We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter

Multiagent Systems for 3D Reconstruction Applications

Metehan Aydın, Erkan Bostancı, Mehmet Serdar Güzel and Nadia Kanwal

Abstract

3D models of scenes are used in many areas ranging from cultural heritage to video games. In order to model a scene, there are several techniques. One of the well-known and well-used techniques is image-based reconstruction. An image-based reconstruction starts with data acquisition step and ends with 3D model of the scene. Data are collected from the scene using various ways. The chapter explains how data acquisition step can be handled using a multiagent system. The explanation is provided by literature reviews and a study whose purpose is reconstructing an area in 3D using a multiagent UAV system.

Keywords: multiagent systems, concepts, architecture, control, communication, 3D reconstruction, image-based reconstruction

1. Introduction

Multiagent systems concern with the coordination of multiple interacting agents in order to complete a single task. Its objective is coordinating relatively simple agents instead of using a complex agent. The coordination of agents gives some abilities to these systems. These abilities are flexibility, robustness and scalability [1, 2].

Flexibility gives the ability of generating an effective solution to different problems. A single agent system can have flexibility but a multiagent system results a more flexible system because using multiple agents can solve and it can be adapted problems which are beyond of capabilities of a single agent. Robustness gives multiagent systems to complete the task despite of failures on the environment and the agents. A single agent system can be robust but it can never be robust as much as a multiagent system because whenever the agent in a single agent system is damaged, the task cannot be completed. However, when an agent in a multiagent system is damaged, the other agents can complete the task. Scalability gives the ability of adding new agents or new groups to the system and working properly with these new elements. This can be provided in a single agent system by changing the elements of the agent but this is not a good solution because this solution has many disadvantages such as cost, re-design and integration. On the other hand, this process can be handled just adding a new agent to the system in a multiagent system.

These abilities give multiagent systems some advantages [3] against a single agent system. These advantages are listed below.

- Multiagent systems are cheaper and consume less energy. Because, its agents are simpler than the agent of a single agent system. Even in the case of having so many agents, it will be cheaper because of complexity of the agent in the single agent system.

- Multiagent systems are faster. Because, the task is shared among agents. This gives the power of parallel processing and results a faster system.

- Multiagent systems are more reliable. If the agent in a single agent system is damaged, the task cannot be completed. But, if any agent in a multiagent system is damaged, the system can still complete the task. This makes multiagent systems more reliable.

However, the coordination of the agents has some challenges. In this chapter, these challenges are addressed and they are explained with a study whose purpose is reconstructing an area in 3D using a multiagent system. The rest of chapter addresses the challenges which are facing while constructing a multiagent system. These challenges are selection of architecture, control techniques and communication mechanism in multiagent systems. The rest of the chapter is structured as follows. Applications areas are addressed in Section 2. An overview to the study is addressed in Section 3. In Section 4, architecture is addressed. Control techniques are explained in Section 5. How multiagent systems communicate is explained in Section 6. The results are presented in Section 7. Finally, the chapter is concluded in Section 8.

2. Application areas

There are many applications of multiagent systems ranging from networking to robotics [4]. But, the advantages and abilities of these systems make them more suitable for some problems [1, 2].

One of the problems is area coverage. This problem can be solved using a single agent system but a multiagent system can be a better solution for this problem. The characteristics of the problem and multiagent systems is the reason for this because when an area is wanted to be covered, it is required that this process should be as fast as and as reliable as possible. These requirements make multiagent systems more suitable for this problem.

The other problem type is dynamic problems. These types of problems are not fixed sized. Their size can change according the time or some other parameters. Using a single agent system cannot fulfill the requirements when the problem size differs. The agent can be either an overhead or not capable enough for the problem. These problems can be easily solved using a multiagent system because it allows us to add or remove agents. In this way, there can be no overheads or capacity issues.

The other problem type is searching. In order to search for an object in an area, a scalable and faster system is needed. A single agent system can provide these properties but when it is constructed once, it will be hard to reconstruct it again. This yields a system which is not fully scalable and fast. However, these types of problems can easily be solved by a multiagent system. When a larger scene must be searched, adding more agents to the system can solve scalability problem and the system will be faster because there are more than one agent in the system, this approach allows searching faster in a given time interval.

3. An overview of the study

There are different 3D reconstruction techniques. An image based 3D reconstruction technique is used in the study. Image based 3D reconstruction generally has a reconstruction pipeline in which there are several steps. The pipeline generally starts with data acquisition. In data acquisition, the study uses a multiagent UAV system. The reason of using a multiagent system is that their advantages and strengths are well suitable for the requirement of image based 3D reconstruction.

In order to reconstruct a scene in 3D with an image-based 3D reconstruction, the objects in the scene must be photographed in all the viewpoints. **Figure 1** shows how the scene must be photographed.

The multiagent system is started with a command which gives an order to one of the agents in the system. The command is given by giving the positions of the area. This command can be given to any agent. The control mechanism of the system is designed to provide that any agent can get a mission and can share it among other agents. This control mechanism is designed for the study. There are some known and used control mechanisms but we want to benefit from the advantages of two known and used control mechanisms. These mechanisms are distributed control and centralized control. The designed control mechanism is a hybrid version of these two mechanisms. It takes the advantages of master-slave side of centralized control and the advantages of distributed mission sharing of distributed control and results a system in which any agent can be a master and any agent can give order to other agents. The control mechanism works with communication mechanism and it can even be said that the control mechanism works above communication mechanism because the most suitable communication mechanism is selected for the designed control mechanism. The selected communication mechanism is direct communication. All the agents can send and get messages with it. The messages determined when the system is designed and there can be different types of agents in the system, in other words, the architecture of the system provides heterogeneous agents. The communication mechanism is selected for this property, too. Any agent which implements it can easily join the system and be part of the mission.

When an area is selected using any of the agent, the agent shares the mission among the other agents equally. Then, the agents start the rotate around the area. They rotate as long as their mission, for example, if 60° are given for an agent, it rotates only 60°. While they are rotating, they capture the video of the scene. The master agents check if all the agents complete their mission. When all of them complete it, they all go to a location which is determined in advance. Then, the videos transferred to a computer and 3D reconstruction pipeline is started.

The 3D reconstruction pipeline includes tree main steps which is similar but its purpose and the steps not same with the work [5]. These steps are summarization of the videos as frames, extracting camera positions and creating point cloud,
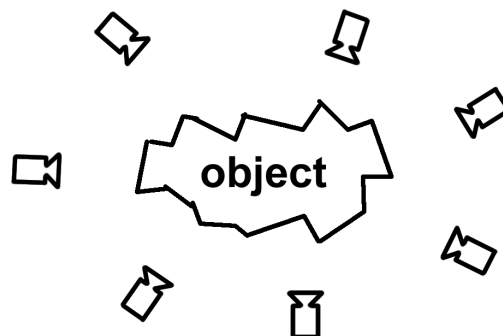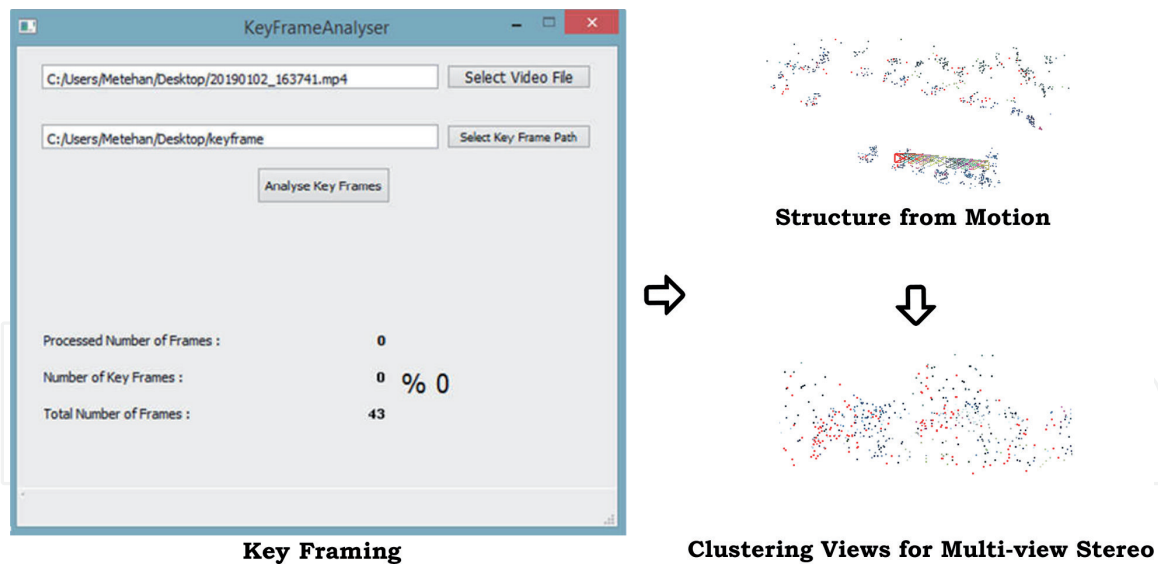

Figure 1.
Photographing an object.

Figure 2.
Overview of the pipeline.

dense reconstruction. The summarization of the video is handled using a technique known as key framing. They are summarized because there are lots of frames in the video which sees the scene with similar view angle. These similar frames do not contribute the 3D reconstruction process. Key framing provides the selection of the frames which are identical and fulfill the view angle. The algorithm [6, 7] used in the key framing process is explained below.

- Select the first frame as a key frame.

- For each of the frames after the key frame, compute the similarity between the key frame and the frame.

- If the similarity is above a similarity threshold, eliminate the frame.

- If the similarity is under a threshold, store it and select as the last key frame.

In the next step, the camera positions are extracted and point cloud of the scene is created. This step is handled using a technique known as structure from motion. The study uses VisualSfM [8] as the structure from motion application. The application takes key frames as input then produces the camera positions and the point cloud of the scene. The most important reason to select this application is that this application uses SIFT feature descriptor. SIFT [9] feature descriptor gives the best precision results [10], which are very important for the study.

The final step of the pipeline is dense reconstruction. The study uses an MVS technique known as Clustering Views for Multi-view Stereo (CMVS) [11]. CMVS tool can be integrated to the VisualSfM application and they can be used in the same interface. The output of the VisualSfM becomes the input of the CMVS. The output of the CMVS, in other words the output of the pipeline because CMVS is the last step of the pipeline, is 3D dense of the scene. **Figure 2** shows an overview of the pipeline.

4. Architecture

Architecture refers to how the agents of the system are constructed and how they differ from each other. There are two main architectures [12]. These are

homogeneous and heterogeneous architectures. Architecture selection should be problem dependent. Both of them have some advantages and disadvantages against each other. According to problem and facilities, one of them can be selected.

4.1 Homogeneous architecture

In homogeneous architecture, all the agents have same capabilities and all of them are identical. A system which has agents from same brand can be an example of homogeneous architecture. The example can be understood well from **Figure 3**. The advantage of this architecture is simplicity. Implementing a homogeneous architecture is simpler than heterogeneous architecture.

4.2 Heterogeneous architecture

In heterogeneous architecture, the agents can have different capabilities and they can be different from each other. There can also be different agent groups to form a heterogeneous architecture. A system which is implemented with a hetero-geneous architecture should not constrain the number of agents in these groups. All the agent groups should have desired number of agents and their agent number can also be different from each other. A system which has agents from some differ-ent brands can be an example of heterogeneous architecture. **Figure 4** illustrates a heterogeneous architecture.

Heterogeneous architectures can be more flexible and cheaper than homoge-neous architectures using some optimizations. Tasks can be allocated according to features of an agent. For example, while the agent which has a better camera is capturing photos, the other one which has a better processing power can handle task sharing issues. Although a multiagent system is not constructed heterogeneous agents, it can be better to design an architecture suitable for heterogeneous agents.

4.3 Architecture of the study

The study is implemented with a fixed brand agent but the agents from differ-ent brands or having different capabilities can be used. This feature is provided by the connection mechanism. The system uses a local area network (LAN) in order to communicate. The communication is handled via messages. There are some types of communication messages in our system. These messages are addressed in
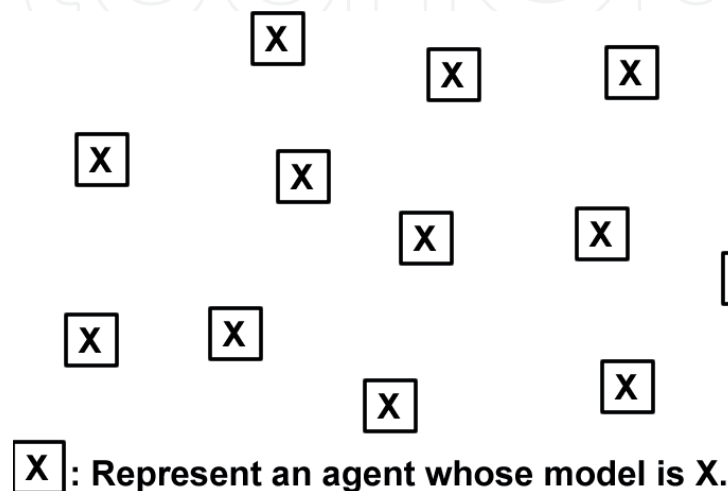

X : Represent an agent whose model is X.

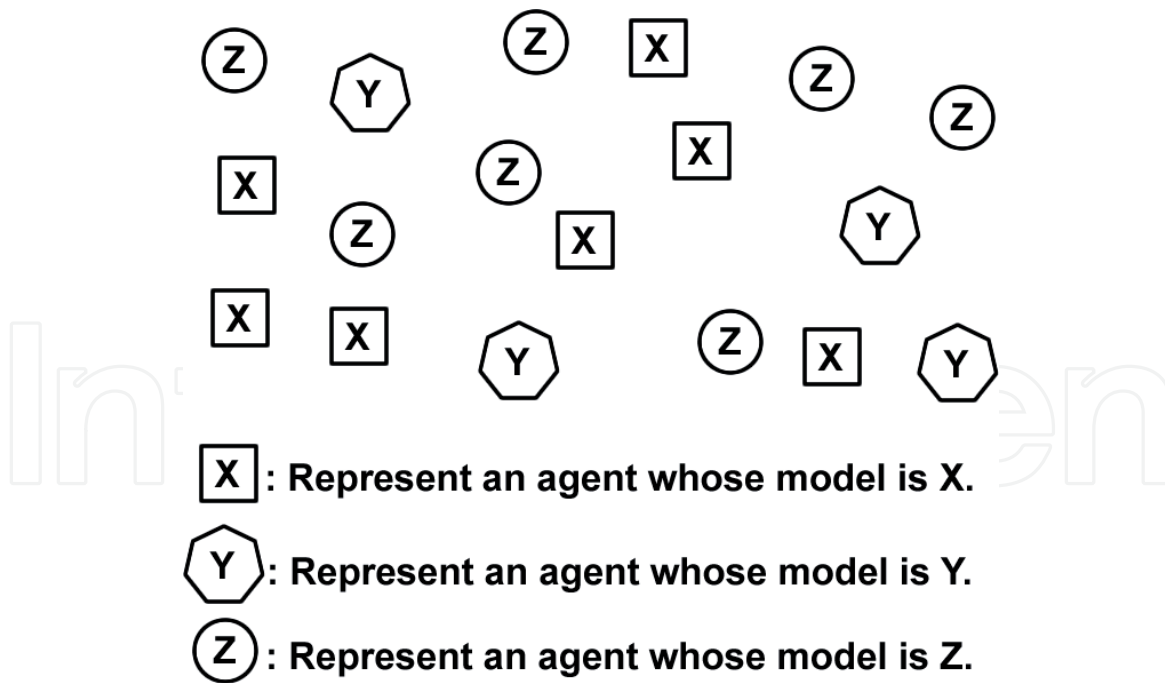**Figure 3.**
A homogeneous multiagent system.

Figure 4.
A heterogeneous multiagent system.

Section 6. One of the reasons to select this type of communication structure is providing a heterogeneous architecture. When an agent implements these messages, it can easily join our system. For example, there is a message type whose purpose is transferring command messages. This message carries information about the coverage area and the task. An agent which has features suitable for our problem and which has implementation to parse the information can take command messages and be a part of our system.

The architecture does not restrict the number of different heterogeneous agents or number of agent. There can be any number of agents as much as an agent can store the address of other agents. This property is provided by implementation of the architecture. In our implementation, all the agents have a list to store the address of other agents. This list does not keep agent dependent information but it only store the address of other agents. In this way, the agents need just the address of the other agents. This improves scalability of the system and gives the power of working with desired number of different agents. This implementation also does not restrict the number of agents from different groups. There can be any number of groups and any number of agents in these groups. These properties make our system more robust, flexible and scalable.

4.4 Implementation of another architecture

When the study uses a homogeneous architecture, a different model agent could not be added and this would constrain the abilities of the system. For example, assume that we want to add an agent and we have an agent from different brand. If the system does not provide heterogeneous agents, we cannot use the agent. Instead, we have to buy a new agent whose brand same as the ones in the system. This increases our costs and this is not a desired situation. On the other hand, some products can be sold in a specific time intervals. Even if we do not buy a constructed agent, its internal components are sold in same way too. This situation results a system which only can be used as long as the agents can be constructed or bought.

When our agent list would have been designed to some fixed different agents, it results a system which is heterogeneous but whenever a different agent must be

added, the implementation of the system must be updated according to the agent, which is wanted to add. This situation would be laborious and could not be applied in some situations. For example, if the implementation is not proper to add a new type of agent or an agent is not proper for the system, it cannot be added. This situation has some drawbacks which are same as in the previous case.

5. Control mechanisms

Control mechanism is one of the challenging and vital components. They affect the whole system from agents to communication styles. A control mechanism determines how the agents are controlled; in other word, it determines how the agents decide for coordination.

Choosing a right control mechanism depends on some parameters. These are performance of agents, communication style of the system, environment, cost and mission type. There can be some parameters depending on the problem. This situation leads to finding different types of control mechanism or variating the existing ones in order to build effective systems. In this chapter, the main three control mechanisms [13] are addressed. These are centralized, distributed and hierarchical control.

5.1 Centralized control

In centralized control, there is a fixed agent to control the coordination of the system. This agent can be called master or mediator. The other agents are called slaves. There is a centralized control illustration in **Figure 5**. All slaves take orders from master. This approach creates a system in which the master agent is constructed a bit different or completely different than slaves. Depending on the problem or the facilities, differences can be set. But, these differences determine the ability of the master and the system. For example, if the master is used only in order to control of the system, it does not need to have a payload. On the other hand, if there are limited numbers of agents and the master must contribute to completion of the task, then it must have a payload.

Centralized control has some advantages and disadvantages. It requires a simple consensus mechanism in which the agents just obey the master. But on the other hand, implementation of centralized control is somehow difficult because it requires design of agents and implementation of the system in two different perspectives. These are how the master and slaves are constructed and how the system is implemented. These can result a master whose cost is higher than slaves and a system which is implemented in a master perspective and in slaves perspective. For example, there must two different programs, one of them is for master and
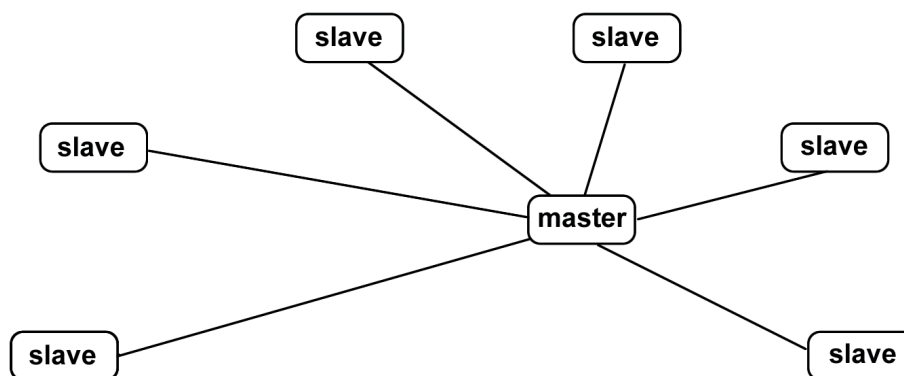

Figure 5.
A centralized control.

the other one is for slaves. The other important disadvantage is that, operational capability of the system depends on the master. If the master is damaged, the system cannot be used or the task cannot be completed.

5.2 Distributed control

In distributed control, all agents are equal for the control. There is not any master agent to control the slaves. Instead, all the agents contribute control equally. This is done by consensus. In multiagent systems, consensus means that the agents decide on an agreement which determines the behavior of the system. There are various consensus algorithms in literature [14, 15]. All of them have some differences and come with advantages and disadvantages against each other. But all of them have the same objective and all of them have the characteristics of distributed control.

Using distributed control mechanism can be advantageous. It results a system in which design and implementation are easy because all the agents are in same level, there is not any need to think about in master and slave perspectives. The other important advantage is that if any of the agents is damaged, the task can still be completed. On the other hand, implementation of a consensus algorithm is a challenging task. They are problem dependent and they affect the communication styles of the system (**Figure 6**).

5.3 Hierarchical control

Hierarchical control acts like a tree and it has a tree characteristics. All agents can have children and can have parent. All agents are responsible for control of their children. There is a root agent at the top of the control. Actually, the root is responsible for the control of the system. It shares the task to its children and its children can share the task among their children. **Figure 7** can give an idea about the mechanism. Depending on the situation, this mechanism can lead a system in which there are one type, two types or three types of different agents. In the case of three different agents, there can be a root, controllers, and leaves. The root can be responsible for sharing task among controllers. The controller can be responsible for the control of the leaves and only the leaves can contribute to completion of the task. In the case of two different types of agents, there can be a root, controllers and leaves but instead of only controlling the leaves, the controllers contribute to completion of the task. In the case of a one type of agent, there can be a root, controllers and leaves but all the agents can contribute to completion of task.

Hierarchical control mechanism generally used in problems which requires multiple tasks and which spreads on a wide area. There are some advantages and disadvantages
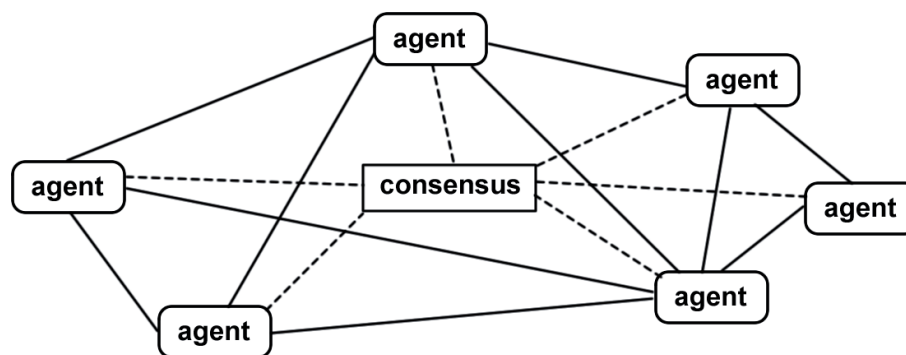

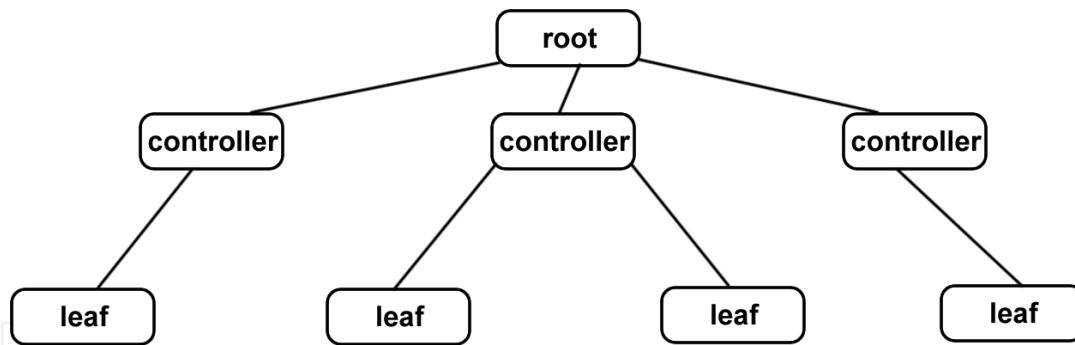**Figure 6.**
A distributed control.

Figure 7.
A hierarchical control.

using hierarchical control. One of the advantages is that it has a simple consensus mechanism in which children obey the order of parents. The other important advantage is that the system is well suitable for multiple tasks which must be done simultaneously. On the other hand, implementation of the system can be difficult in both agent perspective and the system perspective. It is difficult in agent perspective because the system can have different types of agent. This allows constructing different types of agent whose requirements and capabilities are different from each other. It is difficult in system perspective because different types of agents require different implementation.

5.4 Designed control mechanism for the study

A different control mechanism is designed for the study which is mixture of centralized and distributed control. In the mechanism, there is not any fixed master. Instead, all agents can be a master as long as they have mission. When any of the agents have a mission, it shares the mission among others including itself and it became a master until the mission is completed. It is similar to distributed control because there is not any fixed master in the system. It is similar to centralized control because the tasks are shared among agents by a master. This hybrid control mechanism results a system which is very suitable for our problem. **Figure 8** shows the control mechanism of the system.

There are some reasons to design such a control mechanism for the study. The most important reason is that this mechanism offers some advantages against the other control mechanisms. One of the advantages is that it provides a simple consensus algorithm because whenever an agent has a mission, it just gives the mission to others and they just accept the order and then try to complete it. This algorithm is as simple as the consensus algorithm of a centralized control. The other advantage is that it provides easy design and implementations. It provides an easy design because all agents can be constructed same, there is not any need to design different types of agents. It provides an easy implementation because all agents act the same, and there is not need to implement different types of agents. The other important advantage is operational capability. When an agent is damaged, the others can still complete the mission or they can still take a mission because the master is not fixed. Thus, the system does not depend on a single agent.

5.5 A different view

If centralized control is selected for the study, there must be a fixed agent in order to control the slaves. This situation enforces the study designing two different implementations for each of the agent type. On the other hand, the study became depended to the master. If the master gets some damages, the study cannot be used.
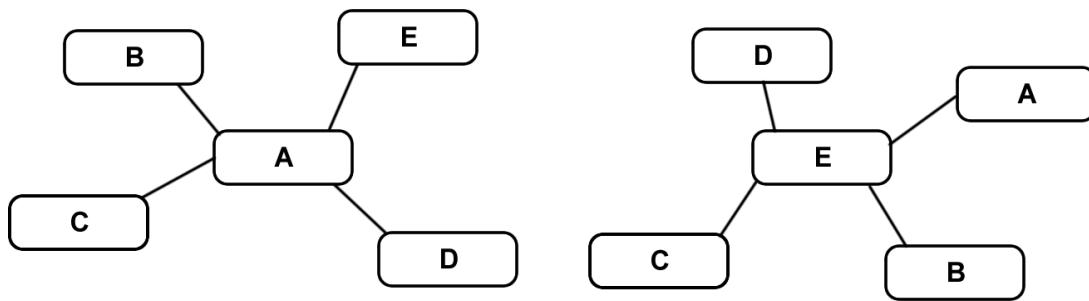
Figure 8.
Control mechanism of the system.

If distributed control is selected, a consensus algorithm is needed for coordination. But, our problem is more suitable for a consensus algorithm like the one in centralized control because our area, which is reconstructed in 3D, is selected in advance. This makes our problem static and can easily be solved with the consensus mechanism of centralized control.

If hierarchical control is selected, there must be a hierarchy among agents but our problem is not suitable well for a hierarchy. A hierarchy may be needed in situations in which the mission has several tasks or it has a large task. But, our problem requires only one task and the areas are not very large.

6. Communication mechanisms

Communication is another challenging and vital component in multiagent systems. Unlike other vital components, it does not affect the coordination directly. Instead, it provides a basis and coordination is provided using it. A communication mechanism determines how the agents communicate.

Communication is effected from components of multiagent systems and some other factors. These factors and components are environment, agents, control mechanism, architecture and weather. These have a critical role in order to select a mechanism. There are three [13, 16] main communication mechanisms. These are direct communication, communication using sense and communication using environment. Each of them has advantages and disadvantages. These are discussed in this section. Depending on the problem and facilities, one of them can be selected.

6.1 Communication using environment

In this mechanism, the agents do not interact with each other. Instead, they use environment as a medium. Thus, they adapt to the changes in the environment. In other word, they make their next movement according to environment. This provides communication among them. This mechanism generally is known as stigmergy [17, 18]. It can be found in nature. Some insects and termites complete their tasks using this type of communication. It is inspired from nature and applied to multiagent systems.

In multiagent systems, this mechanism is provided by implementation of agents and input devices like sensors, cameras or others. Implementation of the agents determines what the agent does when it encounters with an object. Input devices provides observing and identification of the objects which are used as communication medium. All agents have the input devices. While they are observing the environment, they collect information and this information is parsed.

When the information is parsed, the agents act how they are implemented according to information. For example, there can be an obstacle type whose purpose is turning the agents to left. When the agents are implemented suitable for the purpose of the obstacle type and when the agent encounters with the obstacle, they turn to left.

There are some advantages and disadvantages of the mechanism. One of the advantages is that ones the mechanism is set, it can provide a communication mechanism for a lot of agents and number of agents does not affect the system. The other important advantage is that it can be used in complex and large tasks without considering connection range. On the other hand, there are some important disadvantages of the mechanism. One of them is that the cost of using the mechanism can be very high because it can require multiple sensors and multiple sensors require more performance. This approach comes with high cost. Another important disadvantage is that designing such an environment is generally difficult. There are a lot of things to consider. These are objects in the environment, different objects which do not affect the communication, input types of the agents. When these things are determined, the system can be set.

6.2 Communication using sense

In this communication mechanism, the agents communicate with local interactions. One of the agents senses another. In this way, they make inferences about their next movement or their next states and they contribute completing the task according to these inferences.

The mechanism is realized by sensors and implementation of the agents. The sensors sense the other agents, in other words, they gather information about the states of others. Implementation of the agents determines what the agent does when it senses an agent which is desired type or which carries desired features. According to problem, the agents can be implemented in order to sense any agents or any desired features. While they are sensing the other agents, they determine them according to their implementations and when they sense the appropriate ones, they behave according to their implementations. For example, the agents can be implemented for sensing the ones whose distances are 50 cm away from them and when they sense them, they can try to keep the distance constant.

The most important advantage of the mechanism is that it can used to solve some kind of problems. These problems are pattern formation and flocking. In order to solve these, the agents sense their neighbors and they adjust themselves according to them. In contrast, the mechanism is more suitable for distributed control and using a centralized control is much harder.

6.3 Direct communication

In this communication mechanism, the agents communicate with direct messages. Every agent can get and send messages to others. The messages are determined when the system is designed and all the agents use these pre-determined messages.

This mechanism is generally set on a network. Thus, all the messages are sent and get through it. The mechanism is realized using implementation of agents and a component which let agents join the network. The implementation handles what the agents do when a message is received or what kind of message will be send to other agents. The component handles the communication issues. It provides getting messages which are released to the network and sending messages to other agents,

in other word, it provides an interface in order to release messages. According to implementation and the need of the system, the agents can send or get messages from any agents. While agents wait messages from others, they can do whatever they want but when they get a message from other agents, they must behave according to their implementation. For example, an agent can be programmed to sleep when any mission is not assigned to the agent but when it gets a message which carries a mission, it must start completing it.

Direct communication mechanism can be used to implement any type of control mechanism. It can be even used for a control mechanism which is derived from other control mechanisms or whose design is unique and it can be used for any type of problem. In addition, there can be any number of agents in the system but in case of sending or getting too many messages simultaneously, the system can be slower. Or it cannot be used in case of a mission which spreads on a large area. Because, a network which provides connection for long distance is required and this requires a high cost which cannot be afforded.

6.4 Communication used in the study

Direct communication mechanism used in the study. The UAVs used in the study has a feature to connect a network and the system is suitable for any kind of networks. Thus, any network can be used for communication.

The system has tree type of communication messages. These are command messages, communication messages and mission completed messages. All of them have different purpose and all of them contribute to system in different perspectives. In order to send and get these messages, the agents uses their IP address as a communication address. Thus, the agents must know IP address of other agents to send and get messages from them. All the agents have an agent list which stores other agents IP addresses. Agents can get and send these messages according to the agent list and they must behave according to messages. The following subsection explains how these message types are used in the study.

6.4.1 Connection messages

Connection messages set up a connection between two agents. In other words, it allows two different agents to recognize each other. The agent lists are filled using connection messages. These messages carry the IP address of the agent. There are two connection messages in the system. These messages are add agent and confirm agent. They are in following form and each agent can send and get these.

- ADDAGENT#ip_address

- CONFIRMAGENT#ip_addres

An add agent message is sent when an agent wants to add another agent to its list. Its purpose is sending an add request to other agents. In order to add an agent, the agent list of the sender must first be checked to ensure that the agent has not been added yet. If it is so, sender agent send an add agent message to the receiver agent.

Confirm agent message is sent whenever an add message is received. Its purpose is confirming the sender about the connection. A receiver agent does not need to check if the sender is on the list or not. If it takes an add message, the sender is not on the list because when a connection message is completed, both the sender and the receiver add the other agent to its list. This approach yields a system in which

the agent is on same level. When an agent gets confirm agent message, it adds the sender of the message to the list. The algorithm is explained step by step below. **Figure 9** shows the sender and receiver side briefly.

1. The sender checks if the receiver has not been added yet.

2. If the receiver has not been added, sends an add message.

3. If the receiver gets an add message, adds the sender of the message to the agent list

4. The receiver sends a confirm message to the sender.

5. When the sender gets the confirm message, it adds the receiver to its agent list.

6.4.2 Command messages

When any of the agents takes a mission, it shares the mission with its agent list. Before the agent shares the mission, the mission information must be defined. The agent which shares the mission is responsible for defining mission information. The mission is shared in a command message.

In the implementation of the study, a mission information message is defined by a class. The class carries the location of the area, the location of the mission which is assigned to the agent, the number of agents in the system, the ID of the agent. The location area is the location which the master agent takes, in other word, it is the whole scene. The location of the mission is the task which is given an agent to cover. This information is required to defining the area. The number of agents in the system is required because the agent must know how many agents are there in the system. Whenever a mission is shared, all the agents are given an ID by the master agent. The ID of the master agent becomes 1 and the ID of the other agents
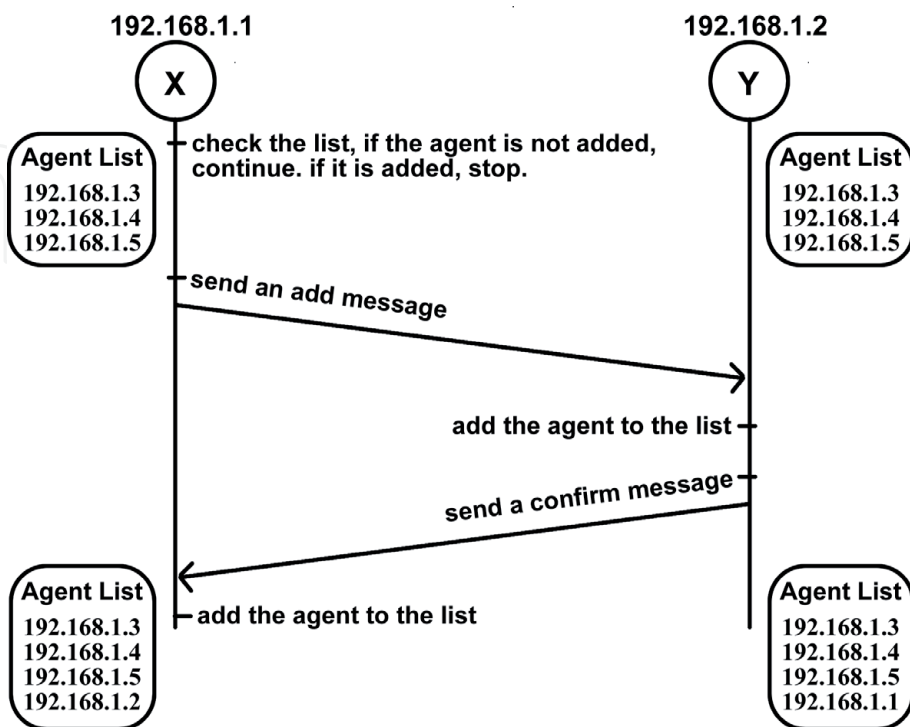

Figure 9.
Connection message.

determined relatively with the master agent. This information is required because it is used in coverage of the area and for checking if the agents complete the mission successfully.

6.4.3 Mission completed messages

The system must be sure that the mission is completed successfully. In order to be sure of the completion of the mission, mission completed messages are sent. There are four types of mission completed message in the system, these are error, state, success and not completed messages. They carry an IP address and they are in the following form.

- ERROR#ip_address

- STATE#ip_address

- SUCCESS#ip_address

- NOTCOMPLETED#ip_address

Actually, there are three types of scenarios to realize mission completed messages. These scenarios arise because there are differences among mission completion of master as first, mission completion of slaves as first and in the case of not taking a response from a slave.

When any of the slaves completes its mission before the master, they send a success message to the master. The purpose of the success message is informing the master about the completion. When the master agent completes the mission before the slaves, it sends a state message to the slaves which have not sent a success message yet. The purpose of state message is determining if the slave completes its mission or not and if it does not complete, determining the reason. The reason can be an error. When the slaves get the state message, they send not completed message if they have not completed their mission but they think they can complete it successfully, in other word, if they have power and facilities to complete the mission. The purpose of not completed message is informing the master about the state of the slave. When the master agent gets a not completed message from a slave, it sends states messages until the agent sends a success message or an error message. The error message is sent in the case of there is an error in the agent or the agent cannot complete the mission because of facilities or power. The purpose of error message is informing the master about the state of the slave. When the master gets an error message, it completes the mission of the sender of the message. When the master sent a state message and a slave does not response in a given time interval, it is counted as error too. The following algorithms explain scenarios and **Figure 10** and **Figure 11** shows the scenarios briefly.

When a slave finishes first:

1. The slave sends a success message to master.

2. Master gets the message and will not send it a state message.

When a master finishes first:

1. The master sends a state message to slaves.

2. The slaves get the message.

3. The slaves send a not completed message if it can complete the mission.

4. The slaves send an error message if it cannot complete the mission.

5. If the master gets a not completed message, it sends states messages until it gets a success or error message.

6. If the master gets an error message, it completes the mission of the sender agent.

When a slave does not response:

1. The master sends a state message and waits for a while.

2. If the slave does not response, the master completes its mission.

6.5 Implementing other communication mechanisms

If communication using environment is selected, there must be some objects whose purpose is defining the coverage area for each of the agents in the scene.
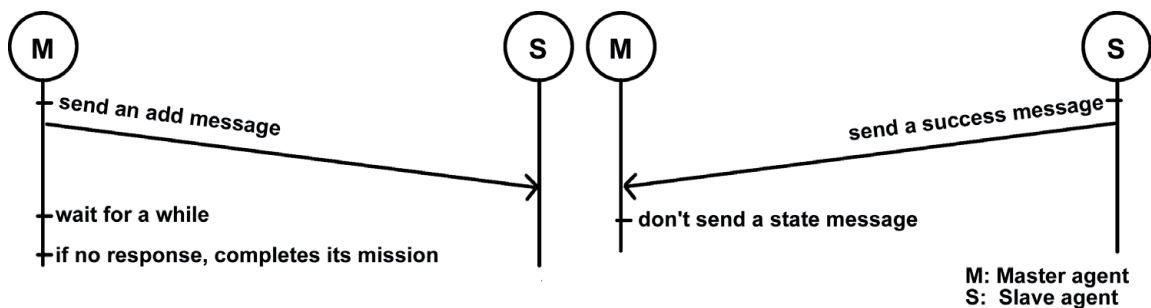
Figure 10.
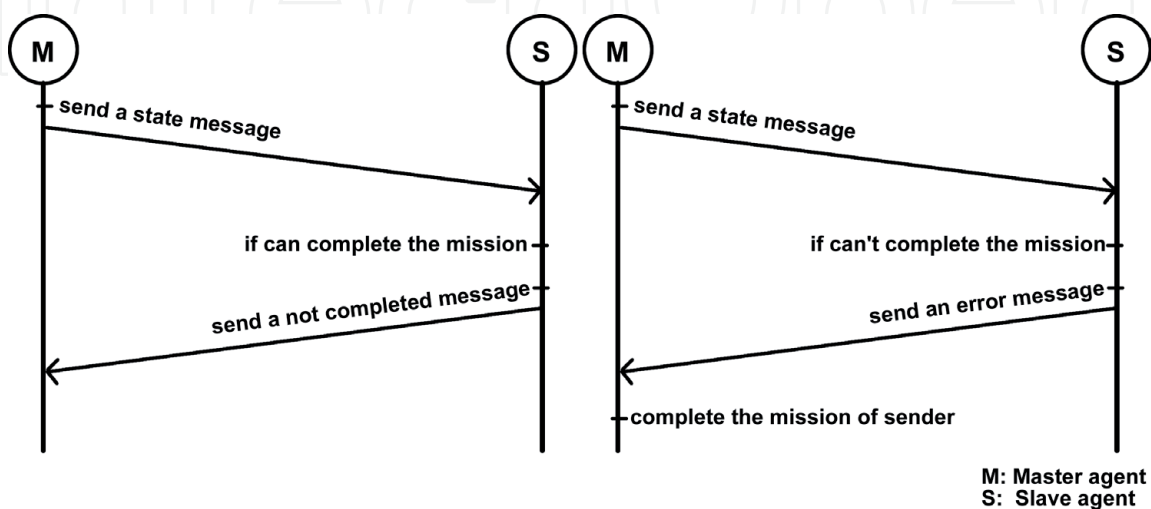*A slave finishes first and a slave does not response.*

Figure 11.
Master finishes first.

There must also be additional sensors or a camera implementation whose purpose is determining the objects in order to communicate. In the case of having sensors on the agent, the agents collect the information which comes from the sensors and tries to determine the coverage area from the collected information. The sensors add additional cost to the agent and the system. They add additional cost to the agent because buying and integrated a sensor to an agent requires additional time and money. In addition, the performance of the agents must be upgraded because the sensors also need process power. They add additional cost to the system because the system requires objects and buying objects requires money. In addition, positioning the objects to the scene requires a human interaction or designing a different study whose purpose is just positioning the object to the scene. In the case of using a camera implementation, while the cameras record the scene, they try to determine the coverage area. This case adds additional complexity to the system because a camera implementing requires detecting objects in the scene and integrating this implementation to the main implementation results a more complex system.

If communication using sense is selected, there must be additional sensors on the agent and there must be different implementation of the system. The additional sensors are required because the agent uses information which comes from the sensors in order to communicate. The types of the sensors can vary according to desired communication range. For example, one sensor type can sense 100 m distances but another can sense 1000 m. This situation restricts coverage area with respect to sensors and results a system whose capabilities is strictly depended on the sensors. In addition, the sensors add the system additional cost to the agent. The reason of the cost is same as with the cost of communication using environment. A different implementation is required because in this case the agents must observe other agents and determines their own coverage regions with respect to other agents.

7. Results

The study is executed with one, two and four agents in four areas. The sizes of the fields are 1600, 2500, 3600, and 4900 m^2, respectively. When they start their mission, a chronometer is starter and it lasts until they finish their missions and the results are recorded. The results are shown in **Table 1**. The results show that when the number of agents is more, the system is faster.

The relationship between the agents and the size of the agents can be seen in **Figure 12** and **Figure 13**.

Figure 14 depicts the 3D point cloud obtained by the multiagent reconstruction approach. The point cloud can easily be converted in to a mesh with the available texture information acquired from the agent cameras.

	1600 m^2	2500 m^2	3600 m^2	4900 m^2
1 Agent	69 sec	83 sec	96 sec	104 sec
2 Agents	39 sec	46 sec	51 sec	55 sec
4 Agents	24 sec	28 sec	31 sec	34 sec

Table 1.
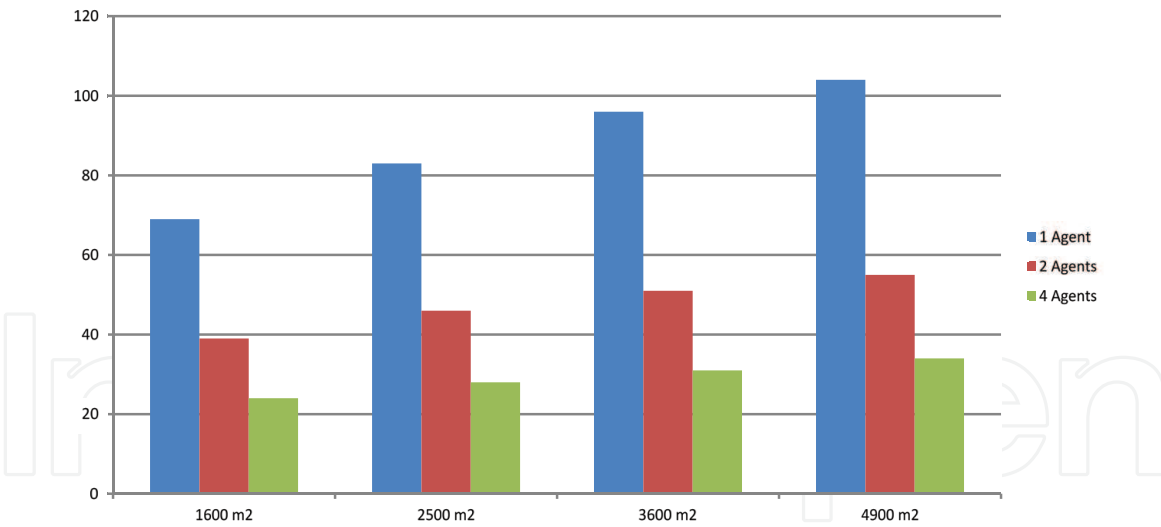*Results for 1, 2 and 4 agents for various scenarios.*

Figure 12.
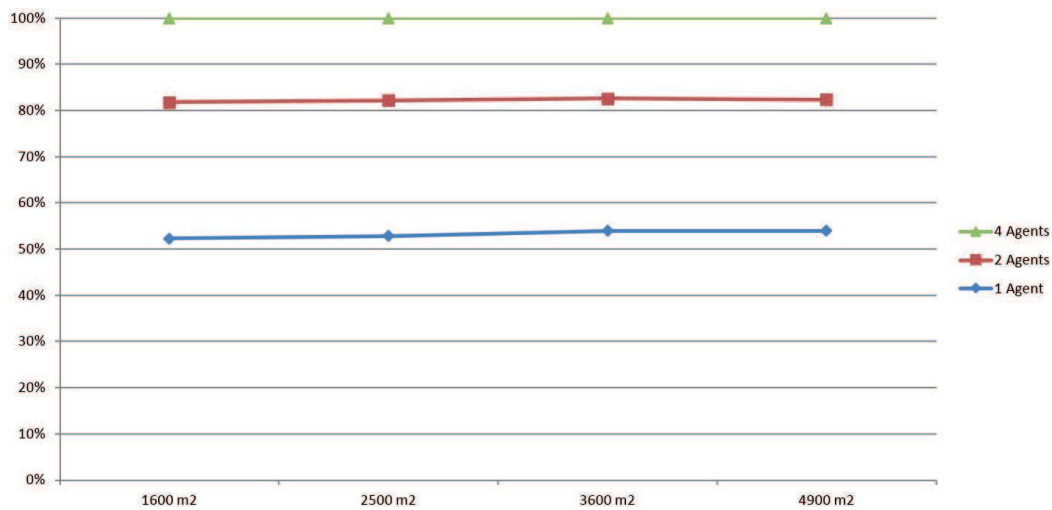*Relationship between the number of agents and the size of the fields.*

Figure 13.
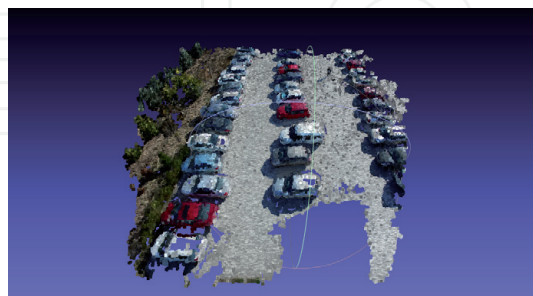*Percentage relationship between the number of agents and the size of the fields.*

Figure 14.
An output from the study.

8. Conclusion

Multiagent systems deal with coordination of more than one agent in order to complete a task. There are several features of these systems. These features are robustness, scalability and flexibility. These features give some advantages to multi

agent system. These advantages make multiagent systems cheaper, faster and more reliable. There are many applications of multiagent systems ranges from networking to robotics.

Although there are many applications and advantages of multiagent systems, designing a multiagent system is a challenging task. There are several issues to consider. The chapter addresses most encountered issues. These are architecture selection, communication mechanism selection and control mechanism selection. They must be implemented in most of systems. The chapter addresses each of them with a study. Each of them is explained in sections and the study is given as an example in each sections. The experimental results of this study shows that a multiagent system is more effective and more advantageous compared to single-agent systems.

Author details

Metehan Aydın[1], Erkan Bostancı[1]*, Mehmet Serdar Güzel[1] and Nadia Kanwal[2]

1 Department of Computer Engineering, Ankara University, Ankara, Turkey

2 Lahore College for Women University, Lahore, Pakistan

*Address all correspondence to: ebostanci@ankara.edu.tr

IntechOpen

References

[1] Şahin E. Swarm robotics: From sources of inspiration to domains of application. In: Şahin E, Spears W, editors. Swarm Robotics Workshop: State-of-the-Art Survey. Lecture Notes in Computer Science, No. 3342. Berlin, Germany; 2005. pp. 10-20

[2] Navarro I, Matía F. An introduction to swarm robotics. ISRN Robotics. 2013;**2013**:10. Article ID 608164

[3] Balaji PG, Srinivasan D. An introduction to multi-agent systems. In Innovations in multi-agent systems and applications-1. Berlin, Heidelberg: Springer; 2010. pp. 1-27

[4] Ali D, Salil K, Raja J. Multi-agent systems: A survey. IEEE Access. 2018;**6**:1. DOI: 10.1109/ ACCESS.2018.2831228

[5] Mutlu B, Hacıömeroğlu M, Serdar GM, Dikmen M, Sever H. Silhouette extraction from street view images. International Journal of Advanced Robotic Systems. 2014;**11**(7):114

[6] Bostanci E. 3D reconstruction of crime scenes and design considerations for an interactive investigation tool. International Journal of Information Security Science. 2015;**4**(2):50-58

[7] Bostanci E. Crime scene investigation in 3D using video frames. In: International Symposium on Digital Forensic Science; Ankara, Turkey. 2015. pp. 8-13

[8] Wu C. VisualSFM: A Visual Structure from Motion System. 2011. Available from: http://ccwu.me/vsfm/

[9] Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 2004;**60**(2):91-110

[10] Güzel MS. Performance evaluation for feature extractors on street view images. The Imaging Science Journal. 2016;**64**(1):26-33

[11] Furukawa Y et al. Towards internet-scale multi-view stereo. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE. 2010

[12] Stone P, Veloso M. Multiagent systems: A survey from a machine learning perspective. Autonomous Robots. 2000;**8**(3):345-383

[13] Cao YU, Fukunaga AS, Kahng A. Cooperative mobile robotics: Antecedents and directions. Autonomous Robots. 1997;**4**(1):7-27

[14] Yu W, Chen G, Cao M, Kurths J. Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics. IEEE Transactions on Systems, Man, and Cybernetics, Part B. 2010;**40**(3):881-891

[15] Gao L, Liao X, Li H, Chen G. Event-triggered control for multiagent systems with general directed topology and time delays. Asian Journal of Control. 2016;**18**(3):945-953

[16] Gregory D et al. A taxonomy for multi-agent robotics. Autonomous Robots. 1996;**3**(4):375-397

[17] Heylighen F. Stigmergy as a universal coordination mechanism I: Definition and components. Cognitive Systems Research. 2016;**38**:4-13

[18] Dorigo M, Bonabeau E, Theraulaz G. Ant algorithms and stigmergy. Future Generation Computer Systems. 2000;**16**(8):851-871