Real time video based moving object tracking is one of the exigent missions in the field of computer vision (Kinjal and Darshak 2012). Videos are sequences of two dimensional images/frames, displayed in a fast frequency so that the human eyes can percept the continuity of its content. Object tracking detects, recognize and track objects from image sequence and describe object behavior. The moving objects are often subject to background clutter, occlusion, dynamic illumination change, appearance change which adds complexity in tracking the objects. This research focuses on tracking human and vehicles using various methods in a robust and accurate manner. This chapter briefs about object detection, object tracking and collision avoidance in general and how it is done in proposed methods.

1.1 INTRODUCTION TO TRACKING

Object tracking is defined as the process of segmenting an object of interest from a video picture and keeping track of its motion, orientation and occlusion in order to obtain vital information. The main objective of video tracking is to associate target objects in consecutive video frames. Tracking is usually performed in the context of higher-level applications that require the location and/or shape of the object in every frame. Tracking generally accords with non-stationary object, target descriptions and the background which changes over time (Behrooz Sadeghi and BehzadMoshiri 2007). Most of the available algorithms (Andrew Senior et al 2006, Aniruddha Kembhavi et al 2011,
Tracking is the construction of correspondence of relationships between “tracked objects” in previous frames and “detected objects” in the current frame (Yilmaz 2006). In general, the processing framework of visual surveillance in dynamic scenes includes the following stages: (1) Object detection, (2) Tracking, (3) Understanding and description of behaviors and (4) Identification (Weiming Hu 2004).

Difficulties in tracking of objects can arise due to abrupt object motion, change in appearance patterns of the object and the scene, non-rigid object structures, object-to-object and object-to-scene occlusions and camera motion. The application of object tracking is pertinent in the tasks of

- Motion-based recognition, human identification based on gait, automatic object detection
- Automated surveillance, monitoring a scene to detect suspicious activities or unlikely events
- Video indexing, automatic annotation and retrieval of the videos in multimedia databases
- Human-computer interaction, gesture recognition, eye gaze tracking for data input to computers
- Traffic monitoring, the real-time gathering of traffic statistics to direct traffic flow
- Vehicle navigation, video-based path planning and obstacle avoidance capabilities

The endless list of applications perform object tracking with high end computation attaining a great level of accuracy. Important assort of new applications will enable this radical development includes location-based services. Location management, that is, the management of transient location information, is an enabling technology for all these applications. As a matter of fact, the location-based services and needs are expanding rapidly. Thus the proliferation of high-powered computers, the availability of high quality and inexpensive video cameras and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms.

The major goal of this thesis is to identify and track the moving objects in such a way that tracking is done in an appropriate, fast and intelligent manner. The proposed work attempts to provide a tracking mechanism for both single and multiple objects which has a dynamic background. Three major techniques namely Signal Processing based techniques perform single object tracking, Database based technique and Artificial Intelligence (AI) based techniques perform multiple object tracking are proposed. In the Signal Processing technique two approaches are proposed. The first approach is Kalman Filters and the second approach is Particle Grouping. In the Database based technique two approaches are proposed. The first approach is Rk-d tree based indexing and the second approach is agent based. The AI based technique two approaches are proposed. The first approach deals with Quantum Computing and the second approach pacts with Multi-Layer Perceptron.
1.2 OBJECT DETECTION

Object detection is identifying an object with a set of known labels from the image or a video sequence and it is the basic step for further analysis of video. This not only creates a focus of attention for higher level processing but also decreases computation time considerably. Commonly used techniques for object detection are background subtraction, statistical models, temporal differencing and optical flow. The functions of object detection are to verify the presence of objects in image sequences and locate it for recognition. The objects can be of different shape, size or color. The orientation of these objects may also vary from time to time. The objects may vary somewhat in different viewpoints, in different sizes/scales or when they are translated or rotated.

1.2.1 Approaches for Object Detection

A common approach for object detection is to use information in a single frame.

1.2.1.1 Point Detectors

Point detectors are used to find interest points in images which should have some properties like it should be distinct that is clear against background, should be independent of geometrical and radiometric distortions, should be robust to noise, should have global uniqueness, should have meaning an expressive texture in their respective localities. Interest points have been long used in the context of motion, stereo and tracking problems. A desirable quality of an interest point is its invariance to changes in illumination and camera view point.
To find interest points, the most common technique uses computation of variation of the image intensities in a [n,n] matrix patch in the horizontal, vertical, diagonal and anti-diagonal directions and select the minimum of the four variations as representative values for the window. A point is declared interesting if the intensity variation is a local maximum of the patch.

1.2.1.2 Background Subtraction

One of the simplest techniques for detection is background subtraction. It refers to the process of segmenting moving regions from image sequences. This process involves building a model of the background and extracting foreground objects from an image. A foreground object can be described as an object of attention which helps in reducing the amount of data to be processed as well as provide important information to the task under consideration. Any significant change in an image region from the background model signifies a moving object. The pixels constituting the regions undergoing change are marked for further processing. Usually, a connected component algorithm is applied to obtain connected regions corresponding to the objects.

There are many challenges in developing a good background subtraction algorithm. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background objects and shadows cast by moving objects. A good background model should also react quickly to changes in background and adapt itself to accommodate changes occurring in the background such as moving of a stationary chair from one place to another. It should also have a good foreground detection rate and the processing time for background subtraction should be real-time.
In practice, background subtraction provides incomplete object regions in many instances or there may be holes inside the object since there are no guarantees that the object features will be different from the background features. The most important limitation of background subtraction is the requirement of stationary cameras. The camera motion usually distorts the background models.

1.2.1.3 Segmentation

Image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. Usually image segmentation is an initial and vital step in a series of processes aimed at overall image understanding.

1.2.1.4 System Learning

Object detection can be performed by learning different object views automatically from a set of examples by means of a supervised learning or inductive learning mechanism. Learning of different object views waives the requirement of storing a complete set of templates. Given a set of learning examples, supervised learning methods generate a function that maps inputs to desired outputs. The learning examples are composed of pairs of object features and an associated object class where both of these quantities are manually defined.

Selection of features plays an important role in performance of learning. Once the features are selected, different appearances of an object can be learned by choosing a supervised learning approach. These learning
approaches include adaptive boosting (Artur 2007), decision trees and support vector machines.

1.2.2 Feature Selection for Object Detection

Mostly features are chosen manually by the user, depending on the application domain. However, the problem of automatic feature selection has received significant attention in the pattern recognition community.

Automatic feature selection methods are divided into filter methods and wrapper methods.

- The filter method tries to select the features based on a general criterion, for example, the features should be uncorrelated.

- The wrapper method selects the features based on the usefulness of the features in a specific problem domain.

A wrapper method of selecting the discriminatory features for tracking a particular class of objects is the Adaboost algorithm (Artur 2007). Adaboost is a method for finding a strong classifier based on a combination of moderately inaccurate weak classifiers. Given a large set of features, one classifier can be trained for each feature.

Adaboost will discover a weighted combination of classifiers (representing features) that maximize the classification performance of the algorithm. The higher the weight of the feature, the more discriminatory it is.

1.2.3 Problems in Object Detection

The detection of objects becomes difficult when there are,
• Changes in lighting or color
• Changes in viewing direction
• Changes in size / shape

The problem in object detection becomes more tedious when the objects are partly or fully obliterated from human vision, in situations where background keeps changing and when the orientation of the object keeps changing from time to time. Some of the important issues in object tracking that are to be considered such as viewpoint related ambiguities, occlusions and coincidences. Only if the object is detected properly, the subsequent steps in object tracking can be done correctly or with less error.

The proposed method uses the artificial intelligence technique that automates the object detection by pairing Haar training (Lienhart and Maydt 2002) which is a weak learner supported with a strong classifier namely discrete adaptive boost classifier.

1.3 OBJECT TRACKING SYSTEM

Tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. It is usually performed in the context of higher-level applications that require the location and/or shape of the object in every frame. The different types of tracking are given in Figure 1.1 (Yilmaz 2006).
Figure 1.1 Different Types of Object Tracking

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. Object tracker may also provide the complete region in the image that is occupied by the object at every time instant. The tasks of detecting the object and establishing correspondence between the object instances across frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm and then the tracker corresponds to objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating the object location and region information obtained from previous frames.

1.3.1 Classification of Tracking

1.3.1.1 Point Tracking

Objects detected in consecutive frames are represented by points and the association of the points is based on the previous object state which include object position and motion (Figure 1.2(a)). This approach requires an external mechanism to detect the objects in every frame.
1.3.1.2 Kernel Tracking

Kernel refers to the object shape and appearance. For example, the kernel can be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames (Figure 1.2 (b)). This motion is usually in the form of a parametric transformation such as translation, rotation and affine.

![Figure 1.2 Different Tracking Approaches. (a) Multipoint Correspondence, (b) Parametric Transformation Of A Rectangular Patch, (c, d) Two Examples Of Contour Evolution](image)

1.3.1.3 Silhouette Tracking

Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps (Figure 1.2 c and Figure 1.2 d). Both of these methods can essentially be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames.

1.3.2 Tracking based on Filter Techniques

The target to be tracked might be a complete object or a small area of an object. In either case, the feature of interest is typically contained within
a target region. The position will be described in X-Y coordinates in pixel units on the image (i.e. image coordinates). Object tracking may also become difficult when the orientation of the object changes from time to time. For these situations a motion model is developed, which indicates the behavior of the object over some time. The objects are tracked using filters that predict the objects position based on their current and previous states. The method used for tracking objects with different motions may vary.

![Diagram of object tracking](image)

**Case 1: Tracking The Object Without Position Prediction Might Be Successful, Case 2: Tracking Without Position Prediction Will Fail**

**Figure 1.3 Tracking Objects and its Prediction**

This process involves getting the frames from which object has to be tracked as input. The position of the detected objects in various frames is used to analyze the behavior of the object. The behavior of the object tells about the movement of the object. The movement of the object may be categorized into linear motion, non linear motion or highly non linear motion.

- Linear motion of the object means the position of the object in the subsequent frames varies linearly from time to time. When
the graph plotted against position of the object versus time gives a constant slope. This slope may be a straight line also when the speed remains constant.

- Non-linear motion of an object means that the position of the object may vary from time to time. At one instant the object may be moving at a lesser speed and suddenly the speed may increase and later decrease. There are small variations in the speed with which the object is moving. When a graph is plotted against position of the object versus the time taken, a variable slope is obtained. This graph obtained is never a straight line.

- Highly nonlinear motion of an object means the object's position may vary largely from time to time. At any instant the object may move at a very high speed, after some time the speed may decrease or increase and the objects position constantly varies with time. When a graph is plotted against position of the object and the time taken, the graph thus obtained has a highly variable slope.

In the proposed method, various filters like Kalman Filter (KF), Extended Kalman Filter (EKF) and Unscented Kalman Filter (EKF) are employed. In this process an Artificial Intelligence (AI) technique namely Radial Basis Function is used to select the filter needed to track the position of the object. This technique helps to select the desired filter based on the motion of the object. Once the filter is selected, the object position is estimated. The Kalman Filter produces estimates of the true values of measurements and their associated calculated values by predicting a value, estimating the uncertainty of the predicted value and computing a weighted
average of the predicted value and the measured value. The highest weight is
given to the value with the least uncertainty. The estimates produced by the
method tend to be closer to the true values than the original measurements
because the weighted average has a better estimated uncertainty than either of
the values that went into the weighted average.

1.3.3 Tracking based on Agent Technique

An agent refers to a component of software and/or hardware which
is capable of acting exactly in order to accomplish the tasks on behalf of the
user. When there is need to monitor multiple objects moving into a large
area, the work becomes too colossal to be handled by a single entity and here
the agent technology comes in handy. It allows the interconnection and
interoperation of multiple systems. Agents can react with their environment
and can also be proactive by initiating a task. The main application areas of
agents include information retrieval, e-commerce and trading, distributed
control and monitoring.

1.3.3.1 Multi-Agent System

Modern manufacturing systems are increasingly becoming highly
dynamic due to the integration with advanced information technology in
response to rapid changes in products and market conditions. A more flexible
platform is critically needed for developing a new generation of
manufacturing systems in order to address the challenges of uncertainty and
flexibility requirements (Stephen et al 2010). A Multi-Agent System (MAS)
consists of a network which are loosely-coupled computational autonomous
agents which can perform actions, they have resources at their disposal and
they possess the knowledge, capabilities or skills. They are situated in a
common environment and they can interact through a set of rules, namely an
interaction protocol. Single agent-based systems differ from multi-agent systems in:

- **Environment**: Agents take into account interference of other agents. They need to coordinate with others in order to avoid conflicts.
- **Knowledge/expertise/skills**: Here agents are distributed
- **Design**: Agents need not be homogeneous and implemented using different methodologies and languages
- **Interaction**: Agents interact following the rules of interaction

The interaction between the agents occurs when two or more agents are brought into a dynamic relationship through a set of reciprocal actions. The elements of interaction include the following.

- **Goals**: The agents objectives and goals are important. Goals of different agents can be conflicting
- **Resources**: Agents require resources (e.g., database) to achieve their objectives.
- **Expertise/skills/capabilities**: Agents may lack the necessary skills, expertise of capabilities for accomplishing one or more of their tasks.

The main advantages of MAS include flexibility, extensibility, robustness, reliability, speed, computational efficiency, reusability and reduced cost. Since the agents are mobile the trustworthiness of this third party is more expected. The security issues are to be considered for these mobile agents Greenberg (1998) and Niklas Borselius, 2002.
1.3.3.2 Agent Communication Language

Most treatments of communication in multi-agent systems borrow their inspiration from the Speech Act Theory. Speech Act theories are pragmatic theories of language. The different Speech Acts include Locutionary Act, Illocutionary Act and Perlocutionary Act. The Foundation of Intelligent Physical Agents (FIPA) has come up with a standard for ACL (Agent Communication Language). ACL provides agents with means of exchanging information and knowledge. It also separates the outer language (the intended meaning of the message) from the inner language (content language). The FIPA ACL provides several communication acts such as inform, agree, query-if, request, request-when, subscribe, etc. Agent communication has three main aspects.

- Syntax: how the symbols of communication are structured
- Semantics: what the symbols denote
- Pragmatics: how the symbols are interpreted

The structure of an ACL message has the following format.

- Performative
  - 20 performatives in FIPA ACL shown in Table 1.1
- Housekeeping
  - Sender, receiver, etc.
- Content
  - The actual content of the message.

Table 1.1 shows all the ACL Communication Acts (CA) which are used by the agents during interaction. The agents encode these CAs in the message whose format is shown in Table 1.1.
<table>
<thead>
<tr>
<th>Communication Act</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept-proposal</td>
<td>The action of accepting a previously submitted propose to perform an action</td>
</tr>
<tr>
<td>Agree</td>
<td>The action of agreeing to perform a requested action made by another agent. The agent will carry it out</td>
</tr>
<tr>
<td>Cancel</td>
<td>The agent wants to cancel a previous request</td>
</tr>
<tr>
<td>Cfp</td>
<td>Agent issues a call for proposals. It contains the actions to be carried out and any other terms of the agreement</td>
</tr>
<tr>
<td>Confirm</td>
<td>The sender confirms to the receiver the truth of the content. The sender initially believed that the receiver was unsure about it</td>
</tr>
<tr>
<td>Disconfirm</td>
<td>The sender confirms to the receiver the falsity of the content</td>
</tr>
<tr>
<td>Failure</td>
<td>Tell the other agent that a previously requested action failed</td>
</tr>
<tr>
<td>Inform</td>
<td>Tell another agent something. The sender must believe in the truth of the statement</td>
</tr>
<tr>
<td>Inform-if</td>
<td>Used as content of request to ask another agent to tell us if a statement is true or false</td>
</tr>
<tr>
<td>Not-understood</td>
<td>Sent when the agent did not understand the message</td>
</tr>
<tr>
<td>Propagate</td>
<td>Asks another agent to forward this same propagate message to others</td>
</tr>
<tr>
<td>Propose</td>
<td>Used as a response to a cfp. Agent proposes a deal</td>
</tr>
<tr>
<td>Proxy</td>
<td>The sender wants the receiver to select target agents denoted by a given description and to send an embedded message to them</td>
</tr>
<tr>
<td>query-if</td>
<td>The action of asking another agent whether or not a given proposition is true</td>
</tr>
<tr>
<td>Query-ref</td>
<td>The action of asking another agent for the object referred to by a referential expression</td>
</tr>
<tr>
<td>Refuse</td>
<td>The action of refusing to perform a given action and explaining the reason for the refusal</td>
</tr>
</tbody>
</table>
### Table 1.1 (Continued)

<table>
<thead>
<tr>
<th>Communication Act</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reject-proposal</td>
<td>The action of rejecting a proposal to perform some action during a negotiation</td>
</tr>
<tr>
<td>Request</td>
<td>The sender requests the receiver to perform some action. Usually to request the receiver to perform another communicative act</td>
</tr>
<tr>
<td>Request-when</td>
<td>The sender wants the receiver to perform some action when some given proposition becomes true</td>
</tr>
<tr>
<td>Request-whenever</td>
<td>The sender wants the receiver to perform some action as soon as some proposition becomes true and thereafter each time the proposition becomes true again</td>
</tr>
<tr>
<td>Subscribe</td>
<td>The act of requesting a persistent intention to notify the sender of the value of a reference and to notify again whenever the object identified by the reference changes</td>
</tr>
</tbody>
</table>

![Figure 1.4 FIPA ACL Message](image)

```
(inform
  :sender CAAgent
  :receiver DBAgent
  :content
    (Get location of Object No.1 )
  :language sl
  :ontology collision-avoidance
)
```

**Figure 1.4 FIPA ACL Message**

As shown in Figure 1.4, the first element of the message is a word which identifies the communicative act being communicated, which defines the principal meaning of the message. There then follows a sequence of message parameters, introduced by parameter keywords beginning with a colon character. No space appears between the colon and the parameter keyword. One of the parameters contains the content of the message, encoded
as an expression in some formalism. Other parameters help the message transport service to deliver the message correctly (e.g. sender and receiver), help the receiver to interpret the meaning of the message (e.g. language and ontology), or help the receiver to respond co-operatively.

A FIPA-ACL message contains 13 fields as shown in Table 1.2. The first and only mandatory field in the message is the performative field that defines the type of communicative act or speech act. By classifying the message using a performative, FIPA-ACL ensures that recipients will understand the meaning of a message in the same way as the sender, removing any ambiguity about the message’s content.

<table>
<thead>
<tr>
<th>Message field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performative</td>
<td>Type of communicative act</td>
</tr>
<tr>
<td>Receiver</td>
<td>Participant in communication</td>
</tr>
<tr>
<td>Sender</td>
<td>Participant in communication</td>
</tr>
<tr>
<td>Reply-to</td>
<td>Participant in communication</td>
</tr>
<tr>
<td>Content</td>
<td>Content of message</td>
</tr>
<tr>
<td>Language</td>
<td>Content language</td>
</tr>
<tr>
<td>Encoding</td>
<td>Encoding of content</td>
</tr>
<tr>
<td>Ontology</td>
<td>Ontology used</td>
</tr>
<tr>
<td>Protocol</td>
<td>Protocol for conversation</td>
</tr>
<tr>
<td>In-reply-to</td>
<td>Conversation control parameter</td>
</tr>
<tr>
<td>Reply-with</td>
<td>Conversation control parameter</td>
</tr>
</tbody>
</table>

In the proposed method agents involved are Migration Agent (MA) and Collision Avoidance Agent (CAA). The Migration Agents are in charge of handling movement of objects from one domain to another as the area of interest is divided into domains. A set of domain forms a cluster and it has a
cluster head. This cluster head monitors number of objects to be controlled by each agent in a domain. If there is no cluster head, agents get overloaded as time complexity of collision prediction depends on the number of objects that are under the agent control. The CAA takes care of inter and intra domain collisions and it uses agent with multiple database management systems for detecting and averting the collision.

1.3.4 Tracking based on System Learning

Neural Networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest. Other advantages include:

1. Adaptive Learning: An ability to learn how to do tasks based on the data given for training or initial experience.

2. Self-Organization: An ANN (Artificial Neural Network) can create its own organization or representation of the information it receives during learning time.

3. Real Time Operation: ANN computations may be carried out in parallel and because of their high level of accuracy they can be used in real-time scenarios.

One of the architecture is the Feed Forward network architecture. Feed-Forward ANNs allow signals to travel in one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-Forward ANNs tend to be straight forward
networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

The commonest type of ANN consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units.

1. The activity of the input units represents the raw information that is fed into the network.

2. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.

3. The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active and so by modifying these weights, a hidden unit can choose what it represents.

There are three major learning paradigms, each corresponding to a particular abstract learning task. They are supervised learning, unsupervised learning and reinforcement learning. Of which the supervised learning is the most suitable learning technique for the object tracking scenario.
1.3.4.1 Supervised Learning

In supervised learning, given a set of example pairs \((x, y) \in X \times Y\), the aim is to find a function \(f: X \rightarrow Y\) in the allowed class of functions that matches the examples. In other words, it is used to infer the mapping implied by the data; the cost function is related to the mismatch between the mapping and the data and it implicitly contains prior knowledge about the problem domain.

A commonly used cost is the mean-squared error, which tries to minimize the average squared error between the network's output, \(f(x)\) and the target value ‘y’ over all the example pairs. When one tries to minimize this cost using gradient descent for the class of neural networks called Multi-Layer Perceptrons, one obtains the common and well-known back propagation algorithm for training neural networks.

In the proposed work, Particle Grouping based tracking with Multi-Layer Perceptron (MLP) learning technique for weight computations and assertion is practiced. The MLP is chosen, since it is proven that MLP gives the most accurate solution for non-linear input and also provides substantial error convergence in minimal iterations. When there are multiple objects to be tracked there lies a serious issue in tracking. That is tracking of objects even in case of occlusion. Occlusion can be classified into three categories: self occlusion, inter-object occlusion and background scene structure occlusion. Self occlusion occurs where only part of the object is visible. This situation most frequently arises while tracking articulated objects. Inter-object occlusion occurs when two objects being tracked occlude each other or one object is hidden behind the other. Similarly, background scene structure occlusion occurs when the object is hidden due to the background. The proposed method handles all these occlusions with the help of bus topology.
1.3.4.2 Radial Basis Function

A radial basis function network is an artificial neural network that uses radial basis functions as activation functions. It is a linear combination of radial basis functions. The Radial basis function networks are neural nets consisting of three layers as shown in Figure 1.4.

The first input layer feeds data to a hidden intermediate layer. The hidden layer processes the data and transports it to the output layer. Only the tap weights between the hidden layer and the output layer are modified during training. Each hidden layer neuron represents a basis function of the output space, with respect to a particular centre in the input space. These functions are called “radial basis functions”. The activation function chosen is commonly a Gaussian kernel. This kernel is centred at the point in the input space specified by the weight vector. The closer the input signal is to the current weight vector, the higher the output of the neuron will be. The transformation from the input space to the hidden-unit space is nonlinear. On the other hand, the transformation from the hidden space to the output space is linear Radial Basis Function networks which are used commonly in function approximation and series prediction.

Some of the major application areas for RBFNs can be listed as follows:

- Image processing
- Speech recognition
- Time-series analysis
- Adaptive equalization
- Radar point source location
- Medical diagnosis
- Process faults detection
- Pattern recognition

As can be seen, the RBF networks are employed mostly in classification problems. The main classification problem involving RBF networks is the pattern recognition problem.

![Radial Basis Function-Neural Networks](image)

**Figure 1.5 Radial Basis Function-Neural Networks**

RBF is used in the proposed automatic Kalman filter bank to automate filter selection.

1.3.5 **Tracking Based on Computation**

1.3.5.1 **Quantum Computing**

A quantum computer is a device for computation that makes direct use of quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. It is a parallel computation in which all $2^n$ basis vectors are acted upon at the same time where ‘m’ represents bits. Quantum computers are different from digital computers based on transistors. Whereas digital computers require data to be encoded
into binary digits (bits), quantum computation utilizes quantum properties to represent data and perform operations on these data. Large-scale quantum computers could be able to solve certain problems much faster than any classical computer by using the best currently known algorithms, like integer factorization using Shor’s algorithm or the simulation of quantum many-body systems.

A quantum computer with a given number of qubits is fundamentally different from a classical computer composed of the same number of classical bits. Qubits can be entangled together, all possible combinations of their states can be simultaneously used to perform a computation. Due to superposition it can act on all possible states simultaneously. For example, to represent the state of a n-qubit system on a classical computer would require the storage of $2^n$ complex coefficients. For example: Consider first a classical computer that operates on a three-bit register. The state of the computer at any time is a probability distribution over the $2^3 = 8$ different three-bit strings 000, 001, 010, 011, 100, 101, 110, 111. If it is a deterministic computer, then it is in exactly one of these states with probability 1. The state of a three-qubit quantum computer is similarly described by an eight-dimensional vector $(a, b, c, d, e, f, g, h)$, called a ket. Here the sum of the squares of the coefficient magnitudes, $|a|^2 + |b|^2 + \ldots + |h|^2$, must equal one. Moreover, the coefficients are complex numbers. For example, the state $(a, b, c, d, e, f, g, h)$ in the computational basis can be written as:

$$a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle$$

Where $|010\rangle = (0, 0, 1, 0, 0, 0, 0, 0)$
The qubits can be in a superposition as in Figure 1.6. This describes the qubit positions in states 0, 1 and in the intermediate state whether it can be either 4 or 5.

![Figure 1.6 Superposition states]

The proposed Quantum Computing method uses the quantum computing method to detect multiple objects. The specification of the objects is stored as qubits since simultaneous access is done. These specifications include the color, shape and size of the object.

1.3.5.2 Linear Programming Approach

Linear programming (LP) or Linear Optimization (LO) is a mathematical method for determining a way to achieve the best outcome, such as maximum profit or lowest cost in a given mathematical model for some list of requirements represented as linear relationships. Linear programming is a specific case of mathematical programming (mathematical optimization).

Linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints. Its feasible region is a convex polyhedron, which is a set defined as the
intersection of finitely many half spaces, each of which is defined by a linear inequality. Its objective function is a real-valued affine function defined on this polyhedron. A linear programming algorithm finds a point in the polyhedron where this function has the smallest (or largest) value if such point exists.

Linear programming are problems that can be expressed in canonical form:

\[
\begin{align*}
\text{maximize} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
\text{and} & \quad x \geq 0
\end{align*}
\]

Where ‘x’ represents the vector of variables (to be determined), ‘c’ and ‘b’ are vectors of (known) coefficients and ‘A’ is a (known) matrix of coefficients. The expression to be maximized or minimized is called the objective function \(c^T x\) in this case). The equations \(Ax \leq b\) are the constraints over which the objective function is to be optimized. (In this context, two vectors are comparable when every entry in one is less-than or equal-to the corresponding entry in the other. Otherwise, they are incomparable.)

Linear programming can be applied to various fields of study. It is used in business and economics, but can also be utilized for some engineering problems. Industries that use linear programming models include transportation, energy, telecommunications and manufacturing. It has proved useful in modeling diverse types of problems in planning, routing, scheduling, assignment and design.

The standard form is the usual form of describing a linear programming problem. It consists of the following four parts:
A linear function to be maximized

e.g. \( f(x_1, x_2) = c_1 x_1 + c_2 x_2 \)

Problem constraints are of the following form

e.g.

\[
\begin{align*}
    a_{11} x_1 + a_{12} x_2 & \leq b_1 \\
    a_{21} x_1 + a_{22} x_2 & \leq b_2 \\
    a_{31} x_1 + a_{32} x_2 & \leq b_3 
\end{align*}
\]

Non-negative variables

\[
\begin{align*}
    x_1 & \geq 0 \\
    x_2 & \geq 0 
\end{align*}
\]

Non-negative right hand side constants

\( b_i \geq 0, \quad i = 1, 2, 3 \)

The problem is usually expressed in matrix form and then becomes:

\[
\max \{ c^T x \mid 0 \leq Ax \leq b \land x \geq 0 \}
\]

The proposed Quantum Computing method develops a graph based on the path travelled by each object which in turn leads to a linear program and thereby the state of the objects are found.

1.3.6 Spatio-Temporal Database (STD)

Traditional Database Management System (DBMS) technology provides a potential foundation for development of moving objects applications, but not fully tuned for this purpose. The reason is that there is a
critical set of capabilities that are needed by Moving Objects Database (MOD) applications which are lacking in existing systems. Existing systems are not well equipped to update dynamically changing data. The key problem is how to locate the moving objects and enable a DBMS to predict the future location by using the efficient index and query manners. Generally, a query in dynamic applications involves spatial objects and temporal constraints. Spatio-temporal database captures spatial and temporal aspects of data. It aims to support extensions to existing models of spatial information systems along with time in order to better describe any dynamic environment. Spatio-temporal databases support time information as well as spatial management for the object at the same moment and can deal with location changes over time. They can be used in the various application areas such as geographic information system, urban plan system, car navigation system, etc.

It shows how flexible querying enhances standard querying expressiveness in many different ways, with the aim of facilitating the extraction of relevant data and information. Flexible spatial and temporal reasoning denotes qualitative reasoning about dynamic changes in the spatial domain, characterized by imprecision or uncertainty (or both). Spatio-temporal database systems are oriented to the integration and management of space, time and spatio-temporal concepts. When dealing with dynamic location aware services, the trajectory is one of the key primitive concepts from spatio-temporal database. Retrieving and examining trajectories favor the study of moving object behavior in space and time. Once the trajectories are recorded and examined, care should be taken to avoid collision between the mobile objects as these collisions may result in changing the trajectory of the objects.
1.3.6.1 Queries

A query is a predicate over the database history (rather than a predicate over a single database state, as in traditional databases). The answer to a query is defined when the predicate is satisfied and it consists of the set of instantiations of the variables that satisfy the predicate. There is a need to distinguish between three types of queries instantaneous, continuous and persistent (Table 1.3). The same query may be entered as instantaneous, continuous and persistent, producing different results in each case. These types differ depending on the history from which the query is evaluated and on the evaluation time. In contrast, the traditional databases are simpler. An instantaneous query is a predicate on the current database state and a continuous query is a predicate on each one of the future states. Finally, the third type of query is a persistent query. Formally, a persistent query at time ‘t’ is a sequence of instantaneous queries on the infinite history starting at ‘t’.

Table 1.3 Types of queries in STDs

<table>
<thead>
<tr>
<th>Query type</th>
<th>Scenario</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instantaneous</td>
<td>Finding out the location of an object at a time instant in the present</td>
<td>Select PLACE, OJBID from TableName where OBJID=7 AND time&lt;=&quot;10:08:33&quot;</td>
</tr>
<tr>
<td>Continuous</td>
<td>Finding out the location of an object at a time instant in the future</td>
<td>Select PLACE, OJBID from TableName where OBJID=7 AND time&gt;&quot;10:08:33&quot;</td>
</tr>
<tr>
<td>Persistent</td>
<td>Finding out the objects crossing a particular location from a given time instant</td>
<td>Select OJBID from TableName where PLACE=&quot;A&quot; AND time&gt;&quot;10:08:33&quot;</td>
</tr>
</tbody>
</table>

Based on the temporal state of the database, queries can be divided into past, present and future queries.
1.3.6.2 Spatio-Temporal Access Methods

A data structure, index is used to efficiently process and access spatio-temporal queries, one needs specific access methods relying on a data structure called an index. A new issue that arises in modern applications involves the efficient manipulation of (static or moving) spatial objects and the relationships among them. As a result, modern database systems should be able to efficiently support that type of data. Towards this goal, appropriate extensions of multidimensional access methods can be exploited in order to index and retrieve spatiotemporal objects, satisfying users demands (Theodoridis 2010). Processing a spatial query leads to the execution of complex and costly geometric operations. For operation such as point queries sequentially scanning and checking involves a large number of disk accesses and leads to expensive evaluation. Hence, both the time-consuming geometric algorithm and the large volume of spatial collections stored on secondary storage motivate the efficient design of spatial access methods, which reduce the set of objects to be processed. A set of criteria was proposed to characterize spatio-temporal data and access structures in the class of interest:

- Data type whether it supports points or regions.
- Temporal type whether the supported temporal dimension is that of valid time, transaction time or both.
- Database mobility whether the changes in the cardinality of the database or the spatial position of the data items or both, can change over time.
- Data loading whether the dataset is known priori or not, whether only updates concerning the current state can be made or any state can be updated.
- Object representation which abstraction (e.g. MBRs-Minimum Bounded Rectangles) is used to represent the spatial objects.
The proposed method collaborates with R tree, k-d tree, trie/graph for locating the dynamic objects using spatio-temporal database and to avoid collisions among the objects. Since the database is vast and takes more time for retrieval of data, it is represented in the form of index tree. Spatio-temporal database allows modeling of the interactions, enables the identification of normal patterns and detect unusual events. A multi dimensional Rk-d tree is developed by combining the features of R tree and k-d tree by initially dividing the space into regions (R tree) and performing division on those regions (k-d tree) and this Rk-d tree provides optimal search performance of time complexity \( O(n\log(\log n)) \). Additional feature to Rk-d tree is Rk-d trajectory trie which is used to index the trajectories of moving objects. This is constructed as a two layered structure, where the upper layer is the Rk-d tree which indexes the moving objects and the lower layer is the trie tree that indexes the trajectories corresponding to the moving objects.

Search over graph databases has attracted much attention due to its usefulness in many fields, such as the intrusion detection in network traffic data and pattern matching over users visiting logs. However, most of the existing work focuses on search over static graph databases while in many real applications graphs are changing over time (Lei Chen and Changliang Wang, 2010). This proposed Rk-d Trajectory graph is dynamic and collision is avoided using Bayesian probability and critical region identification.

The disadvantage of trie tree being the high computational complexity and querying cost as trajectory trie is maintained for each object in the Rk-d tree. Hence a more enhanced data structure Rk-d trajectory graph trajectories of MOVObjs and graph specific queries is proposed. Graph is created and each vertex of the graph carries a hash table which stores the time and corresponding moving object’s unique ID. The collisions are rectified by
forming groups among objects for identifying the critical region using Lyapunov exponent. The probability of migration is checked by Bayes theorem as it yields higher accuracy than other computations and history of evidences is collected which further increases the degree of belief for checking the migration of objects.

1.3.7 Problems in Object Tracking

Tracking objects is complex due to:

- Loss of information caused by projection of the 3D world on a 2D image,
- Noise in images,
- Complex object motion,
- Non-rigid or articulated nature of objects,
- Partial and full object occlusions,
- Complex object shapes,
- Scene illumination changes and
- Real-time processing requirements
- Error merge and Object labeling (Scott 2011)

1.4 COLLISION AVOIDANCE

Collision detection and avoidance are one of the most important issues in an environment with mobile objects. In the proposed Rk-d trajectory tree method collision is avoided by analysis of the past, present and future trajectories of the objects. The challenge lies in detecting and avoiding collision in large expanding environments where tracking the movement of objects becomes difficult. In this case, grouping of the larger environment is
done. Once grouped, the behaviors of objects in each group and their migration patterns have to be observed and managed. The process of objects moving from one group to another is termed as migration. The region where the probability of migration is high is termed as ‘migration sensitive points’. The region where there is a high probability of collision is termed as ‘critical region’. Effective approaches have been used to identify these regions and avoid collision in the environment.

1.5 PROPOSED ARCHITECTURE

This thesis proposes and designs six various ways of object tracking with different technologies distinguished by accuracy, range, frame-rate and different type of objects. Targets are selected in an automatic and data-driven manner. Figure 1.7 portrays the overall architecture.

![Proposed Overall Architecture]

**Figure 1.7 Proposed Overall Architecture**
1.6 CONTRIBUTIONS MADE IN THIS THESIS

This research work contributes the following to track the objects

- Automatic Kalman Filter Bank for single object tracking
- Particle Grouping for single and multiple object tracking
- Agent based framework along with retinotopic representation for time to collision for multiple object tracking
- Rk-d Trajectory Graph to avoid collision for multiple objects
- Quantum Computing for object detection and linear programming for multiple object tracking
- Bus topology along with superposition estimation and distance formulation to handle various kinds of occlusions

1.7 THESIS ORGANIZATION

Chapter 2 relates survey details about the existing methodologies and their drawbacks.

Chapter 3 deals with Signal Processing based Object Tracking. It explains about proposed Automatic Kalman Filter Bank which portrays the Automatic Filter Bank and its performance. Proposed Particle Grouping Based approach reveals about single object tracking and its performance.

Chapter 4 comprises Database Technique for Multiple Object Tracking. This engages with proposed Agent based approach which gives the framework related with agents and their corresponding communications and their performance. It also deals with the proposed Rk-d Graph Trajectory to avoid collision based on trajectory tuning.
Chapter 5 accords with Artificial Intelligence technique for Multiple Object Tracking. This gives detailed methods on proposed Quantum Computing which works on object tracking in the form of Quantum Bits and proposed Particle Grouping with Multi-layer Perceptron approach pact with information about the object region and tracks the objects even in case of critical occlusions.

Chapter 6 consolidates the results and discussion of all the six individual proposed works and also the overall result.

Chapter 7 discusses the concluding remarks of the proposed works.

1.8 SUMMARY

This Chapter summarizes the following

- Needs for Object Tracking
- Problems in Object tracking
- Abstracts the six proposed works for object tracking
  - First method deal with Kalman Filters
  - Second method pact with Particle and Kalman Filter
  - Third method seeks the help of agent and Multiple Spatio- temporal Database
  - Fourth method collaborates r tree, k-d tree and graph to form Rk-d Graph
  - Fifth method accords with quantum Computing
  - Sixth method exploit Bus topology and Multi –layer Perceptron
- List the Organization of the thesis