CHAPTER V

System and Software Design

5.1 Conceptual Design

From the objective of this research described in chapter one, it can be classified into the following software requirement.

a) The designed software can perform camera calibration.
b) It can capture image from all camera synchronously.
c) It can extract 2-D image position from captured image.
d) It can find 3-D position and velocity of moving object.
e) It can increase speed and efficiency of the system by using multiple computers.

In order to handle these requirements, there are several processing modules had been designed including camera calibration module, image capturing module, 2-D image processing module, 3-D pose estimation module and PC to PC communication module. The arrangement of these processing modules is shown in figure 5.1.

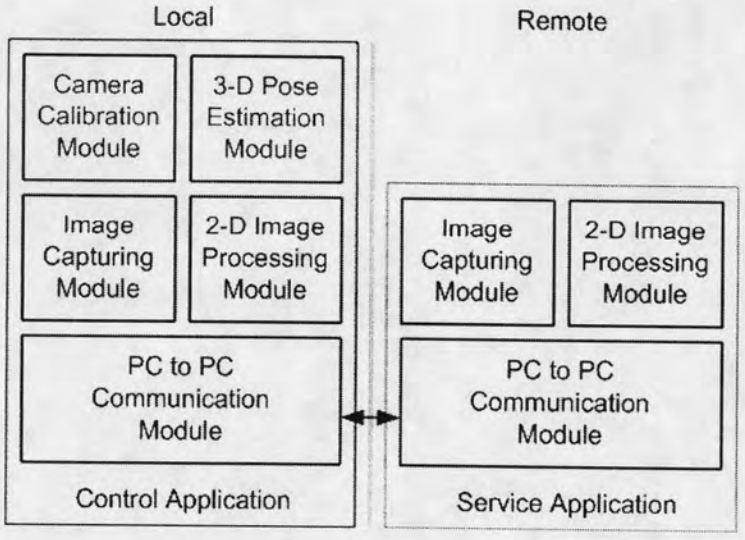

Figure 5.1: The arrangement of the processing modules.

There are two computers have been used in this research, one is local computer or main computer and another one is remote computer. Local computer control overall processes both contained in itself and remote computer. A type of application had been developed for each computer. Control application operates on local computer while service application operates on remote computer.

One major issue of this developed software is to reduce computational time in real-time tracking process, and then the size of data which transferred between PCs should be minimized. In order to minimize transferring data, the developed system had been designed to send only 2-D image position from remote computer to local computer.

5.2 Software Application

5.2.1 Service Application

Service application is software application which runs in the background of Microsoft Windows operating system and operates on remote computer. It uses less resource than the normal software application and has no user interface. It consists of three processing modules including image capturing module, 2-D image processing module and PC to PC communication module

In camera calibration process, service application has been used to capture images of calibration pattern from cameras which attached to remote computer. Then, it sends the captured images to local computer to perform camera calibration. In real-time tracking process, it used to capture images of object with image capturing module. The captured images are sent to 2-D image processing module to find 2-D image positions and then the service application sends 2-D image positions to the local computer to perform 3-D pose estimation.

5.2.2 Control Application

Control application is the main software application which developed in this research. It controls all service applications to work together. It performs image capturing and 2-D image processing as in service application. It also performs camera calibration and 3-D pose estimation in real-time tracking process. Moreover, it provide graphic user interface to make the system controlled by user. The processing modules in control application are:

a) Image capturing module
b) 2-D image processing module
c) PC to PC communication module
d) Camera calibration module
e) 3-D pose estimation module

5.3 Processing Modules

5.3.1 PC to PC Communication Module

PC to PC communication module used to manage communication between PCs in PC-cluster. This module use dataBLIZZARD PCI to PCI data mover card for this purpose. It wraps data into programming data structure called message before transferring and then the message will be pushed in to a message queue to waiting for transferring as show in figure 5.2.

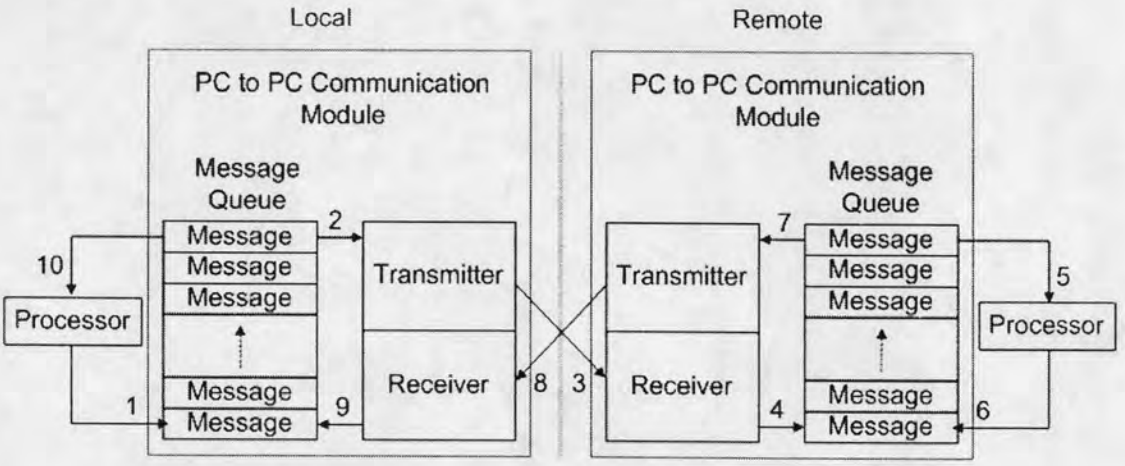

Figure 5.2: PC to PC communication module schematic.

Message is a package of data. It consists of two parts including header and body. Header of the message is the identifier of the message. It contains properties of the message such as sender, receiver, id, direction, data size and category. The body contains data which need to be transferred to the receiver. The size of body must correspond to data size defined in the header part. Message can be classified into three categories including command message, returned message and event message.

Command message is send by transmitter to controls receiver. After receiver processes received command message and some returned data occur,

the receiver will generates returned message and sends it back to the transmitter. Event message is the message used to notify other computers in cluster about the event occur in the transmitter.

According to the message's direction, it can be classified into two types including outgoing (sending) and incoming (received). A message with outgoing direction is created by transmitter and pushed into a message queue of transmitter to waiting for processing. A message with incoming direction is created by receiver to wraps the received data, and pushed into a message queue of receiver to waiting for processing.

The message queue is a queue that used to store transferring messages. It processes each message using First In, First Out (FIFO) method. The message queue will check the message direction before processing. If the direction of the message is incoming direction, then the message will be send to local processor for processing. But if the direction of the message is outgoing, then the message will be send to another computer by using PT as a key of transferring.

In transmitter, after message with outgoing direction was popped from message queue, the all data contained in both header and body of the message will be written into local card's memory and protect this memory from any modification. Then, the local card will send the PT interrupt to the remote card with vector ID, which is an unsigned integer created from bitwise operation between message ID and message category ID.

In receiver, after remote card receives PT signal from transmitter, it will check vector ID and reads data from local card's memory with corresponding data size. Then, receiver creates a new message with the same header and body as transmitted message but different in message direction. The new created message, which has incoming direction, will be pushed into message queue of the receiver to waiting for processing.

5.3.2 Image Capturing Module

Image capturing module has been designed to control capturing image from attached cameras. There are two camera trigger type available including hardware and software trigger type. Software trigger type is to capture by software command but hardware trigger type is to capture by external trigger signal. Hardware trigger type has been used to make synchronous image capturing form multiple cameras (see section 4.1.3).

5.3.3 Camera Calibration Module

Camera calibration process needs to be performed before real-time 3-D pose estimation process. The time constraint is not necessary to be considered to design this module. The various types of calibration method should be used for the best result such as Tsai's calibration method and Zhang's calibration method. Calibration methods and calibration patterns had been designed to be COM (Component Object Model) objects. Each calibration method object will implement IMctCalibration interface and calibration pattern object will implement IMctCalibPattern interface.

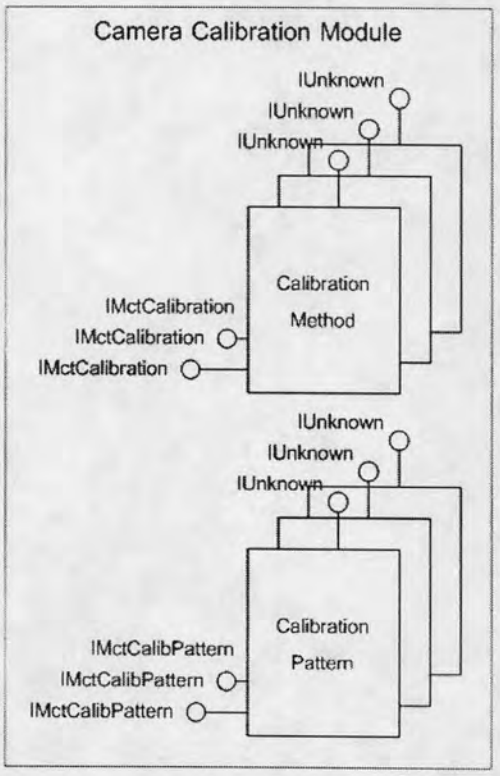

Figure 5.3: COM objects in camera calibration module.

IMctCalibration contains properties and functions of calibration method. These functions use 2-D image points and their corresponding 3-D world points in order to calibrate the camera.

IMctCalibPattern contains properties and functions of calibration pattern. The COM object which implements this interface provides 3-D world points and searches for with their corresponding 2-D image points.

To calibrate the camera, camera calibration module, which contained in control application, performs operation as show in figure 5.4.

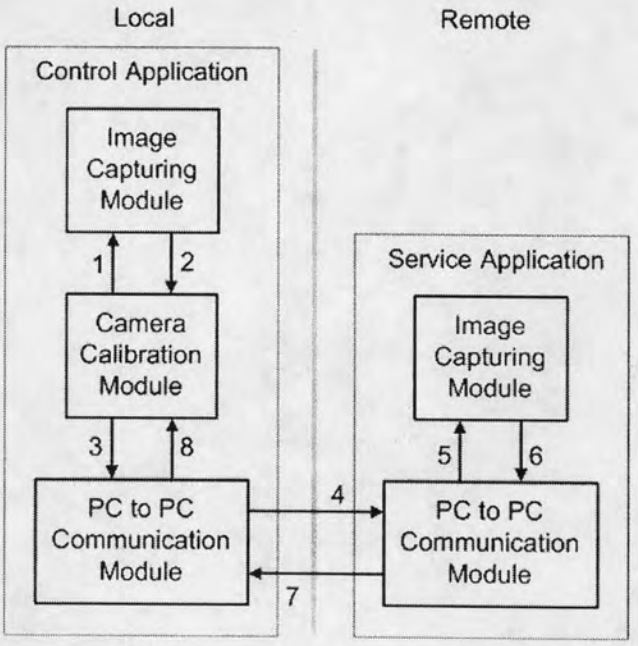

Figure 5.4: Camera calibration process.

1) Camera calibration module sends software trigger signal to local image capturing module.
2) Local image capturing module captures requested images and then send them back to camera calibration process.
3) Camera calibration module sends software trigger signal to local PC to PC communication module.
4) Local PC to PC communication module transform software trigger signal into transferring message and transfers it to remote PC to PC communication module.
5) Remote PC to PC communication module receives transferring message and sends it to remote image capturing module.
6) Remote image capturing module captures requested images and then send them to remote PC to PC communication module.
7) Remote PC to PC communication module transform captured images into transferring message and transfers it to local PC to PC communication module.

8) Local PC to PC communication module receives transferring message and transform it back to captured images. Then captured images are sent to camera calibration module.

Local capturing (1-2) and remote capturing (3-8) perform until the camera calibration module received the number of images corresponding to the requirement of selected camera calibration method then it performs camera calibration to get camera models.

5.3.4 2-D Image Processing Module

2-D image processing module has been used to find target's image position in captured image. The method used to find image position of object called recognition method. It also designed to be COM object which implement IMctRecognition as show in figure 5.5.

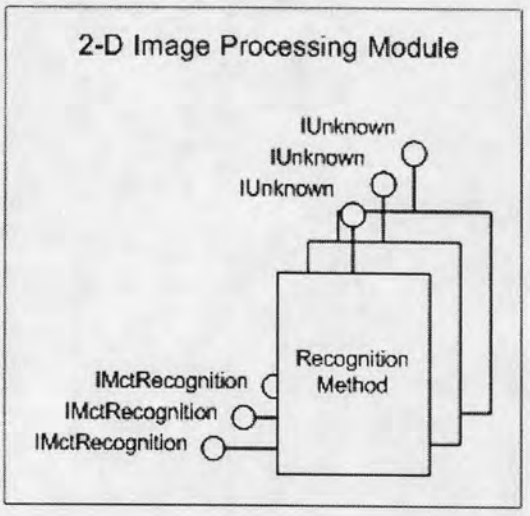

Figure 5.5: COM objects in 2-D image processing module.

2-D Image processing module has been called in real-time tracking process. The calculation time used in this process depends on image size. In order to minimize calculation time of this process, the region of interest has to be estimated before perform recognition. Region of interest can be found by assume projected 2-D images of object moves on image plane with constant acceleration.

5.3.5 3-D Pose Estimation Module

3-D pose estimation module consists of a various type of 3-D reconstruction method. These methods have been designed as COM object that

implements IMctReconstruction interface as show in figure 5.6. They will be called in real-time process, and then calculation time should be considered as a major problem.

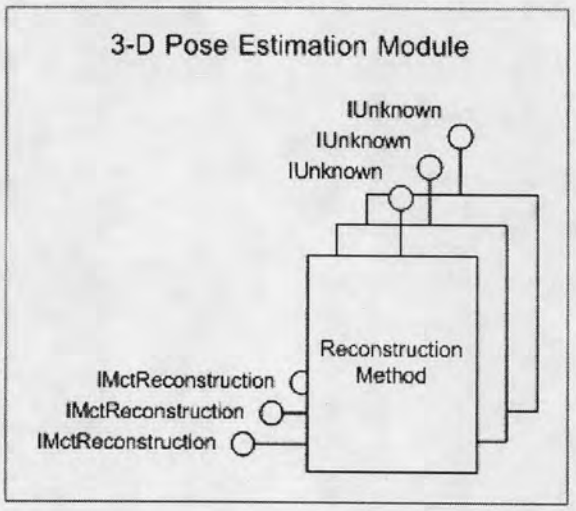

Figure 5.6: COM objects in 3-D pose estimation module.

Before perform 3-D pose estimation, all cameras have to be calibrated and image capturing module need to be set for synchronous image capturing (see section 4.1.3). To perform 3-D pose estimation, 3-D pose estimation module in control application performs operations as show in figure 5.7.

1) 3-D pose estimation module sends software type trigger signal to local image capturing module.
2) Local image capturing module will send software trigger to only primary camera.
3) Primary camera send hardware trigger signal to other cameras via GP2.
4) Both primary and secondary cameras capture images and send them back to image capturing module which attaches to them.
5) Image capturing module passes the captured images to 2-D image processing module to find 2-D image position of object.
6) In local computer, 2-D image processing module sends 2-D image positions to 3-D pose estimation module.
7) In remote computer, 2-D image processing module sends 2-D image positions to PC to PC communication module.

8) Remote PC to PC communication module transforms 2-D image positions into transferring message and transfers it to local PC to PC communication module.

9) Local PC to PC communication module receives transferring message and transform it back to 2-D image positions. Then 2-D image positions are sent to 3-D pose estimation module.

After 3-D pose estimation module received all required images, it performs 3-D reconstruction using a selected reconstruction method component.

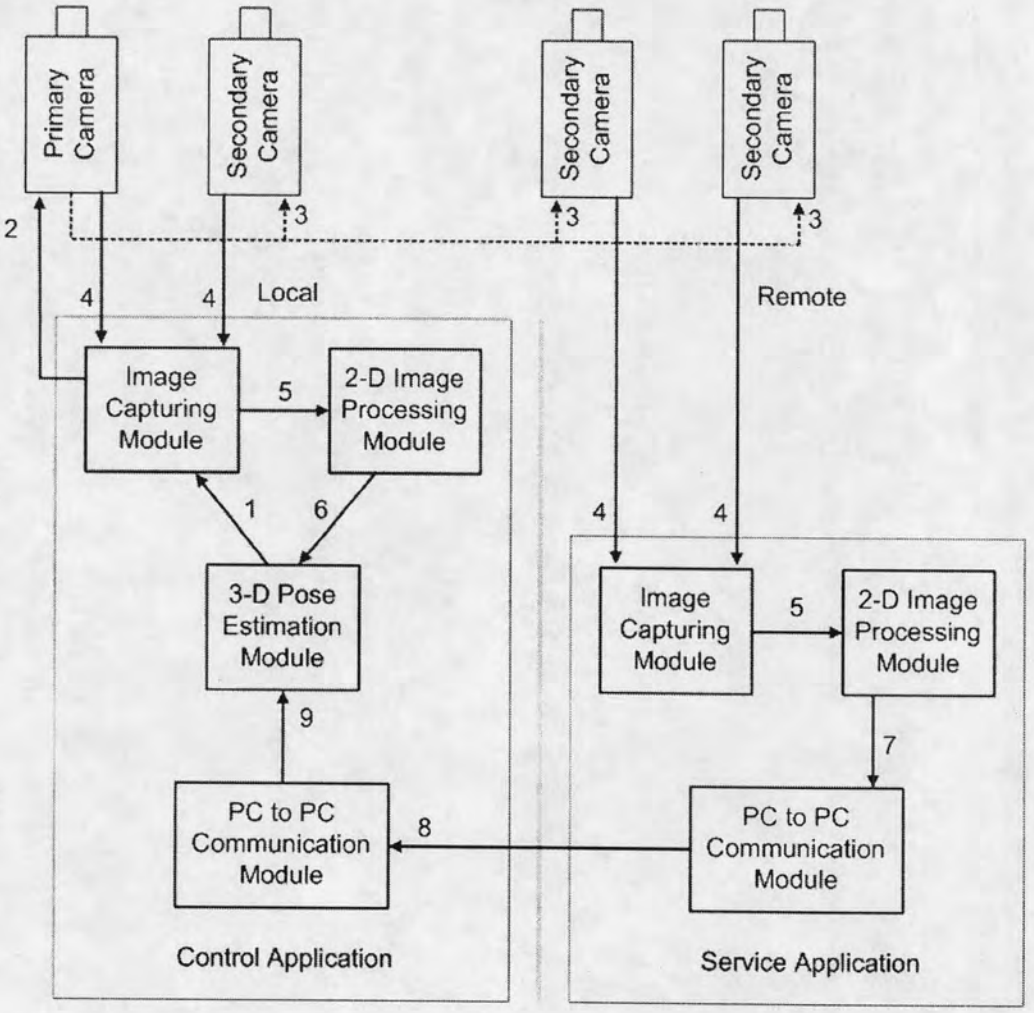

Figure 5.7: 3-D pose estimation process.