# Control Behavior of 3D Humanoid Animation Object Using Reinforcement Learning

# Outline

- Background
- Introduction
- Overview of The Framework
- Learning Algorithm on Manifold
- Experimental Result
- Conclusion and Discussion

# Background

- Humanoid avatar interacts with the graphical environments

# Background

- The ability to learn is a potentially compelling and important quality for interactive 3D human avatars or virtual humans.
- Implementing the method of reinforcement learning to guide the avatar's action.

# Introduction

- By embedding the kind of learning of which avatars are capable into synthetic characters, we can provide them with an equally robust mechanism for adapting in meaningful ways to act like people in the 3D virtual environment which they are interacting.

# Introduction

- Unlike the real world, the graphical environment is simpler. Thus, we can concentrate our research on some basic elements which influence the behavior of virtual humans, such as geometric and physical properties.

# Introduction

- Enabling avatars to take advantage of predictable regularities in their world.
- Allowing them to make maximal use of any supervisory signals, either explicit or implicit, that the graphical world offers.
- Making them easy to be trained by interacting with virtual environment.

# Introduction

- The best action to perform in a given context, which forms a given action is reliable in producing reward.
- The relative reliability of its actions in producing a reward and altering its choice of action accordingly.
- To recognize new and valuable contexts from graphical environment to synthesize new and reasonable actions the training.

# Introduction

- To design a framework to guide the behavior of graphical avatars by the learning algorithm
- To improve the algorithms and make humanoid avatars behave more like real human being and intelligently.

# Overview of The Framework

- By piecing together many some different type of short motion clips, we can further create novel and realistic motions.
- Depends on the motion action context of computer animation, we are able to describe all the subtleties of real human motion.

# Overview of The Framework

- In interactive applications, new input is continuously arriving and the decision for selecting the next clip needs to be made in a very short amount of time.
- In many situations, our application must have interactive property.

# Overview of The Framework

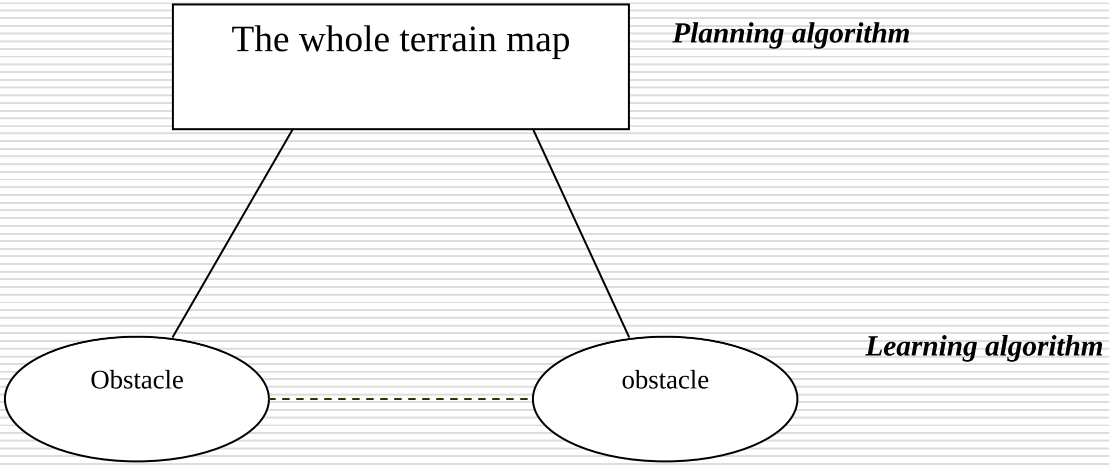
- Only local search can be performed to generate motions in response to the dynamic 3D environments. For instance, towards a task to grasp an object at a particular location, first step is to plan a total strategy of the action, rather then trying to search a point-to-point path on the graph.

# Overview of The Framework

- The whole motion can be separate into different clips.
- On the structure, it is **Hierarchical reinforcement learning** .

# Overview of The Framework

- If the avatar want to go through a terrain full of obstacles:

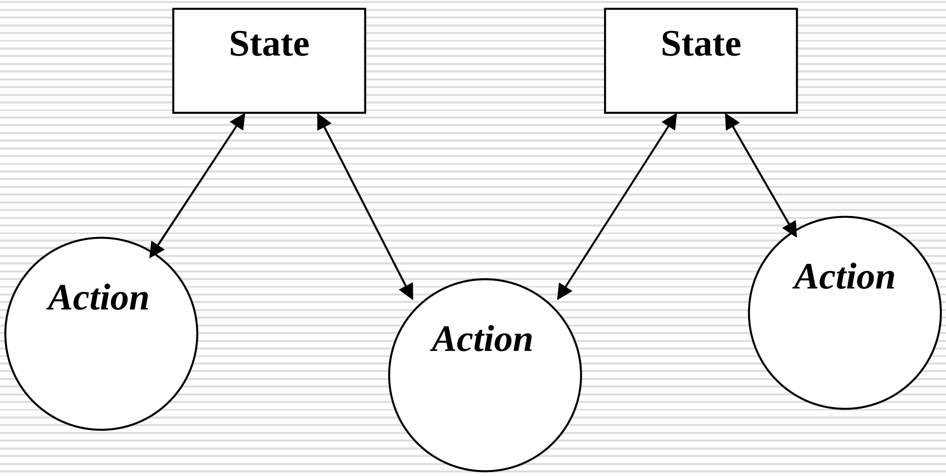


# Overview of The Framework

- On the other hand, in most situations, if the elements of states are too trivial, they cannot support the Markov Decision Process' property. For instance, rotation of joints of a avatar is not proper to be described as a MDP.

# Overview of The Framework

- To describe the processes as a continuous semi-Markov processes in according to the property of their exit streams.



# Overview of The Framework

- To describe the sub task of the whole strategy as a MDP.
- Using dynamic programming to find optimal policy $\Pi$.
- Using reinforcement learning method to train the virtual human.

# Learning Algorithm on Manifold

- If a process is a MDP, then it can be represent a statistical manifold.
- The atomic actions can be organized by some MDP algorithms.

# Learning Algorithm on Manifold

- In this problem, to achieve our goal, the states and actions should support a important property: continue.

# Learning Algorithm on Manifold

- Discretizing the continuous MDP.
- A discrete Markov decision process (MDP) can be defined by $M = (S, A, P^a_{s,s'}, R^a_{s,s'})$ .
- A finite set of discrete states: S, a finite set of actions: A, a transition model $P^a_{s,s'}$ specifying the distribution over future states s' when an action a is performed in state s.
- A corresponding reward model $R^a_{s,s'}$ specifying a scalar cost or reward.

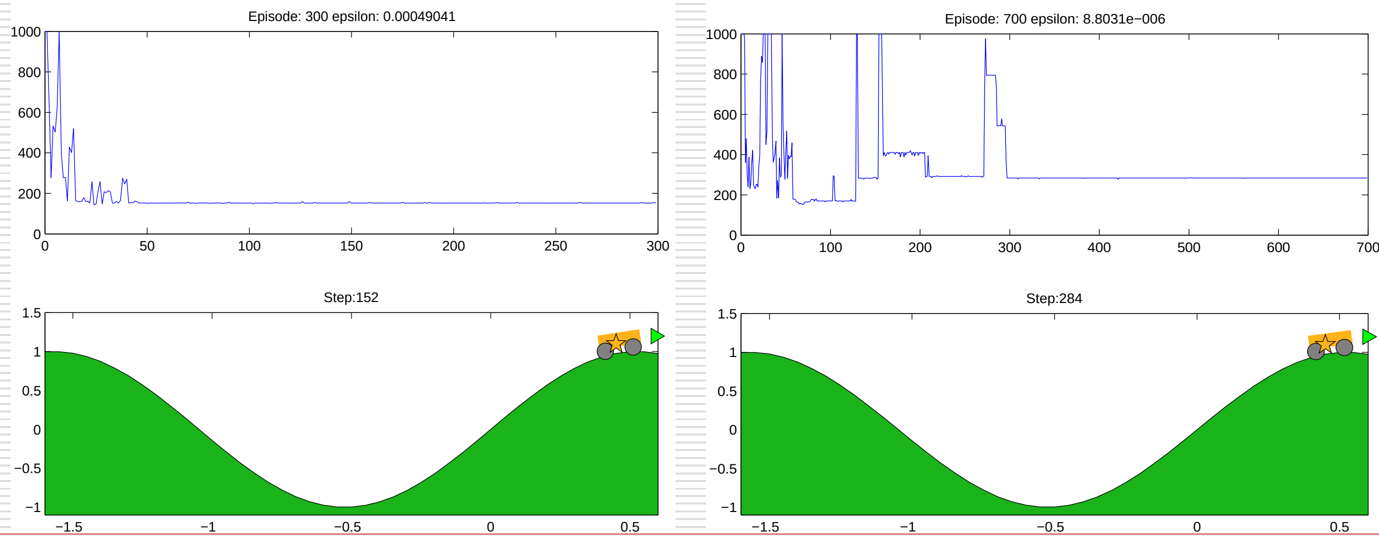
# Learning Algorithm on Manifold

- Any optimal policy $\Pi$ defines the same unique optimal value function V which satisfies the constraints:

$$V(s) = \max_a (R^a_{s,s'} + \gamma \sum_{s' \in S} P^a_{s,s'} V(s'))$$

# Learning Algorithm on Manifold

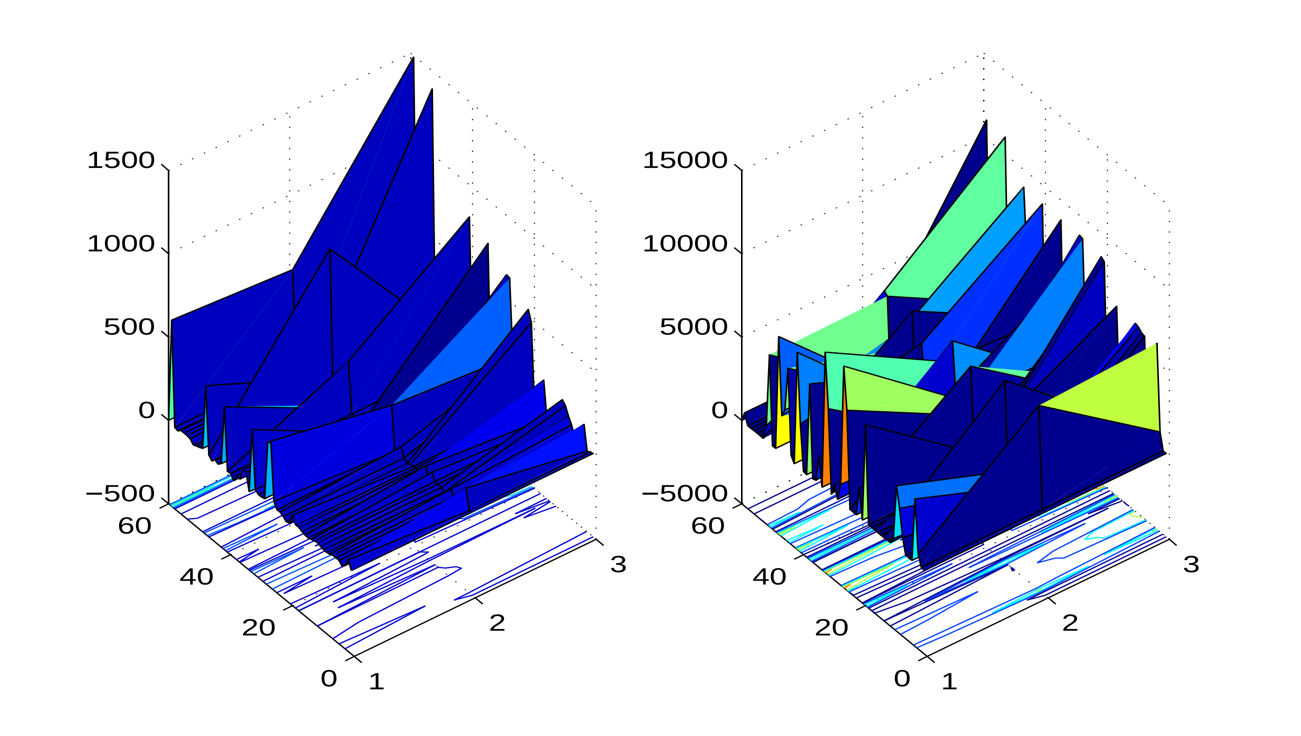
- Thus the question is induced to find out an optimal policy $\Pi$ on a manifold. We used the Mountain Car Problem to test our algorithm:



# Learning Algorithm on Manifold

- Using Temporal-difference (TD) learning methods exploit the property that the value of a state. The value function V(s) can be estimated as the sum of the immediate reward received.

# Learning Algorithm on Manifold

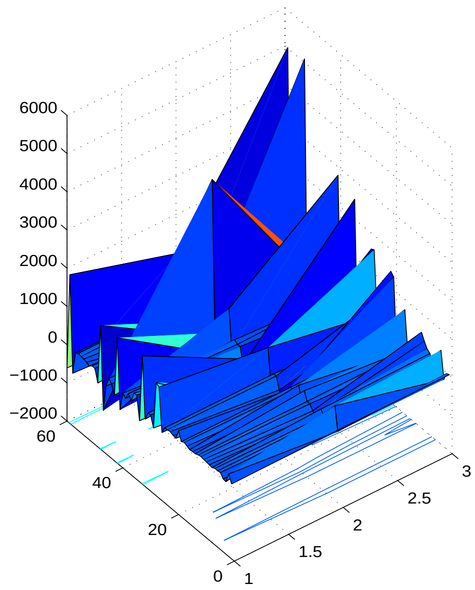
- To treat the value function as a manifold which is decided by states and actions:
- 

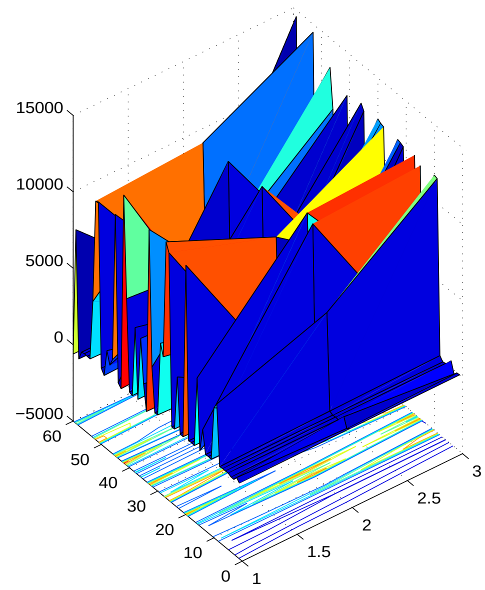
# Learning Algorithm on Manifold

- Formally, a manifold M is a locally Euclidean set, with a homeomorphism from any open set containing an element $p \in M$ to the n-dimensional Euclidean space. Manifolds with boundaries are defined using a homeomorphism that maps elements to the upper half plane. In smooth manifolds, the homeomorphism becomes a diffeomorphism.

# Learning Algorithm on Manifold

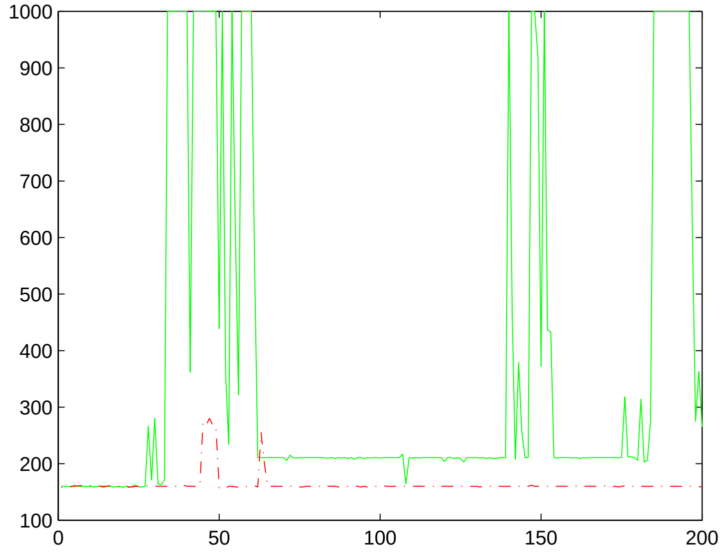
- Using laplacian to get the edge of the manifold and improve the Qtable:

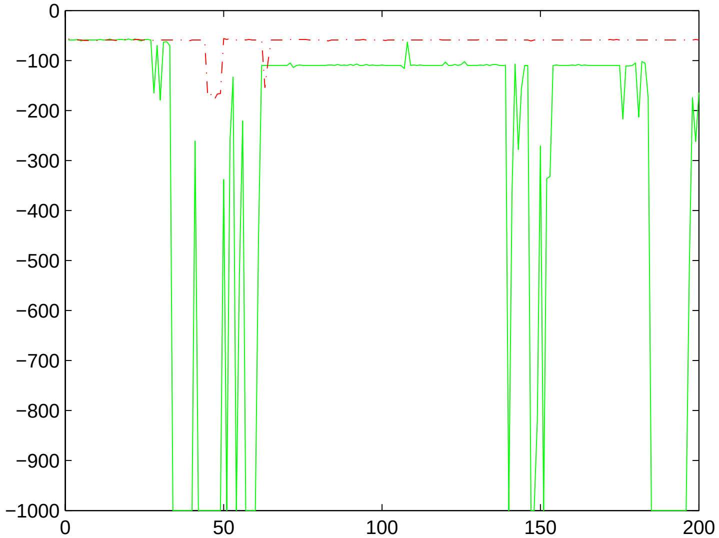


# Learning Algorithm on Manifold

- Comparison between the original Qtable and improved Qtable:
- Steps to reach the goal                Rewards to get



# Learning Algorithm on Manifold

- To treat the value function as a manifold and improve it after a short term training can achieve good result.

# Experimental Result

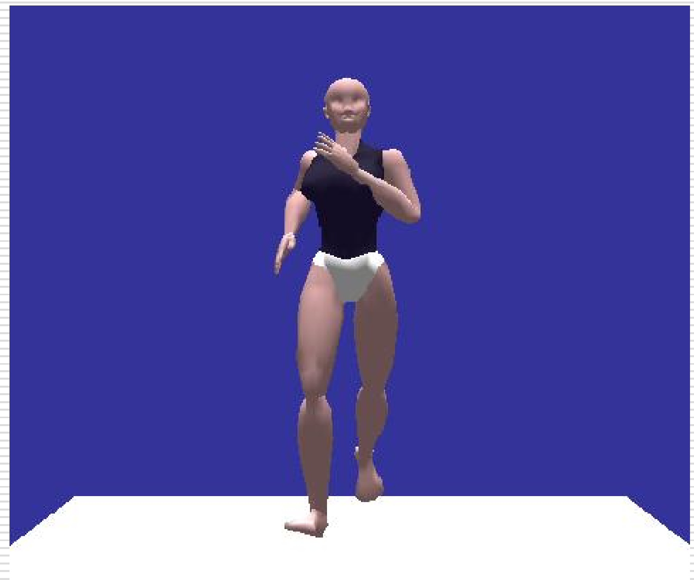
- We used two virtual humans : "Cally" and "Paladin". Where they have a different ability and will behave differently to interact with the environments. The task of the both avatars is to access the object (i.e. a beast), then interacts with it (i.e. shot a arrow to it).

# Experimental Result

- Our 3D graphical platform is the open source software : Delta3D.
- Delta3D is an Open Source engine which can be used for games, simulations, or other graphical applications. Its modular design integrates other well-known Open Source projects such as Open SceneGraph(OSG), Open Dynamics Engine(ODE), Character Animation Library(CAL3D), and OpenAL.

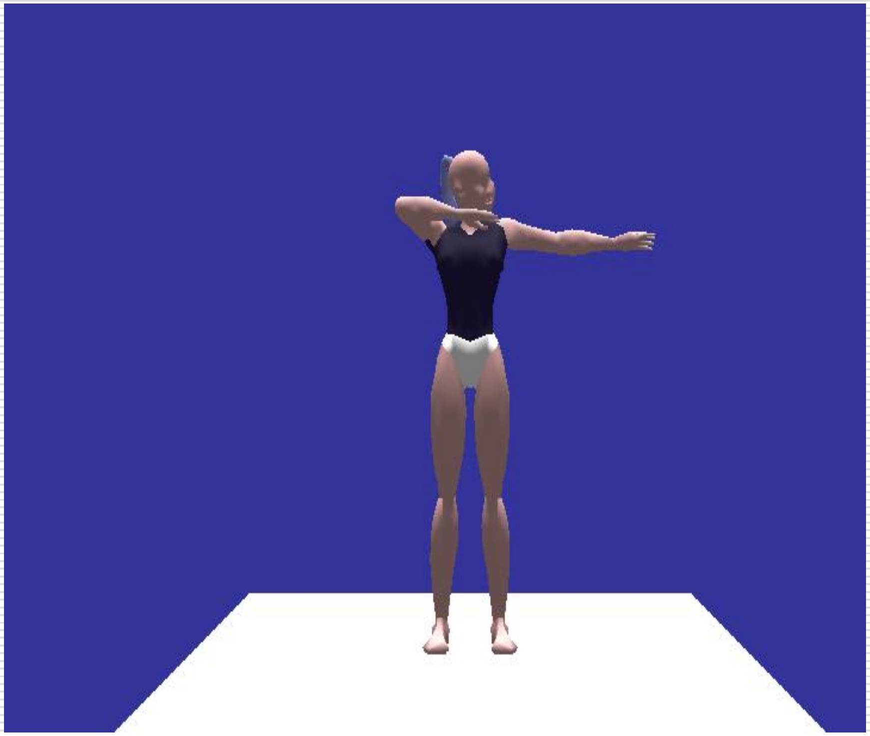
# Experimental Result

- Cally's actions:



# Experimental Result

- Cally's absorbing state actions:



# Experimental Result

- Paladin's actions:

# Experimental Result

- Paladin's absorbing state action:

# Cally Kicks the Spider

# Cally Shots the Lion

# Paladin Shots the Spider

# Paladin Shots the Lion

# Conclusion and Discussion

- We constructed a framework of controlling the motion of 3D human-like avatar by treating it as a Semi-Markov Decision Process.
- We treat the value function as a manifold which depends on the states and actions, and use effective method to improve the Q function in the training process.

# Conclusion and Discussion

- Depends on the shapes of manifold, to research methods to improve the performance of the learning algorithm.
- Research in the area of the 3D models' recognition problem to make virtual human has ability to independently recognize or classify the objects.

# Thank you