A Comprehensive Cooperative Exploration Framework for Ground and Air Vehicles in Unknown Environments

Hai Ngoc Pham and Salah Sukkarieh
ARC Centre of Excellence for Autonomous Systems
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney, 2006 NSW
h.pham, salah@cas.edu.au

Abstract

This paper represents the development of a comprehensive cooperative exploration frame work in which each individual vehicle has the ability to explore an unknown environment by itself and also by cooperative behaviour in a team of several vehicles that consist of ground and air vehicles. Our approach to exploration strategy focuses on solving optimal assignment problem amongst the team by using linear integer programming technique in order to maximize the team's benefit and minimize the time for exploration. The simulation results show the effectiveness of the proposed framework for multi-vehicle exploration in unknown environments.

1 Introduction

In the robotics field, exploration has been defined as the process of effectively covering an unknown environment by a single or a group of autonomous vehicles in order to gain as much information about the world as possible in a reasonable amount of time [Yamauchi, 1997]. There has been extensive work in this area such as [Hayati *et al.*, 1996], [Yamauchi, 1997; Yamauchi, 1998], [Burgard *et al.*, 2000], [Simmons *et al.*, 2000], [Williams *et al.*, 2001], [Makarenko *et al.*, 2002], [Bourgault *et al.*, 2002], [Newman *et al.*, 2003], [Ko *et al.*, 2003], [Thrun *et al.*, 2004], [Burgard *et al.*, 2005], and [Bryson and Sukkarieh, 2006]. They have been aiming to solve exploration problems that focus on different contexts of vehicle architecture such as Mars exploration robots, indoor robots, ground vehicles, and air vehicles; as well as different assumptions and environment conditions. In this present paper the aim of solving exploration problems focuses on proposing a cooperative exploration framework for ground and air vehicles to explore unknown environments.

In [Cao *et al.*, 1997], the authors have raised the interest in systems composed of multiple autonomous mobile robots. Their statements obviously show that effective team-work behaviour by a team of autonomous vehicles will lead to the team completing their mission quicker and/or with more accuracy than relying on a single vehicle. Thus solving problems of team working amongst several autonomous vehicles must be considered [Yamauchi, 1998], [Burgard *et al.*, 2000], [Simmons *et al.*, 2000], [Williams *et al.*, 2001], [Ko *et al.*, 2003], and [Burgard *et al.*, 2005].

In addition, the mission of gaining as much knowledge/information as possible about the unknown world while minimizing any cost to the team will be essential. That involves the necessity of constructing a good "mechanism of cooperation" or "cooperative behaviour" for the team in order to improve the performance of several autonomous vehicles in the team to gain maximum total utility of the system [Cao *et al.*, 1997].

This present paper answers the requirements of building a *complete structure for each autonomous vehicle* that consists of all the important components such as: local low-level and high-level controllers; local dynamic path planning with the ability of obstacle and collision avoidance; local position estimation that can fuse information under vehicle model and environment uncertainties; and cooperative behaviour in dealing with teamwork.

This present research also focuses on answering the question of *how can a group of autonomous vehicles allocate destinations to visit in order to maximize the team and individual benefits.* In this context, the *optimal assignment problem*[1] is essential in which the team members will need to negotiate to allocate potential travel destinations in order to find out the "equilibrium point" (a well-known idea proposed by John F. Nash in [Nash, 1950; Nash, 1953]) that maximizes the whole team's benefit.

The paper is organized as follows. Section 2 presents

[1]The *Optimal Assignment Problem* is a world known problem that was originally researched in game theory by John von Neumann in 1947 [Neumann, 1963]

a summary of work related to exploration. Section 3 demonstrates our approach in constructing a cooperative exploration framework for multi-vehicles in unknown environments. Section 4 depicts results of simulation. Finally, Section 5 summarizes with a conclusion and lists possible ideas for future research.

2 Related Works

There has been a great deal of research on developing good exploration strategies in order to gain as much new information about an environment as possible in a reasonable amount of time.

The work in [Yamauchi, 1997; Yamauchi, 1998] focused on developing a frontier-based exploration technique for identifying the most important areas (frontiers) that the robot - group of robots can visit. This technique has been widely applied in research on exploration of single and multiple autonomous robots in unknown environments. The key idea in this approach lies in the fact that the boundary between open space and unexplored regions will be the areas of most interest and ones that would contribute the most new information about the world. The technique defines a frontier as a region on that boundary and when a robot is moving to successive frontiers, it can constantly gain more knowledge of the environment. The proposed exploration strategy in this present paper is also based on that frontier-based exploration approach.

The researches of [Burgard *et al.*, 2000; Simmons *et al.*, 2000; Burgard *et al.*, 2005] looked at exploration problems in unknown environments by a team of indoor robots that focused on developing an on-line mapping algorithm for likelihood maximization and a coordinated exploration strategy between multiple robots to maximize the overall utility. The key problem addressed in these researches was developing algorithms for allocating appropriate target points for a team of mobile robots in order to maximize the overall benefit of the team. Extensive simulations have been conducted in indoor environments by the authors of those papers in order to prove their proposed algorithms. This present paper's direction is rather close to their research in that it involves exploration in unknown environments by a team of vehicles, however, our paper proposes another algorithm to solve the task assignment problem for allocating potential destinations to multiple vehicles in order to maximize the team's benefit. In addition, the research conducted in our paper focuses on another category of autonomous vehicles such as ground and air vehicles that will have different constraints in both the physical and mathematical models.

In [Makarenko *et al.*, 2002], the authors used a single mobile robot to explore a partially unknown feature map and a partially known occupancy map for examining an integral exploration strategy in a decentralized architecture (Active Sensor Network[Makarenko *et al.*, 2003] and [Makarenko, 2004]). Their strategy focused on maximizing the map quality and minimizing overall exploration time. The approach in their research tried to integrate motion control, localization and map building together by using a SLAM point feature map[Dissanayake *et al.*, 2001] and a occupancy grids map[Elfes, 1989]. The research direction in this present paper can also be regarded as an extension of the research of [Makarenko *et al.*, 2002] in the case of multiple mobile robots exploring an unknown environment.

For developing an exploration strategy of a 6-DOF uninhabited aerial vehicle (UAV) using inertial navigation range/bearing sensors in unknown environments, the authors in [Bryson and Sukkarieh, 2005] and [Bryson and Sukkarieh, 2006] have contributed to fusing concepts in ground-based exploration strategies with path planning and guidance scheme for maximizing navigation accuracy through the unmapped feature terrain of an airborne 6-DOF UAV. In their research, the airborne SLAM algorithm[Kim and Sukkarieh, 2003] and [Kim, 2004] was used for map building and vehicle's pose estimating. The information-theoretic guidance scheme was applied to maximize the total utility of visiting each available destination in the list of potential destinations proposed by making observations of the environment upon the UAV reaching its waypoint. This would be the most appropriate future direction of this present paper in integrating the airborne 6-DOF SLAM algorithm in individual air vehicle in the team of vehicles exploring unknown environments.

3 Cooperative Exploration Framework for Multiple Vehicles in Unknown Environments

3.1 Architecture Overview

The flowchart diagram of the decentralized cooperative exploration framework proposed in this paper is represented in Figure 1.

The team of vehicles with known initial positions will initialize their tasks by initially observing their surrounding environment. The team will then construct the initial explored map of the environment by convolution of all the initial observed maps from each vehicle. The task assignment problem will be solved at this stage by one vehicle in the team assigning each individual vehicle to go to one destination. Each individual member will locally plan its own path to go to that destination and it will then autonomously travel to that goal under the local guidance and navigation modules represented in Section 3.2, with the assumption that each vehicle can localize itself by using an on-board GPS module.

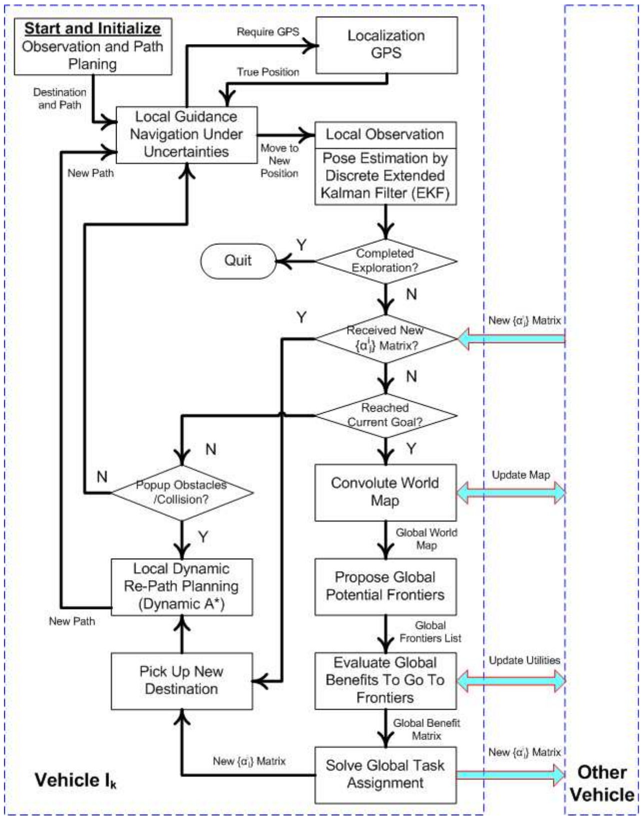

Figure 1: Decentralized cooperative exploration architecture for multiple vehicles in unknown environments.

The team then cooperatively explores the unknown environment under the proposed algorithm illustrated in the above flowchart diagram. The next section will focus on describing all proposed components of a complete autonomous vehicle.

3.2 Local Guidance Under Plant and Sensor Models' Uncertainty

This component integrates the well-known Extended Kalman Filter (EKF) algorithm in predicting and estimating the vehicle process's states for both the ground and air vehicles by minimizing the mean squared estimation error of the states given an observation at the previous time step[Durrant-Whyte, 2001].

Ground Vehicle Model

The geometry and kinematic models of the UTE Car from ACFR[Guivant, 2002] and [Nieto, 2005] shown in Figure 2 are used in this paper.

- The discrete-time representation of the transition motion of the UTE car at the laser point can be approximated as:

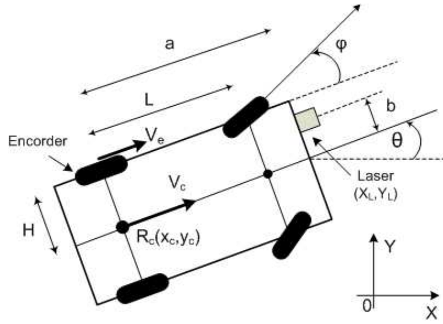

Figure 2: Geometry representation of ground vehicle model - UTE car from ACFR.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \Delta t[v_c\cos(\theta_k) - \frac{v_c}{L}(a\sin(\theta_k) + b\cos(\theta_k))\tan(\phi_{k+1})] \\ \Delta t[v_c\sin(\theta_k) + \frac{v_c}{L}(a\cos(\theta_k) - b\sin(\theta_k))\tan(\phi_{k+1})] \\ \Delta t\frac{v_c}{L}\tan(\phi_{k+1}) \end{bmatrix} \quad (1)$$

$$+ G(k+1)\begin{bmatrix} q_v(k+1) \\ q_\phi(k+1) \end{bmatrix}$$

where $\begin{bmatrix} q_v(k+1) & q_\phi(k+1) \end{bmatrix}^T$ is the additive Gaussian noise vector on the encoder velocity and steering angle of the control input vector $U(k)$.

- Jacobian of the non-linear process transition matrix:

$$\nabla\mathbf{f} = \begin{bmatrix} 1 & 0 & -\Delta t(v_c\sin(\theta) + \frac{v_c}{L}\tan(\phi)T_2) \\ 0 & 1 & \Delta t(v_c\cos(\theta) - \frac{v_c}{L}\tan(\phi)T_1) \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where

$$\begin{aligned} T_1 &= a\sin(\theta) + b\cos(\theta) \\ T_2 &= a\cos(\theta) - b\sin(\theta) \end{aligned}$$

- Source noise transition matrix:

$$\mathbf{G}(k+1) = \begin{bmatrix} \frac{\partial\mathbf{F_1}}{\partial v_e} & \frac{\partial\mathbf{F_1}}{\partial\phi} \\ \frac{\partial\mathbf{F_2}}{\partial v_e} & \frac{\partial\mathbf{F_2}}{\partial\phi} \\ \frac{\partial\mathbf{F_3}}{\partial v_e} & \frac{\partial\mathbf{F_3}}{\partial\phi} \end{bmatrix} \quad (3)$$

where the partial derivatives of each component are:

$$\begin{aligned}
\frac{\partial \mathbf{F}_1}{\partial v_e} &= \Delta t[\cos(\theta) - \frac{\tan(\phi)}{L}T_1]\frac{\partial v_c}{\partial v_e} \\
\frac{\partial \mathbf{F}_1}{\partial \phi} &= -T_1\frac{v_c}{L}\cos(\phi)^{-2} + (cos(\theta) - \frac{\tan(\phi)}{L}T_1)\frac{\partial v_c}{\partial \phi} \\
\frac{\partial \mathbf{F}_2}{\partial v_e} &= \Delta t[\sin(\theta) + \frac{\tan(\phi)}{L}T_2]\frac{\partial v_c}{\partial v_e} \\
\frac{\partial \mathbf{F}_2}{\partial \phi} &= -T_2\frac{v_c}{L}\cos(\phi)^{-2} + (cos(\theta) - \frac{\tan(\phi)}{L}T_1)\frac{\partial v_c}{\partial \phi} \\
\frac{\partial \mathbf{F}_3}{\partial v_e} &= \Delta t\frac{\tan(\phi)}{L}\frac{\partial v_c}{\partial v_e} \\
\frac{\partial \mathbf{F}_3}{\partial \phi} &= \frac{v_c}{L}\cos(\phi)^{-2} + \Delta t\frac{\tan(\phi)}{L}\frac{\partial v_c}{\partial \phi}
\end{aligned}$$

Aircraft Lateral Model

Let us consider Figure 3 which shows the lateral motion representation of an UAV (unmanned air vehicle) in the 2-D horizontal plane. It is assumed that the aircraft is equipped with an autopilot system that can hold altitude and accept heading rate commands ($\omega_{cmd} \triangleq \dot{\psi}$) from a high level controller as a control input.

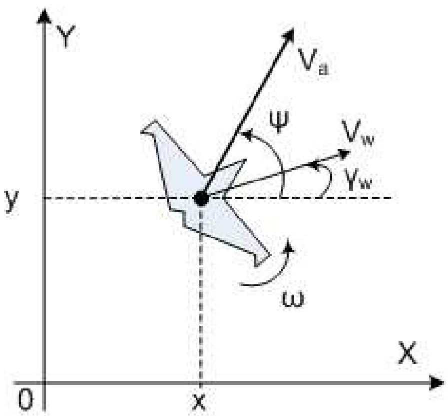

Figure 3: Kinematic of aircraft lateral motion in 2-D horizontal plane. ψ and ω are the heading and turning rate of the heading of the aircraft, respectively. Wind flows with velocity V_w and orientation γ_w.

- Non-linear kinematic for aircraft lateral motions:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \psi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta t V_k \cos(\psi_k) \\ y_k + \Delta t V_k \sin(\psi_k) \\ \psi_k + \Delta t \omega_{k-1} \end{bmatrix} + \mathbf{G}(k) \begin{bmatrix} q_v(k) \\ q_\omega(k) \end{bmatrix} \tag{4}$$

where $\begin{bmatrix} q_v(k) & q_\omega(k) \end{bmatrix}^T$ is the additive Gaussian noise vector on the velocity and heading rate control inputs.

- Jacobian of the non-linear process transition matrix:

$$\nabla \mathbf{f}(k) = \begin{bmatrix} 1 & 0 & -\Delta t V_k \sin(\psi_k) \\ 0 & 1 & \Delta t V_k \cos(\psi_k) \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

- Source noise transition matrix:

$$\mathbf{G}(k) = \begin{bmatrix} \Delta t \cos(\psi_k) & -\Delta t V_k \sin(\psi_k) \\ \Delta t \sin(\psi_k) & \Delta t V_k \cos(\psi_k) \\ 0 & \Delta t \end{bmatrix} \tag{6}$$

- Observation transformation: the sensor is mounted on the aircraft's platform and will be positioned to be downward looking onto the earth's surface. This requires both the rotation and translation of the observation to any particular point located on the earth's surface in order to provide accuracy observation information in the global reference frame.

  The cosine rotation matrix C_e^s converting coordinates from global frame to sensor frame attached in vehicle platform with yaw, pitch, and roll angles of ψ, θ, and ϕ, respectively, is:

$$\mathbf{C_e^s} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

  As we assumed that the UAV is maintained at a constant altitude, then the pitch angle can be neglected. The cosine rotation matrix is now:

$$\mathbf{C_e^s} = \begin{bmatrix} c_\psi & -s_\psi c_\phi & s_\psi s_\phi \\ s_\psi & c_\psi c_\phi & -c_\psi s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \tag{7}$$

3.3 Local Dynamic Path Planning in an Unknown Environment

In an unknown environment, it is necessary that a vehicle could be able to re-plan its trajectory to have a better one due to the changes in its surrounding environment such as popup obstacles obstruct its current trajectory.

The dynamic path planning algorithm is constructed as shown in Figure 4, in which the $\mathbf{A}^*$ algorithm [Nilsson, 1980] is implemented to build the shortest route through the nodes. The path generation module will be applied to create a smooth trajectory through those nodes, and the vehicle will be tracked to follow this trajectory by the path tracking module. The updated environment from the vehicle's sensor will then be used for re-planning the trajectory by the dynamic $\mathbf{A}^*$ algorithm when necessary.

The $\mathbf{A}^*$ algorithm is one of most useful heuristic graph-search methods. The heuristic information (an evaluation function) is denoted as $f(n)$, and is defined as an estimate of the cost to go from the starting node, through node n to the goal node:

$$f(n) = g(n) + h(n) \tag{8}$$

where $g(n)$ is the cost function to go from the starting node to node n, and $h(n)$ is the estimated cost to go from node n to the goal node.

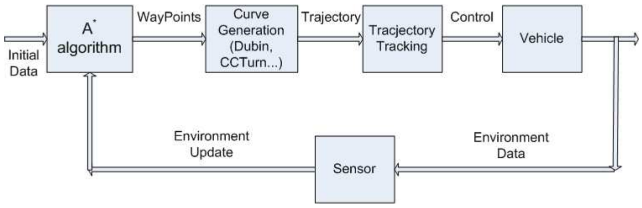

Figure 4: Dynamic $\mathbf{A}^*$ and guidance for vehicle dynamic path planing and following.

One of the most simplest approaches for estimating the heuristic $h(n)$ is the *Manhattan heuristic* that can be represented as the following equation:

$$h(n) = D_1 * (|n.x - G.x| + |n.y - G.y|) \tag{9}$$

where $(n.x, n.y)$ and $(G.x, G.y)$ are the coordinates of the two nodes, and D_1 is the minimum cost to move from one node to its adjacent node.

If the route can be connected by diagonal segments between two adjacent nodes, then the cost of moving diagonally should be different as $D_2 = \sqrt{2}D_1$, and the heuristic should be evaluated through the following *Diagonal heuristic*[Patel, 2005]:

$$\begin{aligned} h_1(n) =& |n.x - G.x| + |n.y - G.y| \\ h_2(n) =& \min\left(|n.x - G.x|, |n.y - G.y|\right) \\ h(n) =& D_1 * (h_1(n) - 2 * h_2(n)) + D_2 * h_2(n) \end{aligned} \tag{10}$$

Because the cost of diagonal moving between two adjacent nodes is obviously smaller than the cost of moving between those nodes via two orthogonal moves, then the diagonal heuristic is better than the Manhattan heuristic.

The implementation of the $\mathbf{A}^*$ algorithm is based on [Rabin, 2004], in which, the C++ object-oriented approach is applied. However, the implementation of the algorithm in this paper is developed in Matlab C/C++ Mex environment that allows real-time transfer of data between $\mathbf{A}^*$ routine and Matlab's program files, and also allows calling the $\mathbf{A}^*$ implementation in Matlab's environment.

3.4 Map Management for Multi-Vehicles in an Unknown Environment

- Map Representation: The idea of constructing a map is based on the use of the dynamic $\mathbf{A}^*$ implementation described in Section 3.3 and the concept of *Occupancy Grids* for mobile robot perception and navigation proposed by A. Elfes[Elfes, 1989]. The map is represented as a two-dimensional occupancy grids map in which each cell is represented by its utility (or value, or probability from the perception point of view) to be used to indicate its occupancy state such as: *empty* - free cell that can be travelled through; and *occupied* - obstacle. In addition, the special cells namely *features* can also be represented for indicating the most attractive areas.
- Frontier-Based Mapping: Each individual vehicle will maintain its own grids map (namely explored map: *mExploredMap*) that is initially unknown by assigning all the cells their initial (prior) utilities. This map is constructed based on the well-known frontier-based exploration approach of B. Yamauchi[Yamauchi, 1997].
- Map Convolution Between a Group of Vehicles: Each vehicle maintains an observed map (*mObservedMap*) that is produced by the vehicle's latest observations at a specific time step. The size of this map depends on the range of the sensors system of each individual vehicle. The explored map (*mExploredMap*) of a vehicle will be updated by "convoluting" this map with the latest observed maps (*mObservedMap*) from observations of this current vehicle and from other vehicles. By implementing this algorithm, the explored map of each vehicle will be "expanded" and the work-load for each vehicle will also be reduced.

3.5 Proposing Potential Destinations

Once a vehicle in the team has just reached its current destination, it will play the main role in proposing the new global list of potential frontiers that is based on the latest global explored map of the environment. This process is performed by updating all the latest frontiers detected by the vehicles in the team. The list of potential destinations stores information about those frontiers, such as utility and position of each frontier.

3.6 Evaluating Benefits To Go To Each Potential Destination

For selecting a single "best" frontier to visit, each individual vehicle is required to evaluate its own benefit to reach each single frontier in the current list of potential frontiers.

In the case of cooperative exploration, each vehicle in the group will negotiate with other teammates in order to pick its own "best" frontier in a global list of potential frontiers. The benefit to go to each individual frontier by each vehicle can be defined as the subtraction between the expected information gain that the frontier might contribute (for the knowledge of the unknown world) and the cost the vehicle might have to expend in order to reach that frontier[Simmons *et al.*, 2000], [Makarenko *et al.*, 2002], and [Bryson and Sukkarieh, 2005]. For simplicity, the benefit to go to a potential frontier will be estimated throughout the following three aspects:

- Evaluating Frontiers' Utility: The utility of a frontier can be evaluated based on the information it might provide about the unknown area around it. That means, this information can be estimated as a tool to measure how much of the unknown area the vehicle can map when it travels to this frontier. Then, for simplification, it can be assumed that the frontier's utility is approximately proportional to the number of unknown cells adjacent to this frontier.

- Evaluating Cost-To-Frontier: In order to evaluate the cost of traveling to a frontier, the shortest distance (optimal - deterministic motion assumption) path to the frontier from the vehicle's current position is estimated.

- Evaluating Steering Cost: In order to reduce the expense of turning from the vehicle's current position to the new goal point, the cost of this steering will also be considered by the vehicle itself. By bidding this cost in negotiation with other vehicles in the team, the current vehicle might pick its next frontier to go with the lowest expense for steering at this stage. For simplification, this cost will be estimated by the different angle ($|\Delta\theta|$) between the vehicle's current orientation θ and the orientation of the vector connecting the vehicle's current position (the mass of centre of the vehicle) and its new frontier (the centre of the frontier cell). The bigger the $|\Delta\theta|$, the more cost the vehicle has to pay.

- Total Benefit: The *total benefit* b^i_j of selecting frontier j from vehicle i can finally be evaluated as the following equation, in which the *frontier's utility* u^i_j, the *traveling cost* $t\_c^i_j$ and the *steering cost* $s\_c^i_j$ to go to this frontier will be considered and weighted by their relative weights w_u, w_t, and w_s, respectively:

$$\begin{aligned} \text{Total\_Benefit}^i_j &= \sum \text{Utility}^i_j - \sum \text{Cost}^i_j \quad (11) \\ \Rightarrow b^i_j &= w_u \cdot u^i_j - (w_t \cdot t\_c^i_j + w_s \cdot s\_c^i_j) \end{aligned}$$

3.7 Multi-Vehicle Task Assignment for Selecting Destinations

When the $n \times m$ benefit matrix $\{b^i_j\}$ of the team of n vehicles has been produced, in which each element b^i_j of the matrix represents the evaluated total benefit that might be received by vehicle i if this vehicle picks the frontier j in the list of m frontiers as its next destination to visit, then the key problem in the negotiation model among vehicles in the team will be focused on solving the task assignment problem. Solving this task assignment problem will answer the question of which vehicle will be assigned to which destination in order to maximize the benefit of the whole team. And the root x^* of this problem can be illustrated in the following form:

$$x^* = \arg\max_{i,j} \sum_{i,j}^{n,m} (w_u \cdot u^i_j - (w_t \cdot t\_c^i_j + w_s \cdot s\_c^i_j)) \quad (12)$$

The task assignment problem can be represented as the following mathematical representation:

Assume that the team has n vehicles: $V = \{I_1, I_2, \ldots, I_n\}$, and m tasks (destinations) to choose from: $T = \{J_1, J_2, \ldots, J_m\}$. And each vehicle $I_i \in V$ can select any task $J_j \in T$ to do in order to receive this following benefit:

$$b^i_j = u^i_j - c^i_j \quad (13)$$

where: u^i_j is the utility (value) of doing task J_j by vehicle I_i, and c^i_j is the total cost that might be paid by vehicle I_i if it chooses task J_j.

In addition, the task assignment problem has to satisfy two constraints below:

- Constraint C_1: One vehicle I_i can only receive one task to do. That means a vehicle cannot select more than one task to do.
- Constraint C_2: One task J_j can only be assigned to one vehicle. That means there has never been a case where more than one vehicle can do the same task.

Solving the task assignment becomes solving the following global optimization exercise that is mathematically represented in the form of **Linear Integer Programming**[Bertsekas, 1991; Owen, 1995]:

$$\text{Find } \alpha^i_j \text{ to maximize: } f(\alpha) = \sum_{i,j}^{n,m} \alpha^i_j b^i_j \quad (14)$$

subject to:

$$\sum_j \alpha^{\mathbf{i}}_j = 1 \qquad : C_1 \quad (15)$$

$$\sum_i \alpha^i_{\mathbf{j}} = 1 \qquad : C_2 \quad (16)$$

where $m \geq n$, α^i_j are n non-negative integers ($\alpha^i_j \in \{0, 1\}$):

$$\begin{aligned} \alpha^i_j &= 1 \qquad \text{if task } J_j \text{ is } \textbf{Assigned} \text{ to vehicle } I_i \\ \alpha^i_j &= 0 \qquad \text{if task } J_j \text{ is } \textbf{Not} \text{ assigned to vehicle } I_i \end{aligned}$$

3.8 Solving Task Assignment Problem For Multiple Vehicles by AMPL/CPLEX

Mixed integer linear programming can be used for solving the global optimization problem represented in Equations 12 and 14, and one of the useful tool that can be integrated on-board for each individual vehicle is AMPL

(A Modeling Language for Mathematical Programming) developed by Fourer et al.[Fourer *et al.*, 1985].

For on-line simulation in Matlab, each vehicle is designed with an engine for on-line manipulating with the AMPL/CPLEX kernel. The global benefit matrix $\{b_j^i\}$ produced by the team of vehicles will be transferred to AMPL/CPLEX by this engine and the vehicle will send a command to request the AMPL/CPLEX kernel to solve the team's task assignment problem. The result matrix $\{\alpha_j^i\}$ is then received by the whole team, and each vehicle will pick its own destination according to the corresponding element of the result matrix that has a result of 1.

4 Simulation Results

In this research, all the code is developed in Matlab script language, except the implementation for $\mathbf{A}^*$ algorithm which is developed in C++. The object-oriented programming approach is applied for designing and implementing the classes for vehicle objects, environment's map, map's cells, and the AMPL/CPLEX engine. All simulations are conducted in Matlab environment with the interface for constructing unknown environments and adding vehicle objects with known initial positions to the map and with no prior information about the world.

- **The team of 3 UTE cars exploring an unknown environment:** Figures 5a,b,c,d show four snapshots of the simulation for a group of three ground vehicles exploring an unknown environment. The white areas represent the regions that have been explored; the dark gray areas are unexplored; observed obstacle blocks are marked by black colour; and the goals of vehicles are represented by the blue circles. Figures 6a,b illustrate the stage after some simulation time steps where vehicle 3 has just approached its goal and then the team re-solve the new task assignment. Figure 6c,d on the other hand show the results when vehicle 2 has just finished visiting its current destination.

  After **34** seconds exploration, the team has successfully explored the entire environment as can be seen in the snapshots in Figures 7a. The exploration is terminated when the team has covered almost all the environment. The set of Figures 7b,c,d represent the true and estimated errors of all three states of vehicles 1, 2, and 3, respectively.

- **The team of 2 UTE cars and 3 UAVs exploring an unknown environment:** In this simulation, the team is exploring the same unknown environment as was simulated in the previous exploration by the team of 3 ground vehicles. And the team completes exploring the environment in

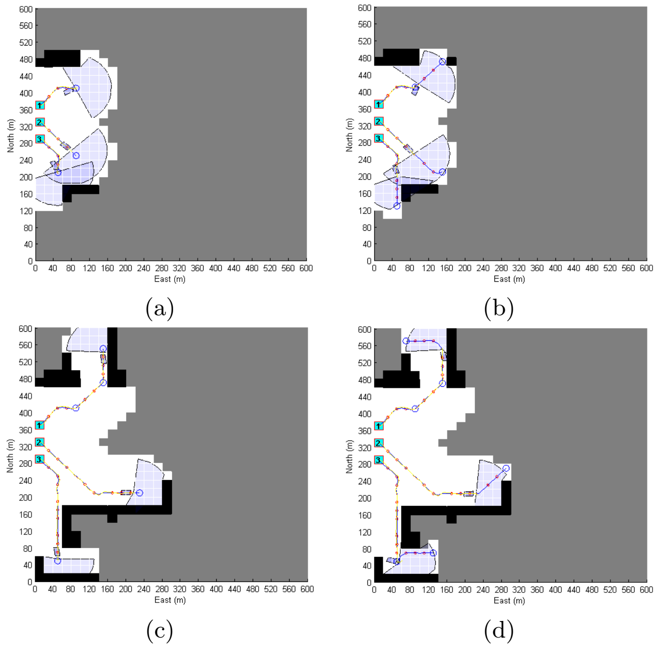

Figure 5: The team of 3 ground vehicles: (a) just before vehicle 1 reaches its current goal; (b) new assigned tasks for all vehicles after vehicle 1 has reached its goal; (c) just before vehicle 3 reaches its current goal; (d) new assigned tasks for all vehicles after vehicle 3 has reached its goal.

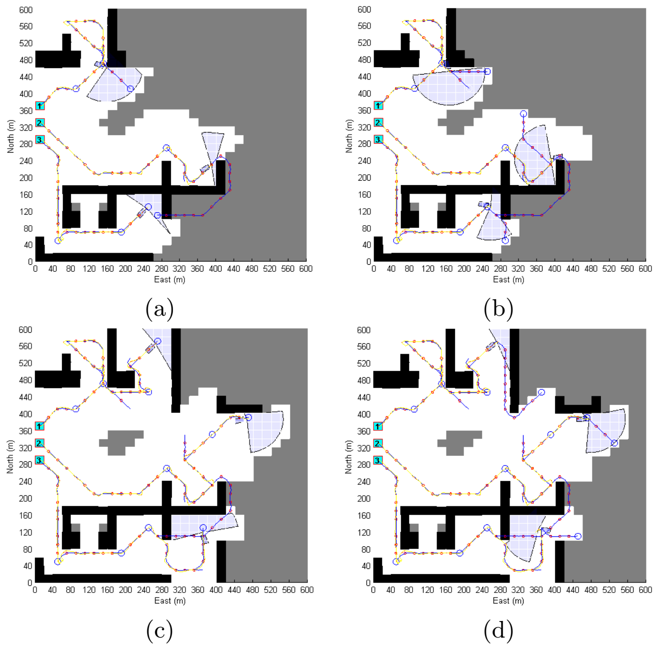

Figure 6: The team of 3 ground vehicles: (a) just before vehicle 3 reaches its current goal; (b) new assigned tasks for all vehicles after vehicle 3 has reached its goal; (c) just before vehicle 2 reaches its current goal; (d) new assigned tasks for all vehicles after vehicle 2 has reached its goal.

23 seconds, that means it is just around 67.5% of the time needed to explore that environment by the team represented in the previous simulation. Figures 8a,b depict the trajectories of all 5 vehicles in

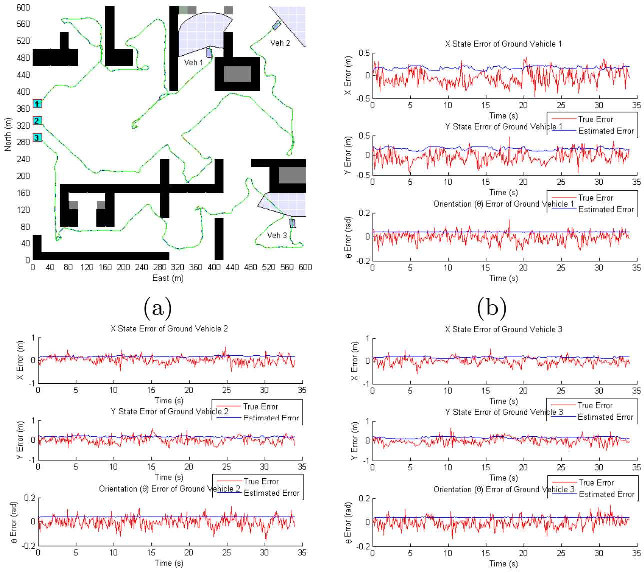

Figure 7: The team of 3 ground vehicles: (a) trajectories of 3 vehicles to complete the exploration; (b) the true and estimated state errors of vehicle 1; (c) the true and estimated state errors of vehicle 2; (d) the true and estimated state errors of vehicle 3.

3-D and 2-D representations, respectively.

- **Comparing some simulations results** Other simulations have also been conducted for cases of either 1 vehicle or 2, 3, 5, and 8 vehicles where each individual vehicles is modeled as either the ground or air vehicle models and integrated the implementation of the EKF algorithm for states estimating. The equivalent simulations in which the greedy algorithm instead of the algorithm proposed in this paper is used for solving the task assignment problem in the team are also conducted. Then the exploration times of all conducted simulations are represented in Figure 9 for comparison. In this figure, the blue rectangles represent the exploration times that were performed by the equivalent numbers of vehicles (shown in the horizontal axis) that used the proposed cooperative framework in this paper. On the other hand, the red circles indicate the exploration times finished by the same equivalent numbers of vehicles but using the greedy algorithm as the team strategy. Table 10 summarizes the exploration times of those simulations.

  From this comparison, it can be said that the proposed cooperative algorithm in this research makes the team exploring quicker than the greedy algorithm strategy does, especially when the number of vehicles and the size of the environment are increased. In the case of having only 1 or 2 vehicles in a team, the two approaches show similar capability. In situation when 8 vehicles explore the map, the exploration times performed by both strategies are quite similar. The difference in the exploration times in this cases could be greater if the simulations of the team of 8 vehicles are conducted in a

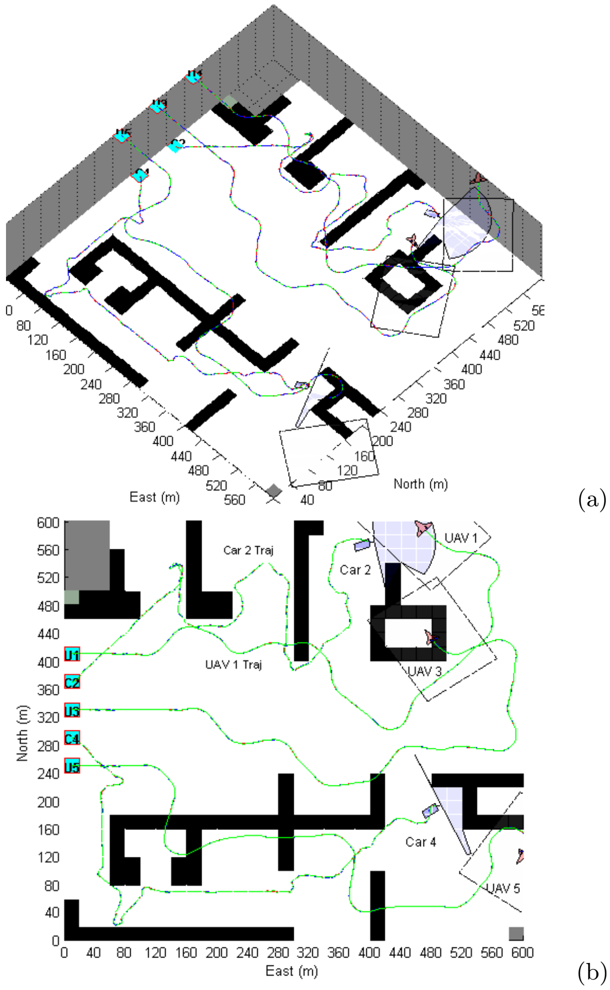

Figure 8: The team of 2 UTE and 3 UAV: (a) trajectories in 3-D; (b) 2-D trajectories.

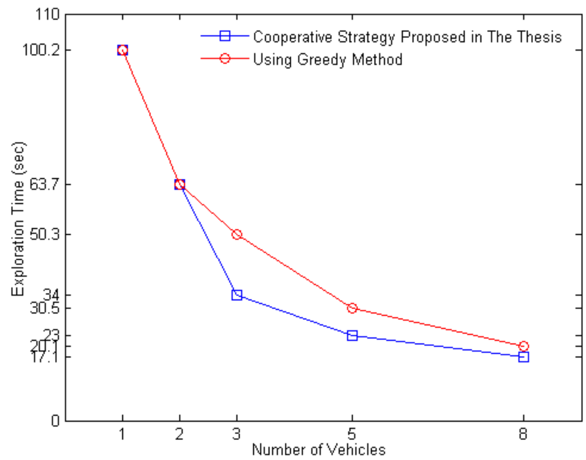

Figure 9: Comparison of exploration times by different team exploring the same unknown environment.

No. of Vehicles	Exploration Time (s)	
	Proposed Strategy	Greedy Method
1 Car	100.2	100.2
1 Car & 1 UAV	63.7	63.7
3 Cars	34	50.3
2 Cars & 3 UAVs	23	30.5
4 Cars & 4 UAVs	17.1	20.1

Figure 10: Table summarizes the exploration times.

larger unknown environment. This is because there might be much more potential destinations to select from at a time during exploration and the task assignment problem might be more complicated and expensive to solve.

5 Conclusion and Future Works

This paper provides a comprehensive frame work for cooperative exploration of multiple autonomous vehicles in unknown environments in the context of a decentralized world. This exploration strategy is designed to improve the quality of gaining information about the world and the overall exploration time of the team as well. The following are the ideas for future directions:

- experimental validation with ground and air vehicles developed at ACFR.
- implementation of SLAM algorithm based on the well-known Extended Kalman Filter[Dissanayake *et al.*, 2001] for ground vehicle localization without any external supports and integrating a map maintained by SLAM and the occupancy grids map in order to extend the quality of the estimation of each grid's utility[Makarenko *et al.*, 2002].
- integrating the real-time airborne SLAM algorithm proposed and tested in [Kim and Sukkarieh, 2003; Kim, 2004] and [Bryson and Sukkarieh, 2006] for autonomous air vehicle localization.

Acknowledgments

This work is supported in part by the ARC Centres of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

[Bertsekas, 1991] Dimitri P. Bertsekas. *Linear network optimization : algorithms and codes.* Cambridge, Mass. : MIT Press, 1991.

[Bourgault *et al.*, 2002] Frédéric Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyt. Information Based Adaptive Robotic Exploration. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:540–545, 30 Sept – 5 Oct 2002.

[Bryson and Sukkarieh, 2005] Mitch T. Bryson and Salah Sukkarieh. An Information-Theoretic Approach to Autonomous Navigation and Guidance of an Uninhabited Aerial Vehicle in Unknown Environments. *IEEE IRS/RSJ International Conference on Intelligent Robots and Systems*, pages 1021–1026, 02–06 August 2005.

[Bryson and Sukkarieh, 2006] Mitch T. Bryson and Salah Sukkarieh. Active Airborne Localisation and Exploration in Unknown Environments using Inertial SLAM. *IEEE Aerospace Conference*, Mar 2006.

[Burgard *et al.*, 2000] Wolfram Burgard, Dieter Fox, Mark Moors, Reid Simmons, and Sebastian Thrun. Collaborative Multi-Robot Exploration. *IEEE International Conference on Robotics and Automation*, (1):476–481, 24–28 April 2000.

[Burgard *et al.*, 2005] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E. Schneider. Coordinated Multi-Robot Exploration. *IEEE Transactions on Robotics*, 21(3):376–386, June 2005.

[Cao *et al.*, 1997] Y. UNY Cao, Alex S. Fukunaga, and Andrew B. Kahng. Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, 4:1–23, 1997.

[Dissanayake *et al.*, 2001] M.W.M. Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transaction on Robotics and Automation*, 17(3):229–241, 2001.

[Durrant-Whyte, 2001] Hugh F. Durrant-Whyte. Introduction to Estimation and the Kalman Filter. Lecture Notes, Australian Centre for Field Robotics, Department of Mechanical and Mechatronic Engineering. The University of Sydney, 2001.

[Elfes, 1989] Alberto Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. *IEEE Computer*, 22(6):46–57, June 1989.

[Fourer *et al.*, 1985] Robert Fourer, David M. Gay, and Brian W. Kernighan. *AMPL : A Modeling Language for Mathematical Programming.* Duxbury Press, 2^{nd}, 2002.

[Gerkey and Matarić, 2003] Brian P. Gerkey and Maja J. Matarić. Multi-Robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures. *IEEE International Conference on Robotics and Automation*, 3:3862–3868, Sep 2003.

[Guivant, 2002] José E. Guivant. Efficient Simultaneous Localization and Mapping in Large Environments. *PhD Thesis,* Australian Centre for Field Robotics, Department of Mechanical and Mechatronic Engineering, The University of Sydney, 2002.

[Hayati *et al.*, 1996] Samad Hayati, Richard Volpe, Paul Backes, J. Balaram and Richard Welch. Microrover Research For Exploration of Mars. *American Institute of Aeronautics and Astronautics (AIAA)*, 1996.

[Kim, 2004] Jong-Hyuk Kim. Autonomous Navigation for Airborne Applications. *PhD Thesis,* Australian Centre for Field Robotics, Department of Mechanical and Mechatronic Engineering, The University of Sydney, 2004.

[Kim and Sukkarieh, 2003] Jong-Hyuk Kim and Salah Sukkarieh. Autonomous Airborne Navigation In Unknown Terrain Environments. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1031–1045, July 2003.

[Ko *et al.*, 2003] Jonathan Ko, Benjamin Stewart, Dieter Fox, Kurt Konoliget, and Benson Limketkai. A Practical, Decision-theoretic Approach to Multi-robot Mapping and Exploration. *Proceedings of the 2003 IEEE/RSl International Conference on Intelligent Robots and Systems*, Oct 2003.

[Makarenko, 2004] Alexei A. Makarenko. A Decentralized Architecture for Active Sensor Networks. *PhD Thesis,* Australian Centre for Field Robotics, Department of Mechanical and Mechatronic Engineering, The University of Sydney, 2004.

[Makarenko *et al.*, 2002] Alexei A. Makarenko, Stefan B. Williams, Frédéric Bourgault, and Hugh F. Durrant-Whyte. An Experiment in Integrated Exploration. *IEEE IRS/RSJ International Conference on Intelligent Robots and Systems*, 1:534–539, 30 Sept–5Oct 2002.

[Makarenko *et al.*, 2003] Alexei A. Makarenko, Eric Nettleton, Ben Grocholsky, Salah Sukkarieh, and Hugh F. Durrant-Whyte. Building a Decentralized Active Sensor Networks. *The 11th International Conference on Advanced Robotics*, pages 332–337, 2003.

[Makarenko *et al.*, 2003] Alexei A. Makarenko, Stefan B. Williams, and Hugh F. Durrant-Whyte. Decentralized Certainty Grid Maps. *IEEE IRS/RSJ International Conference on Intelligent Robots and Systems*, 4:3258–3263, 27–31 October 2003.

[Nash, 1953] John F. Nash. Two-person Cooperative Game. *Econometrica*, 21:128–140, 1953.

[Nash, 1950] John F. Nash. The Bargaining Problem. *Econometrica*, 18(2):155–162, April 1950.

[Neumann, 1963] J. v. Neumann. *Discussion of a maximum problem.* John von Neumann: Collected Works (A. H. Taub, ed.), vol. VI Pergamon Press, Oxford. pages 89–95, 1963.

[Newman *et al.*, 2003] Paul M. Newman, Michael Bosse, John J. Leonard. Autonomous Feature-Based Exploration. *IEEE International Conference on Robotics and Automation*, 1:1234–1240, 2003.

[Nieto, 2005] Juan I. Nieto. Detailed Environment Representation for the SLAM Problem. *PhD Thesis,* Australian Centre for Field Robotics, Department of Mechanical and Mechatronic Engineering, The University of Sydney, 2005.

[Nilsson, 1980] Nils J. Nilsson. *Principles of Artificial Intelligence.* Tioga Publishing Co, USA, 1980.

[Owen, 1995] Guillermo Owen. *Game Theory.* Academic Press, INC., California, 3^{rd}, 1995.

[Patel, 2005] Amit J. Patel. Amit's Thoughts on Path-Finding and A-Star. *URL: http://theory.stanford.edu/ amitp/GameProgramming/*, 2005.

[Rabin, 2004] Steve Rabin. *AI Game Programming Wisdom.* Hingham, MA: Charles River Media, USA, 2004.

[Simmons *et al.*, 2000] Reid Simmons, David Apfelbaum, Wolfram Burgard, Dieter Fox, Mark Moors, Sebastian Thrun, and Håkan Younes. Coordination for Multi-Robot Exploration and Mapping. *American Association for Artificial Intelligence*, 30 July–3 August 2000.

[Thrun *et al.*, 2004] Sebastian Thrun, Scott Thayer, William Whittaker, Christopher Baker, Wolfram Burgard, David Ferguson, Dirk Hähnel, Michael Montemerlo, Aaron Morris, Zachary Omohundro, Charlie Reverte, and Warren Whittaker. Autonomous Exploration and Mapping of Abandoned Mines. *IEEE Robotics & Automation Magazine*, 11(4):79–91, 2004.

[Williams *et al.*, 2001] Brian C. Williams, Phil Kim, Michael Hofbaur, Jonathan How, Jon Kennell, Jason Loy, Robert Ragno, John Stedl, and Aisha Walcott. Model-based Reactive Programming of Cooperative Vehicles for Mars Exploration. *Int. Symp. on Artificial Intelligence, Robotics and Automation in Space*, June 2001.

[Yamauchi, 1997] Brian Yamauchi. A Frontier-Based Approach for Autonomous Exploration. *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, 1997.

[Yamauchi, 1998] Brian Yamauchi. Frontier-Based Exploration Using Multiple Robots. *Proceedings of the Second International Conference on Autonomous Agents*, pages 146–151, May 1998.