can only be consistent through the pose of the robot. Finally, we designed and did simulation experiments to compare our proposed methods with the state of the art relative algorithm. The proposed algorithm reduce robots position RMSE (m) by 25.30% orientation position

RMSE (deg) by 35.84% and landmark position RMSE (m) by 33.90% compared with cluster matching algorithm which is drastic improvement. In the future work, we plan to apply machine learning to the multi-robot collaboration SLAM, which means trying to adjust some of the key parameters in the cooperative mechanism according to the effect of the SLAM.

Author Contributions: Conceptualization, Z.P.; Methodology, Z.P., M.E.H.S. and M.Z.Q.; Software, Z.P.; Validation, S.P.; Investigation, Z.P.; Data Curation, G.L.; Writing—Original Draft Preparation, Z.P.; Writing—Review & Editing, S.P., M.Z.Q.; Visualization, M.Z.Q.; Supervision, S.P.; Project Administration, S.P.; Funding Acquisition, S.P.

Funding: This paper is supported by National Natural Science Foundation of China [grant number 61375081]; a special fund project of Harbin science and technology innovation talents research [grant number RC2013XK010002].

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

References

1. Papadimitriou, F. Artificial Intelligence in Modelling the Complexity of Mediterranean Landscape Transformations. *Comput. Electron. Agric.* **2012**, *81*, 87–96. [CrossRef]
2. Lamastra, I.; Balderacchi, M.; Di Guardo, A.; Monchiero, M.; Trevisan, M. A novel fuzzy Expert System to assess the Sustainability of the viticulture at the wine-estate scale. *Sci. Total Environ.* **2016**, *572*, 724–733. [CrossRef] [PubMed]
3. Ducan, B.; Ulam, P. Behavior as a model for multi-robot systems. In Proceedings of the 2009 International Conference on Robotics and Biomimetics (ROBIO), Guilin, China, 19–23 December 2009; pp. 25–32.
4. Hayat, S.; Yanmaz, E.; Brown, T.X.; Bettstetter, C. Multi-objective UAV path planning for search and rescue. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 5569–5574.
5. Kurdi, H.; How, J.; Bautista, G. Bio-Inspired Algorithm for Task Allocation in Multi-UAV Search and Rescue Missions. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, USA, 4–8 January 2016.
6. Hajjdiab, H.; Laganière, R. Vision-Based Multi-Robot Simultaneous Localization and Mapping. *Conf. Comput. Robot Vis.* **2004**, *8345*, 155–162,
7. Martinelli, A.; Pont, F.; Siegwart, R. Multi-Robot Localization Using Relative Observations. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005.
8. Rashid, A.T.; Frasca, M. Multi-robot localization and orientation estimation using robotic cluster matching algorithm. *Robot. Auton. Syst.* **2015**, *63*, 108–121. [CrossRef]
9. Atanasov, N.; Le Ny, J.; Daniilidis, K.; Pappas, G.J. Decentralized Active Information Acquisition:Theory and Application to Multi-Robot SLAM. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA) Washington State Convention Center Seattle, Washington, DC, USA, 26–30 May 2015.
10. Mehrez, M.W.; Mann, G.K.I. An Optimization Based Approach for Relative Localization and Relative Tracking Control in Multi-Robot Systems. *J. Intell. Robot. Syst.* **2016**, *85*, 1–24. [CrossRef]
11. Fox, D, Burgard, W. A Probabilistic Approach to Collaborative Multi-Robot Localization. *Auton. Robots* **2000**, *8*, 325–344. [CrossRef]
12. Silva, O.D.; Mann, G.K.I. Efficient distributed multi-robot localization: A target tracking inspired design. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 434–439.
13. Khorshidi, A.; Shahri, A.M.; Oskoei, M.A. Evolutionary Particle Filter Applied to Leader-Labor Multi-robot Localization for Communication Failure and Kidnapped Situations. In Proceedings of the 4th International Conference on Robotics and Mechatronics, Tehran, Iran, 26–28 October 2016.
14. Al Hage, J.; El Najjar, M.E.; Pomorski, D. Fault Tolerant Collaborative Localization for Multi-Robot System. In Proceedings of the 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 21–24 June 2016.
15. Todescato, M.; Carron, A. Multi-robot localization via GPS and relative measurements in the presence of asynchronous and lossy communication. In Proceedings of the Control Conference, Aalborg, Denmark, 29 June–1 July 2016; pp. 2527–2532.

16. Wu, M.; Huang, F.; Wang, L.; Sun, J. Cooperative Multi-Robot Monocular-SLAM using Salient Landmarks. In Proceedings of the International Asia Conference on Informatics in Control, Bangkok, Thailand, 1–2 February 2009; pp. 151–155.
17. Deutsch, I.; Liu, M.; Siegwart, R. A Framework for Multi-Robot Pose Graph SLAM. In Proceedings of the 2016 IEEE International Conference on Real-time Computing and Robotics, Angkor Wat, Cambodia, 6–9 June 2016.
18. Schuster, M.J.; Brand, 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.
19. Kai, M.W.; Stachniss, C. Bridging the gap between feature- and grid-based SLAM. *Robot. Auton. Syst.* **2010**, *58*, 140–148.
20. Forster, C.; Lynen, S.; Kneip, L.; Scaramuzza, D. Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, Tokyo, Japan, 3–7 November 2013; pp. 3962–3970.
21. Koch, P.; May, S.; Schmidpeter, M.; Kühn, M.; Pfitzner, C.; Merkl, C.; Koch, R.; Fees, M.; Martin, J.; Ammon, D.; et al. Multi-Robot Localization and Mapping Based on Signed Distance Functions. *J. Intell. Robot. Syst.* **2016**, *83*, 409–428. [CrossRef]
22. Milford, M.; Wyeth, G. Hybrid robot control and SLAM for persistent navigation and mapping. *Robot. Auton. Syst.* **2010**, *58*, 1096–1104. [CrossRef]
23. Liang, Z.; Zhu, S.; Fang, F.; Jin, X. Simultaneous Localization and Mapping in a Hybrid Robot and Camera Network System. *J. Intell. Robot. Syst.* **2010**, *24*, 1–24. [CrossRef]
24. Stentz, A. Optimal and efficient path planning for partially-known environments. In Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; Volume 4, pp. 3310–3317.
25. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]