What is perception in AI?

●Perception is a process to interpret, acquire, select and then organize the sensory information that is captured from the real world.

For example: Human beings have sensory receptors such as touch, taste, smell, sight and hearing. So, the information received from these receptors is transmitted to human brain to organize the received information.

●According to the received information, action is taken by interacting with the environment to manipulate and navigate the objects.

●Perception and action are very important concepts in the field of Robotics. The following figures show the complete **autonomous robot.**

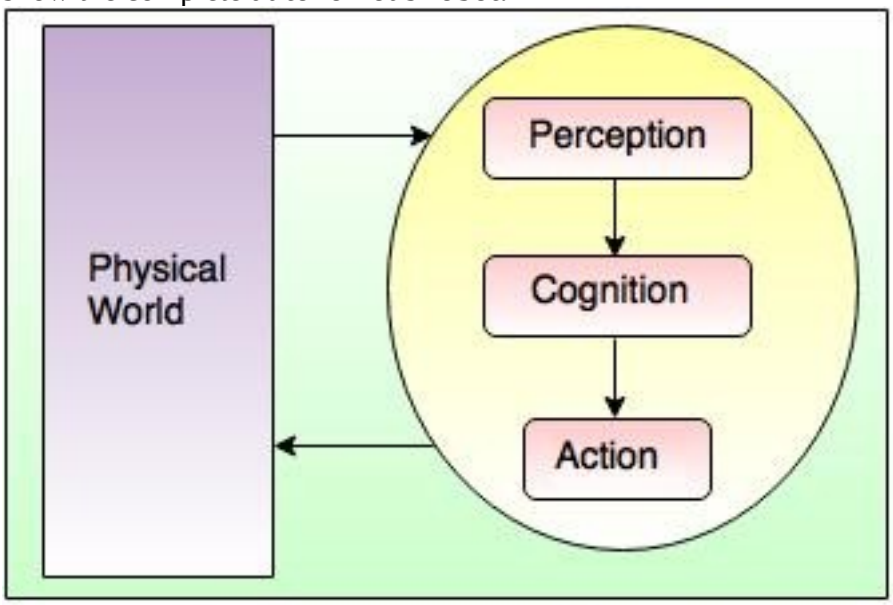

Fig: Autonomous Robot

●There is one important difference between the artificial intelligence program and robot. The AI program performs in a computer stimulated environment, while the robot performs in the physical world.

For example:

In chess, an AI program can be able to make a move by searching different nodes and has no facility to touch or sense the physical world.

However, the chess playing robot can make a move and grasp the pieces by interacting with the physical world.

Image formation in digital camera

Image formation is a physical process that captures object in the scene through lens and creates a 2-D image.

Let's understand the geometry of a pinhole camera shown in the following diagram.

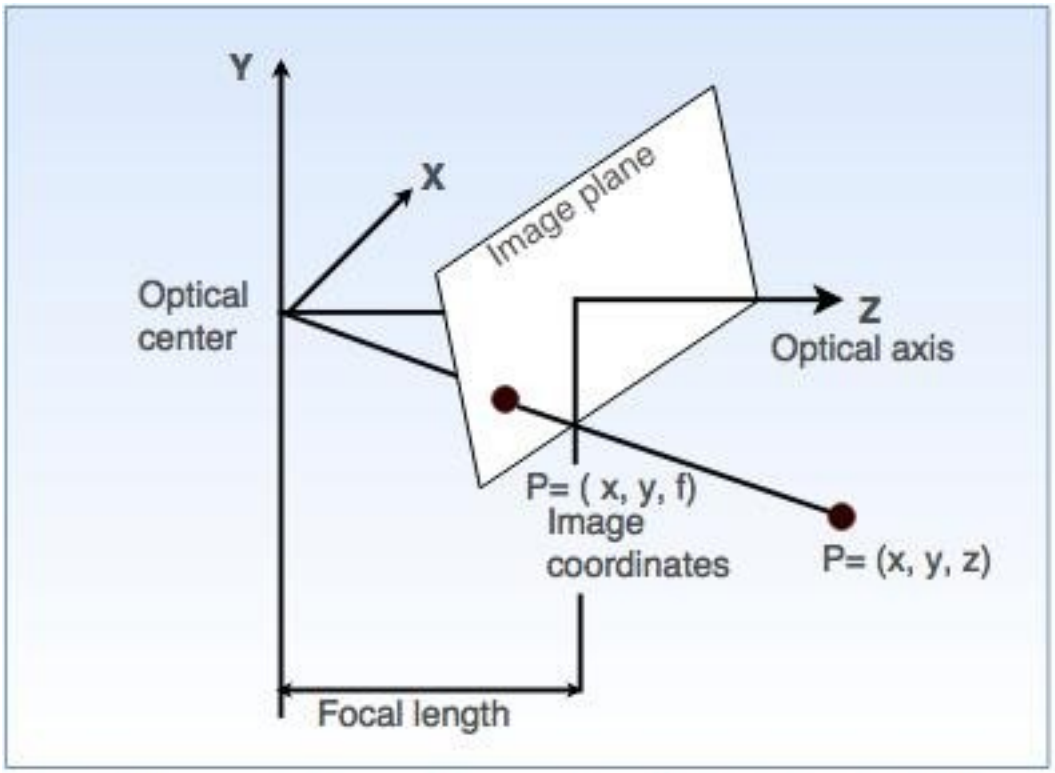

Fig: Geometry of Image Formation in the pinhole camera

In the above figure, an optical axis is perpendicular to the image plane and image plane is generally placed in front of the optical center.
So, let P be the point in the scene with coordinates (X,Y,Z) and P' be its image plane with coordinates (x, y, z).

If the focal length from the optical center is f, then by using properties of similar triangles, equation is derived as,

-x/f = X/Z so x = - fX/Z .....equation (i)
-y/f = -Y/Z so y = - fY/Z .....equation (ii)

These equations define an image formation process called as perspective projection.

What is the purpose of edge detection?

- Edge detection operation is used in an image processing.
- The main goal of edge detection is to construct the ideal outline of an image.
- Discontinuity in brightness of image is affected due to:

i) Depth discontinuities

ii) Surface orientation discontinuities

iii) Reflectance discontinuities

iv) Illumination.

3D-Information extraction using vision

Why extraction of 3-D information is necessary?

The 3-D information extraction process plays an important role to perform the tasks like manipulation, navigation and recognition. It deals with the following aspects:

1. Segmentation of the scene

- The segmentation is used to arrange the array of image pixels into regions. This helps to match semantically meaningful entities in the scene.
- The goal of segmentation is to divide an image into regions which are homogeneous.
- The union of the neighboring regions should not be homogeneous.
- **Thresholding** is the simplest technique of **segmentation.** It is simply performed on the object, which has an homogeneous intensity and a background with a different intensity level and the pixels are partitioned depending on their intensity values.

2. To determine the position and orientation of each object

- Determination of the position and orientation of each object relative to the observer is important for manipulation and navigation tasks.

For example: Suppose a person goes to a store to buy something. While moving around he must know the locations and obstacles, so that he can make the plan and path to avoid them.

- The whole orientation of image should be specified in terms of a three dimensional rotation.

3. To determine the shape of each and every object

- When the camera moves around an object, the distance and orientation of that object will change but it is important to preserve the shape of that object.

For example: If an object is cube, that fact does not change, but it is difficult to represent the global shape to deal with wide variety of objects present in the real world.

- If the shape of an object is same for some manipulating tasks, it becomes easy to decide how to grasp that object from a particular place.
- The **object recognition** plays most significant role to identify and classify the objects as an example only when the geometric shapes are provided with color and texture.

What is a Robot Architecture?

There are many different ways in which a robot control program can be put together. In order to program a robot in a structured and principled fashion, we use an appropriate robot control architecture.System developers have typically relied upon robotic architectures to guide the construction of robotic devices and for providing computational services (e.g., communications, processing, etc.) to subsystems and components.

Robot ArchitectureA control architecture provides a set of principles for organizing a control system.It provides structure and constraints which aid the designer in producing a well-behaved controller.To be successful a system designer has to decide how (in what order? with what priority?) does he put together multiple feedback controllers in a principled fashion and how to scale up control to more complex robots, which generally have to deal with many behaviors at once.How would you put multiple feedback controllers together?How would you decide which one to use when and for how long and in what priority relative to the others?

Robot Architecture Major Classes/Categories

Intuitively, this means that there are infinitely many ways to structure a robot program, but they all fall into one of major classes/categories of control:Deliberative Control : Think hard, act later.SPA, serial, complete each step first – then proceedReactive Control : Don’t think, (re)act.Direct connection between perception to action, no memory, no planning.Hybrid Control : Think and act independently, in parallel.Deliberative and Reactive modules run independently at different time scalesBehavior-Based Control : Think the way you act.Distributed by behavioral task decompositionEach behavior has its restricted planning and execution capabilities

The Choice of the Control Architecture

In many cases, it is impossible to tell, just by observing a robot's behavior, what control architecture it is using. Only for simple robots, it is often the case.However, when it comes to more complex robots, i.e., robots that have to deal with complex environments and complex tasks, the control architecture becomes very important.The different properties of an environment that will impact the robot's controller (and therefore the choice of control architecture):noisy,speed/response time of sensors and effectorstotal/partial hidden state/ observablediscrete v. continuous state ; static v. dynamic ...Similarly, the properties of the robot's task impact the choice of the control architecture. The task requirements can constrain the architecture choice.

Parallel Processing Paradigm.

As robot control is engaged to deal with more complex problems, centralized supervisory architectures encounter barriers to real time performance caused by computational complexity coupled with insufficient computing power and sensor resources.Despite startling advances in hardware and software technology and similarly surprising cost reductions, these fundamental barriers remain unchanged.The parallel-processing paradigm may be the only technology to challenge this fact.

Asynchronous and Synchronous processes

The other leading architectural trend is typified by a mixture of asynchronous and synchronous control and data flow.Asynchronous processes are characterized as loosely coupled and event-driven without strict execution deadlines.Synchronous processes, in contrast, are tightly coupled, utilize a common clock and demand hard real- time execution.

Some Criteria for Selecting a Control Architecture

support for parallelism: the ability of the architecture to execute parallel processes/behaviors at the same time.hardware targetability:how well the architecture can be mapped onto real-robot sensors and effectors.how well the computation can be mapped onto real processing elements (microprocessors).run-time flexibility: does the architecture allow run-time adjustment and reconfiguration? It is important for adaptation/learning.modularity: how does the architecture address encapsulation of control, how does it treat abstraction?Does it allow many levels, going from feedback loops to primitives to agents?Does it allow re-use of software?

Some Criteria for Selecting a Control Architecture

niche targetability: how well the architecture allows the robot to deal with its environment robustness: how well does the architecture perform if individual components fail? How well does it enable and facilitate writing controllers capable of fault tolerance?

ease of use: how easy to use and accessible is the architecture? Are there programming tools and expertise?

performance: how well does the robot perform using the architecture? Does it act in real-time? Does it get the job done? Is it failure-prone?

The above issues allow us to compare and evaluate different architectures relative to specific robotic designs, tasks, and environments. But not all tasks, environments, and designs are comparable.

Time Scale.

Time-scale is an important way of distinguishing control architectures.Reactive systems respond to the real-time requirements of the environment,while deliberative system look ahead (plan) and thus work on a longer time-scale. Hybrid systems must combine the two time-scales in an effective way, usually requiring a middle layer; consequently they are often called three-layer architectures.Finally, behavior-based systems attempt to bring the different time-scales closer together by distributing slower computation over concurrent behavior modules.

Representation Another key distinguishing feature between architectures is representation of the world/environment, also called world modeling.Some tasks and architectures involve storing information about the environment internally, in the form of an internal representation of the environment.For example, while exploring a maze, a robot may want to remember a sequence of moves it has made (e.g., "left, left, right, straight, right, left"), so it can back-track and find its way.Thus, the robot is constructing a representation of its path through the maze.The robot can also build a map of the maze, by drawing it using exact lengths of corridors and distances between walls, etc. .This is also a representation of its environment, a model of the world.If two robots are working together, and one is much slower than the other, if the fast robot remembers/learns that the other is always slower, that is also a type of a model of the world, in this case, a model of the other robot.

Different World Models.

There are numerous aspects of the world that a robot can represent/model, and numerous ways in which it can do it, including:

* spatial metric or topological: maps, navigable spaces, structures
* objects instances of detectable things in the world
* actions outcomes of specific actions on the self and environment
* self/ego stored proprioception: sensing internal state, self- limitations, etc.
* intentional goals, intended actions, plans* symbolic abstract encoding of state/information

Amount and Type of Representation

The amount and type of representation or modeling used by a robot is critically related to the type of control architecture it is using.Some models are very elaborate; they take a long time to construct and are therefore kept around possibly throughout the lifetime of the robot's task (for example detailed metric maps).Others may be relatively quickly constructed and transient, used quickly and discarded or updated (for example the next few steps in a short plan, the immediate

goal, etc.)How long it takes to construct/build a model is an important aspect of the robot's controller.

Amount and Type of Representation

How long it takes to use it is equally important.Consider maps again:

* it takes a long time to construct an accurate and detailed metric map, because it requires exploring and measuring the environment.

* furthermore, it takes time to use such a map as well (even if it took no time to construct it, but it was given to the robot by the designer);

* one must find the free/navigable spaces in the map, and then

* search through those to find the best path to the goal.Similarly, any internal model can require time to construct and be used, and these timing requirements directly affect the time-scale of the controller.

Control Architectures and Internal States/Representations.

A control architecture can make it easy or difficult to store internal models (just as a programming language can make it more or less convenient to build and store structures) and manipulate them, i.e., compute with them.How internal state, i.e., information a robot system keeps around, relates to representation.- In principle, any internal state is a form of representation.- What matters is the form and function of that representation.- The reason two different terms are employed is as follows: state refers to the "status" of the system itself, whereas "representation" refers to arbitrary information that may be contained in the system.

Monitoring approaching the goal

We define a system as “intelligent” by the way it achieves the Global Goal. We need a mechanism which has the capability to plan the needed actions to enable reaching the final (Global) goal.Evaluate the current situationReport to userRe-plan according to the contingent events that already occurred.