國立交通大學

資訊科學與工程研究所

碩士論文

基於邁布里奇攝影集的馬類步態動作生成

Horse Gait Synthesis Based on Muybridge’s Photographs

研 究 生：黃亭潔

指導教授：林文杰　教授

中 華 民 國　一百零一 年 一 月

基於邁布里奇攝影集的馬類步態動作生成
Horse Gait Synthesis Based on Muybridge’s Photographs

研究生：黃亭潔　　Student：Ting-Chieh Huang

指導教授：林文杰　　Advisor：Wen-Chieh Lin

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

January 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年一月

Horse Gait Synthesis Based on Muybridge's Photographs

Student: Ting-Chieh Huang Dr. Wen-Chieh Lin

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

Horse locomotion exhibits rich variations in gaits and styles. Although there have been many approaches proposed for animating quadrupeds, there is not much research on synthesizing horse locomotion. In this thesis, we present a horse locomotion synthesis approach. A user can arbitrarily change a horse's moving speed and direction and our system would automatically adjust the horse's motion to fulfill the user's commands. At preprocessing, we manually capture horse locomotion data from Eadweard Muybridge's famous photographs of animal locomotion, and expand the captured motion database to various speeds for each gait. At realtime, our approach automatically changes gaits based on speed, synthesizes the horse's root trajectory, and adjusts its body orientation based on the horse's turning direction. We propose an asynchronous time warping approach to handle gait transition, which is critical for generating realistic and controllable horse locomotion. Our experiments demonstrate that our system can produce smooth, rich and controllable horse locomotion in real time.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Character animation has been studied in computer graphics for many years. With the rapid development of computer hardware and graphics algorithms, 3D animation techniques are now wildly used in cartoons, video games and digital special effects. In computer animation, animals are a very common character. To generate more realistic animal animation, the data-driven approach, which relies on real motion data as synthesis or editing resources, seems to be a good candidate. Nevertheless, it is not convenient and sometimes even difficult to capture quadruped motion although we are now able to collect a great amount and variety of human motions using commercial motion capture devices.

In this thesis, we propose a synthesis approach to animate quadruped motion based on a small motion database. In particular, we focus on generating horse locomotion as it is basic and essential motion while exhibiting large variations. Moreover, this is also a challenging problem as a horse has six different gaits and changes its gaits at different speeds. If we can solve the gait transition problem of horse, the proposed approach should be applicable to the gait transition problem of other quadrupeds.

We propose a realtime system that allows a user to arbitrarily change a horse's moving speed and direction. Our system automatically changes the horse's gaits according to its speed. We construct a motion database by manually capturing horse postures from Eadweard Muybridge's series pictures of 11 locomotion using MAYA. In order to let a horse walk or run at arbitrary speed, we expand our horse locomotion data by time warping the captured motion to various speeds based on zoological studies [15]. The studies show that a horse's stride length at standing phase increases with its progressing speed, while the contact time of a stride decreases as a power function of its speed.

To be able to generate smooth gait transition between two arbitrary gaits at any phase of a stride, we propose an asynchronous time warping method to handle the gait transition problem. We also develop an on-line root trajectory generation and body orientation adjustment approach to control the global position and orientation of the horse. Instead of using the same generic time to blend the motion of two gaits for all four legs, our asynchronous time warping method performs time warping at each leg separately. We blend each leg's motion of two gaits at its own generic time and gradually adjust each leg's pace to ensure that the four legs would converge to the new gait. Furthermore, our root trajectory generation and body orientation adjustment approach computes the position and orientation based on each foot's ground contact state at each time frame.

The contributions of this thesis are proposing: 1) a real-time horse gait synthesis approach that can automatically change a horse's gait smoothly according to the speed and direction specified by a user; 2) An asynchronous time warping method that can generates smooth gait transition between any two gaits; 3) A root trajectory generation and body orientation adjustment method that preserves horse gait style while the horse turns and/or changes speed.

CHAPTER 2

Related Work

Generating quadrupedal motions receives much research attention in robotics and computer animation. It also benefits from many research studies in zoology and biomechanics. We refer the readers to an excellent survey by Skrba et al. [33] for an overview of quadruped animation techniques including quadruped motion capture and simulation.

People have long been interested in quadruped motions and analyzed them using different motion capture devices. Comparing to human animations, animal locomotion data are very hard to capture. Wild animals like lions, panthers or crocodiles which are not in our circle of life be hard to capture. Kokkevis et al. [18] used a technique called chronophotography which can record many phases of a single moment on the same image. Eadweard Muybridge [23] used a series of 24 cameras to capture animal locomotion including galloping horses, deer, buffaloes, camels, dogs, cats and many other quadrupeds. Different with above traditional methods which can only capture 2D information, Favreau et al. [11] used statistical analysis to analyze video data and make it applicable to the generation of 3D motion. They first segment video into binary images, and then used PCA to find regular motion patterns in these images and serves as an input parameter to continuously and robustly predict 3D motion of animals using radial basis function.

In computer graphics, plenty methods were proposed to simulate quadruped locomotion. There have been many researches using the mix of kinematics and dynamics approach. Girard et al. [12] proposed a generated quadruped animation system PODA. After user using IK (inverse kinematics) or FK (forward kinematics) to set a serial posture of animal movement, the system solve animal's body trajectory by taking into account its vertical motion, horizontal motion and angular motion. Limb trajectories are calculated during body movement to adjust to the variations in foot placement and body speed. Torkos and van de Panne [35] [36] applied trajectory optimizations techniques to let user decide where to place the footprints on the ground, and used a hybrid physically-based and kinematic system to control the body movement. Kokkevis et al. [18] used kinematics, dynamics and control theory to create animal motion. They divided quadruped to body and legs subsystems. Given the desired velocity, the animal heading controller can compute the appropriate aggregate force and torque vector for animal body joints. Then they dynamically control the stance leg and use kinematics to position the legs in the air. Mesh-based animation approaches do not use skeleton bones to generate motion [16][39][41][30][31]. Alternatively, James and Twigg [16] simulate quadruped animation by estimating 'proxy bone' transformations and vertex weights of meshes. In [40] given only the basic shape of a legged animal model, their system can fully automatically synthesize a visually plausible gait without relying on any pre-authored or recorded motions. They used a hybrid optimization which combines a spacetime optimization and a sampling-based derivative-free optimization method. Kry et al. [20] used the natural vibration modes of the body which are related to morphological parameters such as the shape, size, mass, and joint stiffness to generate efficient locomotion. For a model, they computed a set of basis motions and combined these motions to describe different gaits of locomotion of that model. Tsai et al. [37] proposed a physically based method which is able to adjust the original motion to meet adaptation requirements when animating 3D virtual characters. Input a character skeleton and a motion data, user can specify the kinematic constraints and environment constraints. Their system will calculate physically consistent adjustment to fill the adaptation requirements. Karl [32] proposed a novel system creating virtual creatures that move and behave in simulated three-dimensional physical worlds. He used genetic algorithm to evolve the morphologies and the neural systems of those creatures. The neural system is used to control the muscles within the creatures.

The other main procedure of generating quadrupeds locomotion in computer graphics is using the controller system. Controllers system can produce realtime interaction, letting quadrupeds deal with unexpectedly circumstances. In [22] input animation was observed by using video footage to position the horse's legs correctly and then used an animation controller which keeps the physics based horse walking towards a target position. Raibert et al. [27] let user input speed, gait, path and the initial position of the legs and the control system can control animal moving speed and direction. All characters used control algorithms to maintain balance, hopping, speed, posture and elastic energy stored in the legs. The movement of the legs was modeled on a spring system. They treated a bipedal run as two parts of the one-legged hop where the functions for a one-legged hop are applied to each leg in turn. And a quadruped trotting is treated as two parts of the biped, where diagonal legs form pairs. In [21] user can arbitrarily position objects around the scene and the control system could produce satisfactory animations. In real time user can control the speed of motion through their user interface. An animator was allowed to create animations for planar characters as lamps, cats and humans following his intuition about motions. Van de Panne and Michiel [38] used parameters to simulate different motions physically. They found the optimal values of the parameters for different creatures to move with by looking at their moving speed. Using many trials to see which values for limb position and orientation give the quickest and most accurate result for different creatures. Ringrose [29] demonstrated that quadrupeds can be self-stabilizing in transverse and rotary gallops using appropriate body and leg design and circular foot profiles. Coros et al. [10] combined a PD controller system which is defined by parameterized trajectories and an internal virtual force system to simulate an integrated set of gaits and skills for a physics-based quadruped.

In 3D games, users wish to interact with the virtual characters, or even create their own characters. Being able to not preprocess virtual characters becomes a potential concern. Blumberg and Galyean [8] created an interactive environment with virtual characters. Their system is able to propagate high level instructions to the geometry of animal character, which can let the character responds to human commands. Tomlinson and Blumberg [34] used AI technology to bring their wolf character to life. By studying the social interactions between wolfs, and combining hard-coded behaviors with learned behaviors, their wolves are able to store previous interactions in their memory, which can lead to new emergent behaviors. In [14] animators will first specify the semantics of various aspects of the animation, and then their system used the semantic data to record the motion in a generalized form, in realtime, the system would specialize these data onto the different character morphologies, in this way they can generate the individual retargeted motions.

In robotics, balancing and the ability to adapt various terrains is always a major problem. AI and the interaction with humans are also very important now. Boston Dynamics [9] [25] [19] [42] produced ”BigDog” and ”Little Dog”. ”BigDog” has the animal alike legs can absorb the shocks from the ground and is available to recycle energy when it moves from one step to next step. ”Little Dog” are built to study animal locomotion. It is capable of walking in rough terrain which demands careful motion planning. AIBO [13] is a Japanese robot dog which is able to see and moving around while preserving balance. SCOUT [26] used one degree of freedom legs to produce walking, climbing, and galloping animation.

Biomechanics and zoology also provides much important information that should be referred. the major research direction is the influence of animal body sizes on their locomotion gaits. Alexander et al. [2] [5] [3] [4] developed dynamic similarity hypothesis which explains the relationship between animal size, speed, mass and external forces. This hypothesis states that we can describe the movements of two bodies as dynamically similar when one motion appears identical to the other by multiplying all linear dimensions, time intervals and all forces each by some constant factor. They concluded that motions dynamically equivalent have a constant quantity v^2/gh called Froude number. Donald et al. [15] show that horse step length increase with its speed and time of contact decrease as a power function of speed. Some animal behavior strategies [28][7] give us ideas. For instance, dogs usually use front loges to support body and use back legs to speed up [6].

The difference between our method and the above research is we emphasize the transition between gaits. Coros et al. [10] using a linear interpolation method to process gait transition. To achieve a smooth transition, their method blend the gait parameters of the current motion and the goal motion to produce the desire parameters. When transiting from a very slow gait to a very fast gait, their system will sequentially transit to the gaits among the current gait and the goal gait. In our thesis, we propose an asynchronous time warping method to accomplish our gait transition. This method can allow a horse to change its gait arbitrarily between *Walk*, *Amble*, *Trot*, *Rack*, *Canter* or *Gallop*. The transition will begin at the moment user makes the changing speed command, which could be any part of a horse stride. In the nature world, a horse can transit its gait directly from a slow gait to a fast gait. Our method results in a more nature transition for horse locomotion.

CHAPTER 3

Approach

There have been many researches studying quadrupeds animation, but most researches did not focus on the transition between two types of gaits. Due to the characteristics that quadrupeds change their gaits at different speeds, the gait transitions become an important part of their locomotion. Our system uses a data-driven method to obtain the resources we need. Then we parameterize our motion data letting it be able to transit among different gaits. We build a system letting users arbitrarily control horse locomotion. A user can decide horse forward speed and change horse direction in realtime. We deal with the transition between gaits while the horse's speed is changing and plan the root trajectory when the horse turns. In order to build a system of horse locomotion, we need to input horse motion data. Due to the difficulty of getting horse motion capture data, we use Eadweard Muybridge's series pictures of horse locomotion as our input. We also observe horse locomotion video to see how does horse change its step order while its speed change. In this chapter, we introduce our horse locomotion system in details.

Figure 3.1 shows an overview of our approach which is consisting of two phases: **Preprocessing** and **Realtime Motion Synthesis**. At first, we manually capture horse motion from

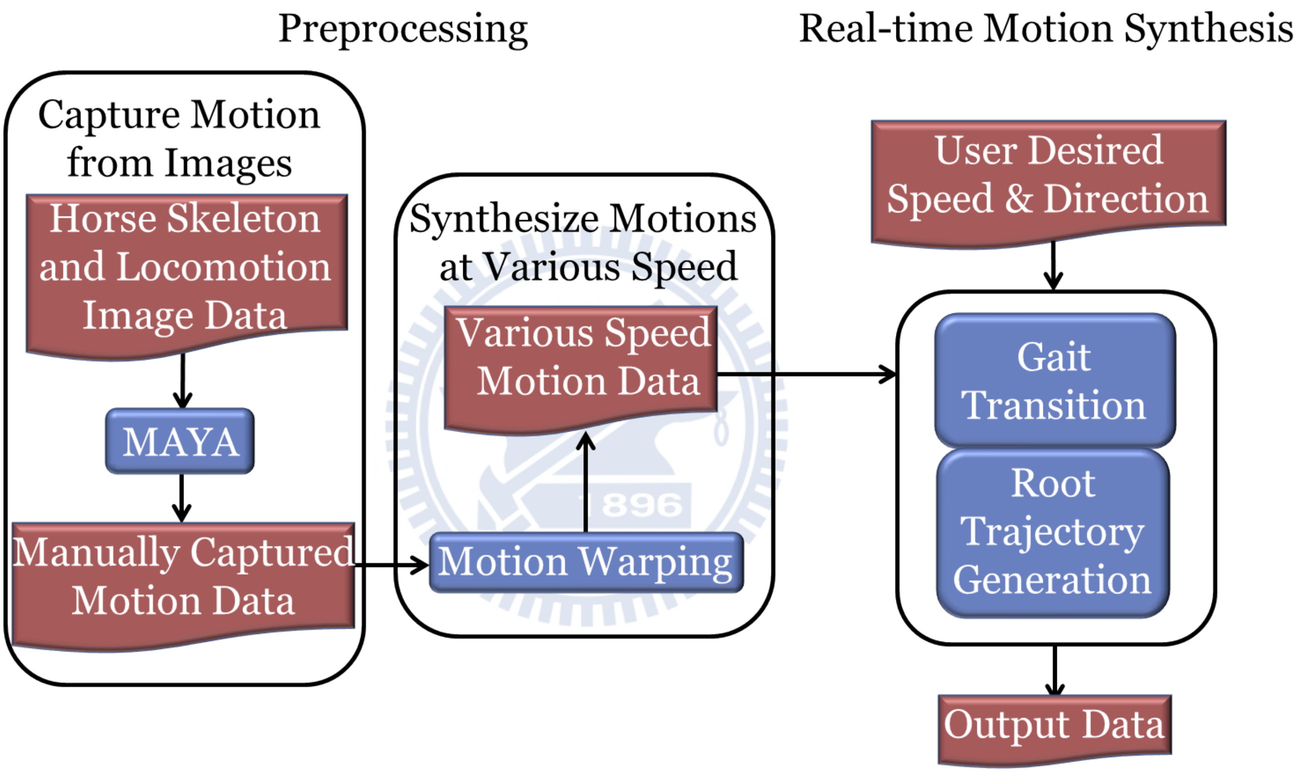

Preprocessing
Real-time Motion Synthesis
Capture Motion from Images
Horse Skeleton and Locomotion Image Data
MAYA
Manually Captured Motion Data
Synthesize Motions at Various Speed
Various Speed Motion Data
Motion Warping
User Desired Speed & Direction
Gait Transition
Root Trajectory Generation
Output Data

Figure 3.1: *The flowchart of our horse locomotion system. The red modules represent data, and the blue modules represent computation.*

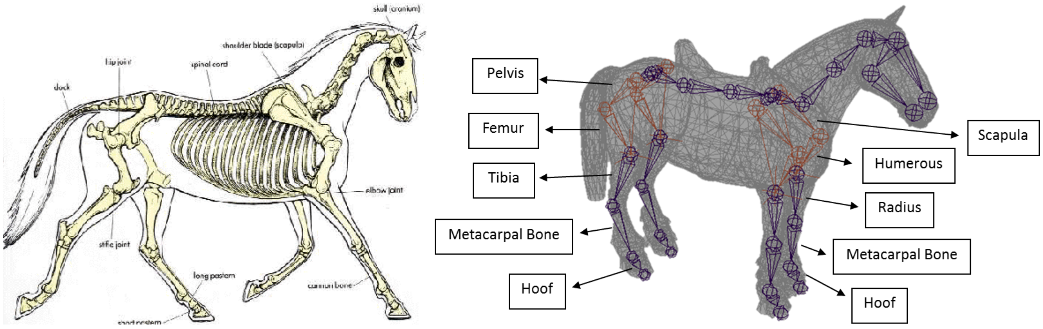
images. With a 3D horse model and a picture of real horse skeleton, we build a 3D horse skeleton in MAYA. Using six types of horse locomotion gaits from Eadweard Muybridge's series pictures, we manually capture horse locomotion data at eleven different speeds from these images. Having the limited horse motion data, we warp these motions to motions at various speeds and ensure the horse step length increases with its speed, and its time of contact is a power function of speed as describe in [15].

As a horse changes its gait when its speed changes rapidly, we need to deal with the gait transition problem. We propose an asynchronous time warping method to transit the horse motion between two different gaits. This method lets the horse four legs warp individually but still preserve the motion consistence. Because our system works as on-line simulation, we cannot estimate the horse root trajectory in advance. In realtime we determine if the horse's legs are on the ground and compare their movements in consecutive frames to plan the root trajectory.

3.1 Capture Motion from Images

Since we do not have horse motion capture data as our input, we need to produce horse motion data for our system. We use Eadweard Muybridge's work [23] to manually capture horse motion data through MAYA. Details are described as follows.

At the beginning, we create a skeleton of horse based on a picture of a real horse skeleton and a horse model. Since our main goal is to handle horse locomotion, we focus on horse limbs. A horse has two forelimbs and two hindlimbs. The major bones of forelimbs are Scapula, Humerous, Radius, Metacarpal Bone, Long Pastern Bone, Short Pastern Bone, and Coffin Bone. Hindlimbs's major bones are Pelvis, Femur, Tibia, Metatarsal Bone, Long Pastern Bone, Short Pastern Bone, and Coffin Bone. On both forelimbs and hindlimbs we combine Long Pastern Bone, Short Pastern Bone, and Coffin Bone as Hoof. Figure 3.2 demonstrates the 2D skeleton image and our 3D horse skeleton. In our horse skeleton, we use eight bones from head to root to represent the horse **Body**. One bone as **head**, three bones as **neck** and four bones as **spine**.



(a) Real horse skeleton 2D image (b) Building 3D horse Skeleton

Figure 3.2: *Create a 3D horse skeleton in MAYA*

In Eadweard Muybridge's work, a stride consists a series of movement pictures. Figure 3.3 shows one of Muybridge's photographs. Each picture was taken at the same interval of time. We pose the skeleton to match the posture in each picture of a stride to capture a key posture for our motion. Figure 3.4 shows the process we use to capture a horse's motion. After capturing every posture in a stride, we can generate a smooth horse locomotion data through the keyframing function in MAYA. We warp the time of the duration of a stride in our system to match the correct speed measured in Muybridge's photographs.

From [23], we know that *Walk* is the slowest gait for a horse. It is characterized by an immutable sequence of limb movements. When a horse commences to walk from still standing, it always starts with a hindlimb and then follows by the lateral forelimb. Next comes with the hindlimb of the other side and ends with the diagonal forelimb. In the normal walk of a horse, it will always be found that the legs which support the body during one stride are thrown twice on the laterals, twice on the diagonals, twice on two forelimbs and one hindlimb, and twice on two hindlimbs and one forelimb. *Amble* is a development of *Walk*. It is an accelerated *Walk* since it has the same sequence of foot impacts in higher rate of speed. The difference between *Walk* and *Amble* is that a hindlimb and a forelimb are alternately lifted from the ground in advance of its following leg stepping on the ground in *Amble*. This results in throwing the duty of support alternately on one leg and on two legs. A forelimb and a hindlimb alternately becomes single

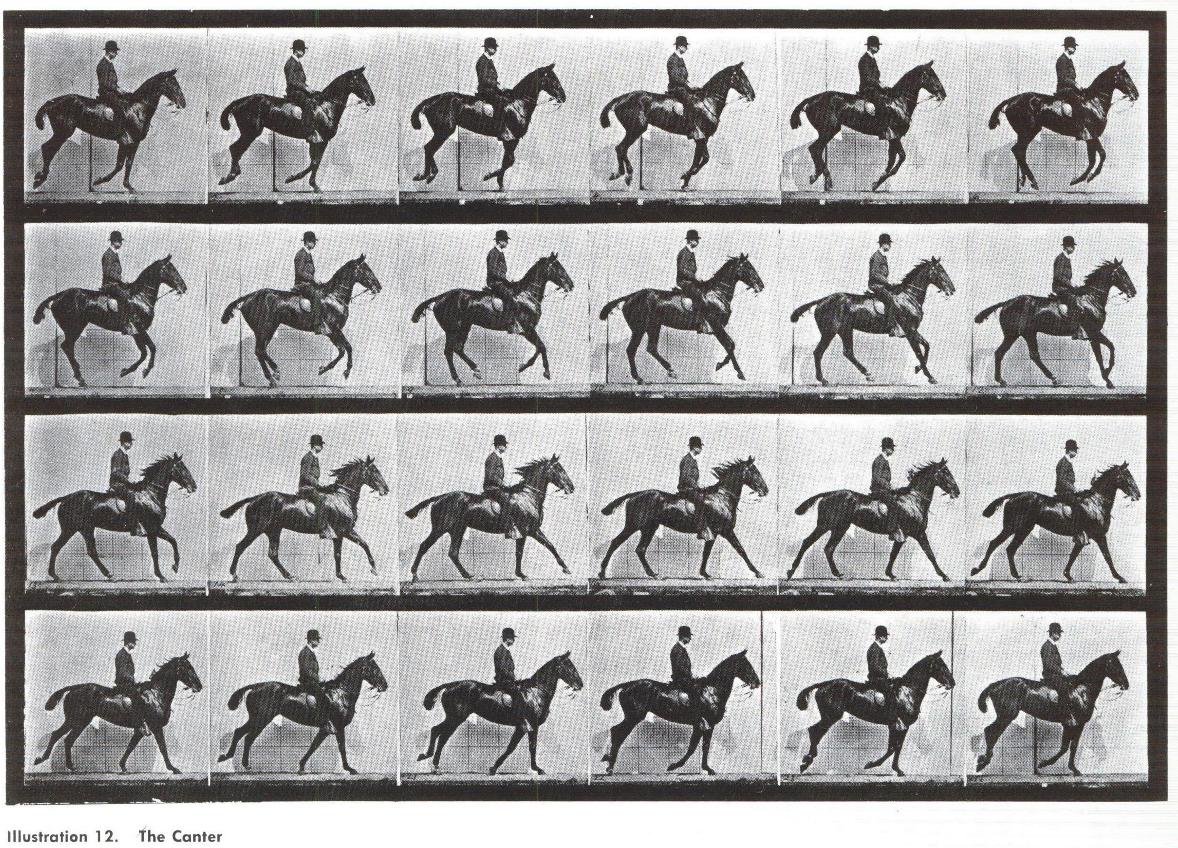


Figure 3.3: *Eadweard Muybridge's horse locomotion picture*

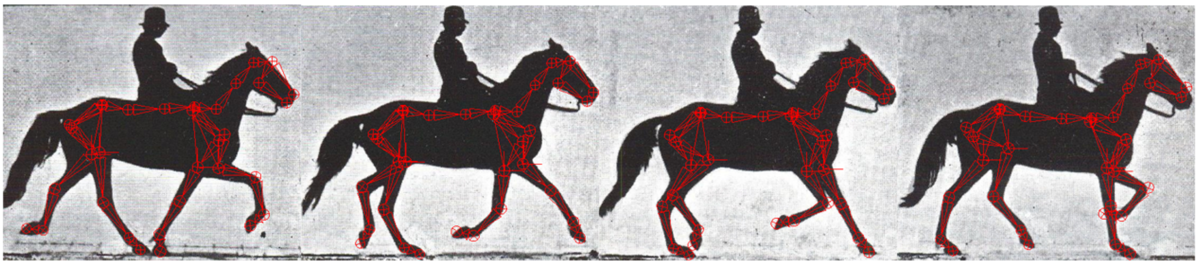


Figure 3.4: *Aligning skeleton with the image*

support, and the diagonal limbs and lateral limbs supplying duplex support by turns. *Trot* and *Rack* are similar gaits using different pairs of a horse limbs. *Trot* is a system of progress in which each pair of diagonal legs are alternately lifted, thrust forward, and placed on the ground. In contrast, each pair of lateral legs are alternately used in *Rack*. The body of the horse will make a transit, without support, twice during each stride in both *Trot* and *Rack*. A horse can trot in a wide range of speed. In horse racing it can be trained even faster. *Rack* is a little bit faster than *Trot*, but it is a very rare gait for a horse. *Canter* and *Gallop* are very different from the gaits above. Instead of having the same regularity of intervals in a stride, they use one spring leg to bounce to the air. We should consider each leg as an unit in these two gaits. Although *Canter* has the same sequence of foot-fallings as *Walk*, besides the irregular stepping phase, it has an unsupported transit during a portion of the stride. The spring is invariably taken from a forelimb, while the landing is effected on the diagonal hindlimb. *Canter* can be considered as a variation of *Gallop*. It is faster than the average *Trot*, but slower than *Gallop*. *Gallop* is the fastest gait of a horse. There are two systems of galloping, transverse-gallop and rotatory-gallop. Assuming a horse starting its gallop with left hindlimb, then the sequence of its foot step in transverse-gallop would be left hindlimb, right hindlimb, left forelimb and finally right forelimb. In contrast, rotatory-gallop uses left hindlimb, right hindlimb, right forelimb and left forelimb in order. In most cases, a horse uses transverse-gallop. Figure 3.5 shows these six gaits we use in our system. On the left side, it shows the time duration of each leg in a stride. The dark blue represents the standing phase, while the light blue represents a leg in the air. On the right side, it shows the sequence of foot impacts the ground. The arrow points to the heading direction, the two triangles stand for forelimbs, and the two circles stand for hindlimbs. From left to right the arrows reveal a stride of one gait.

According to Eadweard Muybridge's photographs and the zoological studies, we let our horse use specific gaits at different speeds. Figure 3.6 shows the principle we use to decide our horse progressing speed, the orange blocks are the data we captured from the images. The horse ***Walks*** when its speed is one to five miles per hour. The horse uses ***Amble*** gait at the speed of six to nine miles per hour; ***Trot*** and ***Rack*** at the speed of ten to twenty miles per hour; ***Canter*** at the

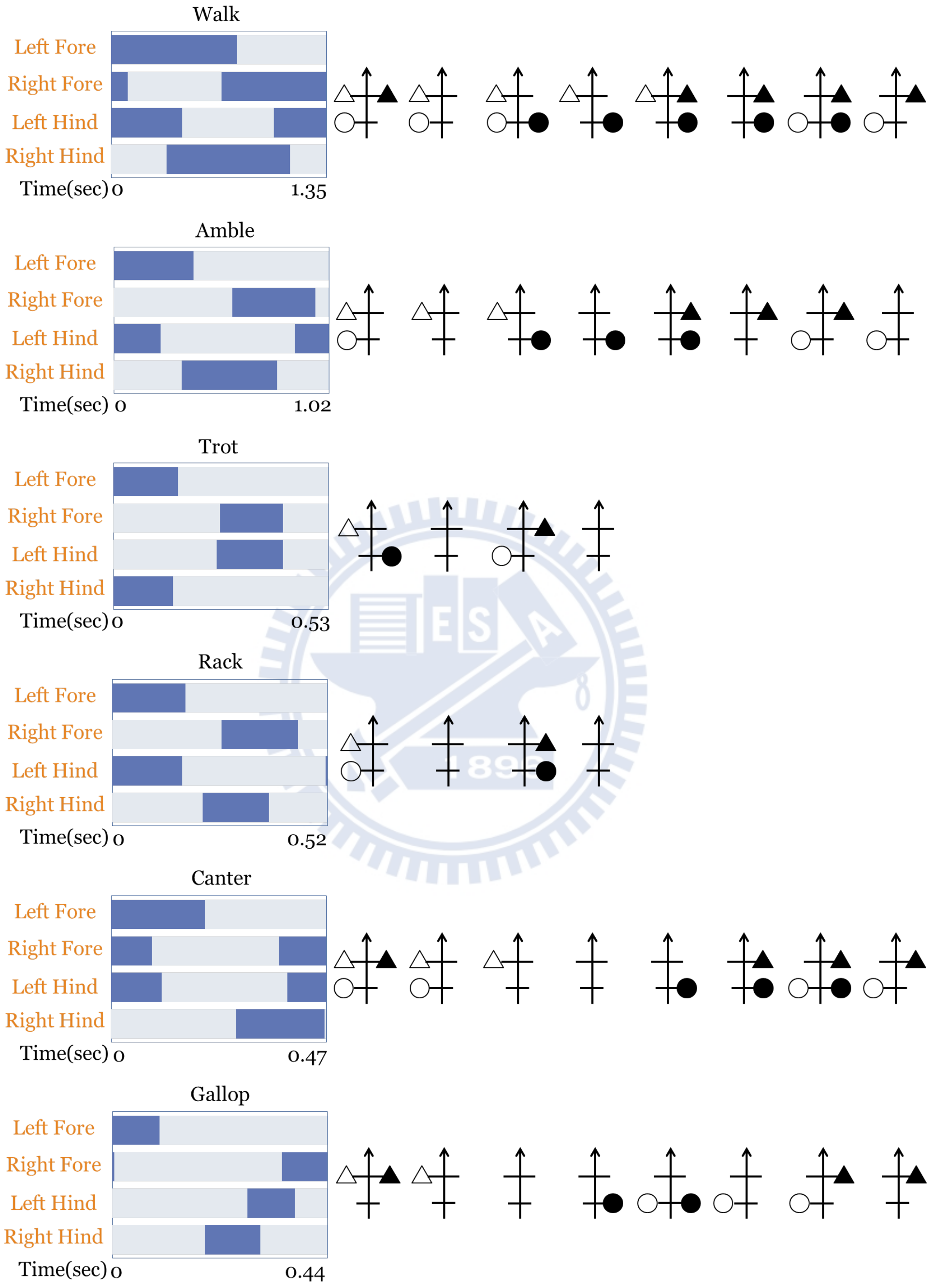


Figure 3.5: *The left side represents the time duration of the feet at different gaits. The right side shows the support afforded by the feet of a horse during a normal stride at different gaits.*

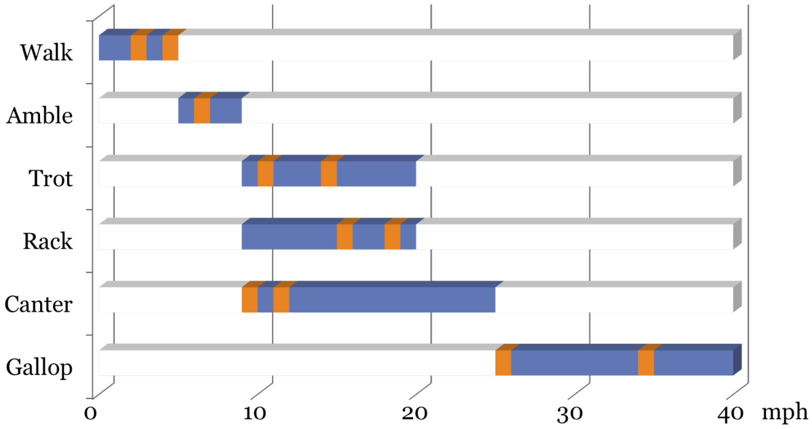


Figure 3.6: *Speed Decision*

speed of ten to twenty five miles per hour; ***Gallop*** at the speed of twenty six to forty miles per hour. When a user slightly changes speed at the range of ten to twenty miles per hour, system will not change the current gait the horse is using. Otherwise, system will randomly pick a gait between ***Trot***, ***Rack***, and ***Canter*** for the horse to switch to.

We first describe the joints and the notation used in this thesis in Figure 3.7 and Table 3.1. ***Hip/Shoulder Position*** is the position of a horse's shoulder or hip. ***Foot Position*** is the position of the end of the leg. ***Fetlock Joint Angle*** represents the angle between the horse's metatarsal bone and its hoof. ***Pastern*** is a part of the leg of a horse between the ***Fetlock Joint Angle*** and the top of the hoof. We set ***Root*** at the end of the horse's back. A serial bones from ***Head*** to ***Root*** is ***Body***. We define a pose as $Q = \{p^r, q^1, q^2, q^3, q^4, q^b\}$, where p^r is the global position of the root joint, q^1, q^2, q^3 and q^4 are the joint angles of right-fore leg, left-fore leg, right-back leg and left-back leg, respectively, and q^b are joint angles of **Body**.

3.2 Synthesize Motions at Various Speed

We let users arbitrarily control a horse's speed and direction. Since we only have limited motion data captured from Muybridge's images, we need to synthesize motions at different speeds for our system.

Table 3.1: Notation

D	The distance the body travels during the stance phase.
T_c	The duration of horse stance phase of a stride.
v	Speed.
G	The height of ground.
M_1	The original gait motion before transition.
M_2	The target gait motion to transit to.
L_1	The total frame numbers in M_1 for one stride.
L_2	The total frame numbers in M_2 for one stride.
Δt_1	The average incremental generic time of M_1.
Δt_2	The average incremental generic time of M_2.
B	The amount of blending frames.
T_1^j	The actual time of leg j at in M_1.
T_2^j	The actual time of leg j at in M_2.
t_1^j	The generic time of leg j at in M_1.
t_2^j	The generic time of leg j at in M_2.
w_1	The weight for M_1.
w_2	The weight for M_2.
K^j	The position where leg j strikes the ground.
N	The number of legs standing on the ground.

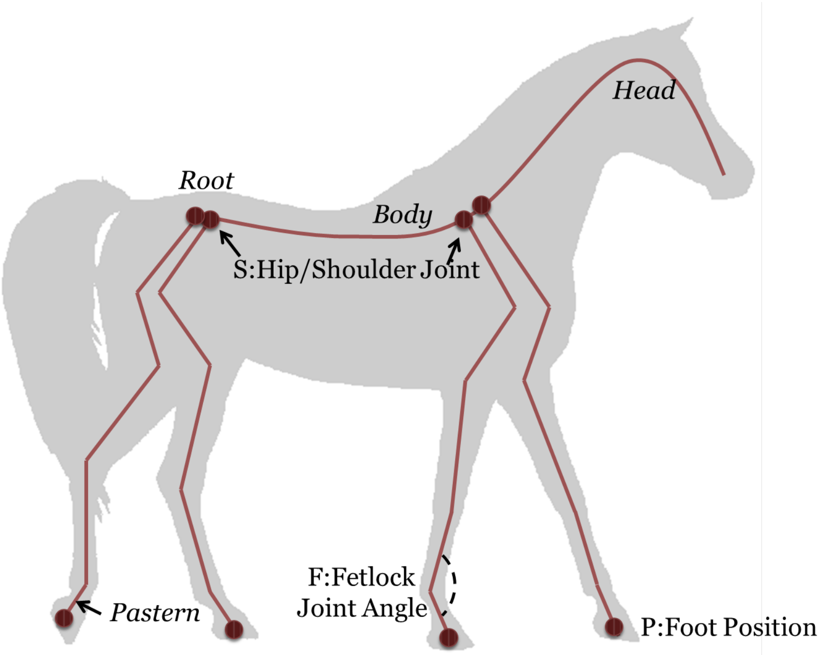


Figure 3.7: Joints notation

Figure 3.8 shows the flow chart of we constructing motions at various speeds. First we use Figure 3.6 to decide the speed range for every gait. Then we calculate the step length of those speeds at which we lack of motions. We adjust the horse step size in our captured motion data to conform to the step length we generated. Having the new step length, we can calculate the horse's stride length. Stride length is the distance a horse travels during a stride, including the step length and the flying distance. Using the stride length, we warp our original captured motion to the proper velocity.

Our method mainly considers the relationship between horse speed and time of contact in [15]. Time of contact T_c is the duration of the horse stance phase of a stride. Hoyt et al. [15] show that time of contact decreases as a power function of speed, and horse's step length increases with its speed. Step length D, which is the product of speed v and time of contact T_c, is the distance that the body travels during the stance phase. D increases with v. From the above descriptions, we can get Equations 3.1 and 3.2. a and b are two parameters. b is a negative number and significantly different from -1.0. In order to get the step length of different speed,

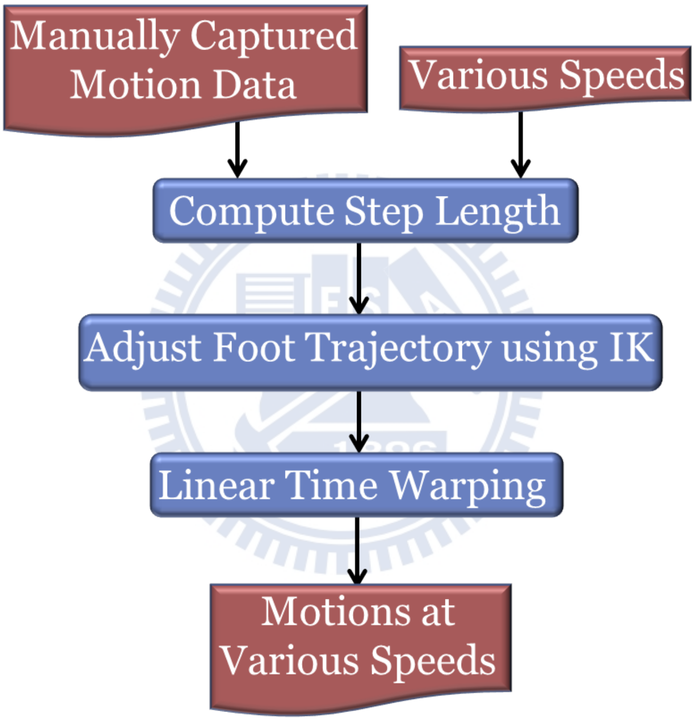

Manually Captured Motion Data
Various Speeds
Compute Step Length
Adjust Foot Trajectory using IK
Linear Time Warping
Motions at Various Speeds

Figure 3.8: *Synthesize Motions at Various Speed*

we need to know the time of contact of those speed. So we try to find the parameters a and b of each gait.

$$T_c = a \cdot v^b, \tag{3.1}$$

$$D = v \cdot T_c, \tag{3.2}$$

Figure 3.9 shows the two sets of data we use. With these information, we can find parameters a and b for each gait. We also demonstrate the parameters we found in Figure 3.9.

From Eadweard Muybridge's work we only have one suitable motion data for the gait ***Amble***. Gait ***Amble*** and ***Walk*** have the same step order. ***Amble*** is the faster ***Walk***. We use the parameters a and b of gait ***Walk*** to estimate the parameters a and b of gait ***Amble***. We first use the parameters a and b of gait ***Walk*** to calculate the T_c at the speeds 7 and 9 mph. Then we compare the T_c of speed 7 mph we calculated with the T_c of gait ***Amble*** we captured from the Muybridge's images. And we shift the T_c of speeds 7 and 9 mph to let the T_c of speed 7 mph align with the T_c of gait ***Amble***. Then we use the new T_c of speeds 7 and 9 mph to compute the parameters a and b of gait ***Amble***. Figure 3.10 explains this process. The brown diamond is the data we obtained from the image.

After we get the a and b parameters for each gait, we can calculate the T_c for all speeds using Equation 3.1. We also get the D of all speeds in every gait through Equation 3.2. We preserve the characteristics that T_c decreases as a power function of v, and D increases with v. Figure 3.11 shows the trend of step length and time of contact at different speeds in our system. The yellow blocks are the data we obtained from Muybridge's photographs. We can find that step length increases with speed, and time of contact decreases as a power function of speed.

We use the computed step length to modify the captured motion in order to get the new motions at various speeds. At the initial state our horse heading the global X direction, and the global Y direction is pointed to the sky. At each gait motion we adjust the horse foot position P at *XY* plane. Considering the position of ***Hip/Shoulder Position*** S as the origin, and the

	Speed(mph)	T_c (sec)	a	b
Walk 1	3	1.36865	1.688057	-0.71476
Walk 2	5	0.95		
Amble 1	7	1.020508	1.826353	-0.51021
Amble 2	×	×		
Trot 1	15	0.339225	1.407662	-0.74782
Trot 2	11	0.427778		
Rack 1	16	0.404915	2.386013	-0.90153
Rack 2	19	0.3468		
Canter 1	12	0.399537	2.005463	-0.96045
Canter 2	10	0.476		
Gallop 1	35	0.351412	1.14589	-0.42978
Gallop 2	26	0.3993		

Figure 3.9: *Two sets of motions and parameters a and b*

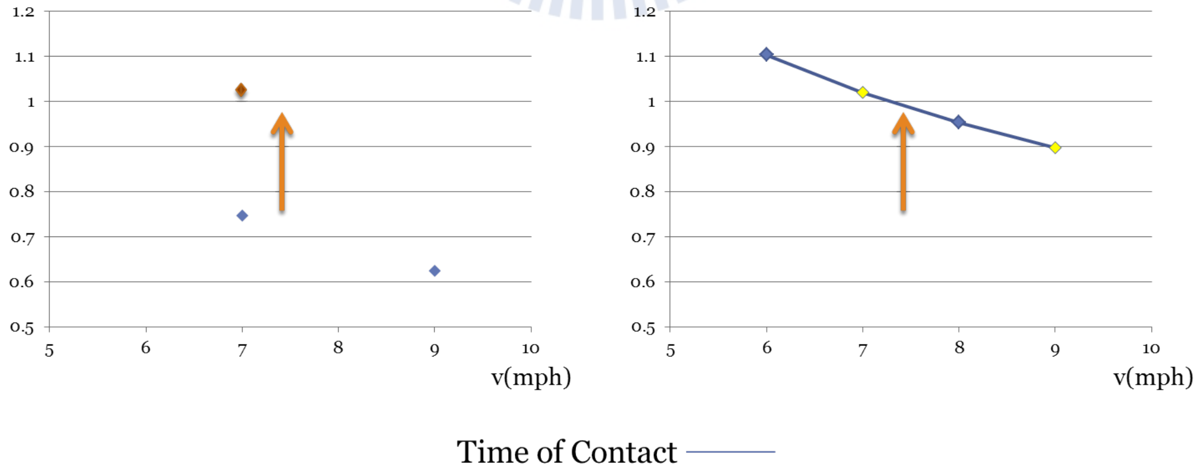


Figure 3.10: *To estimate the parameters a and b*

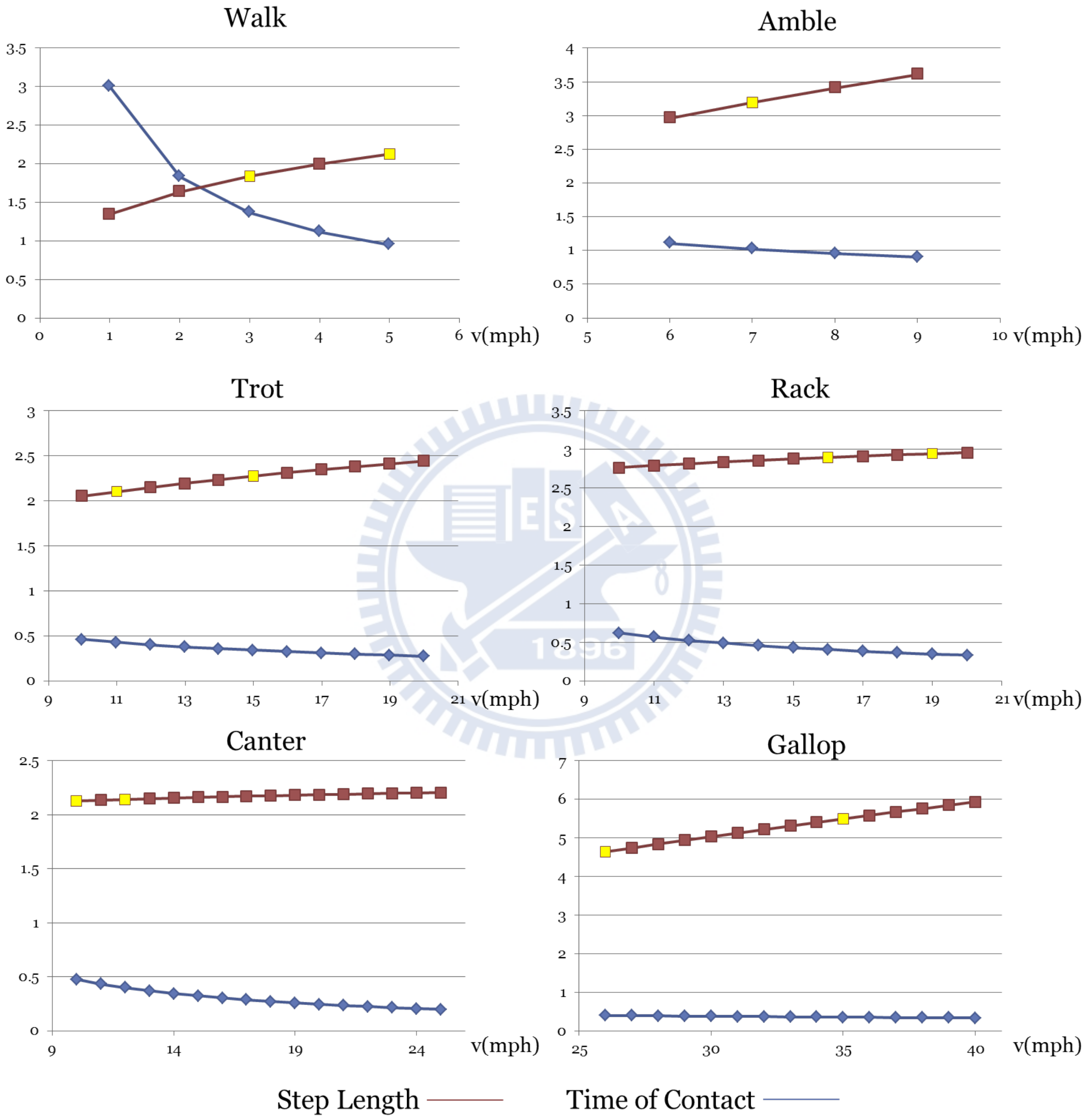
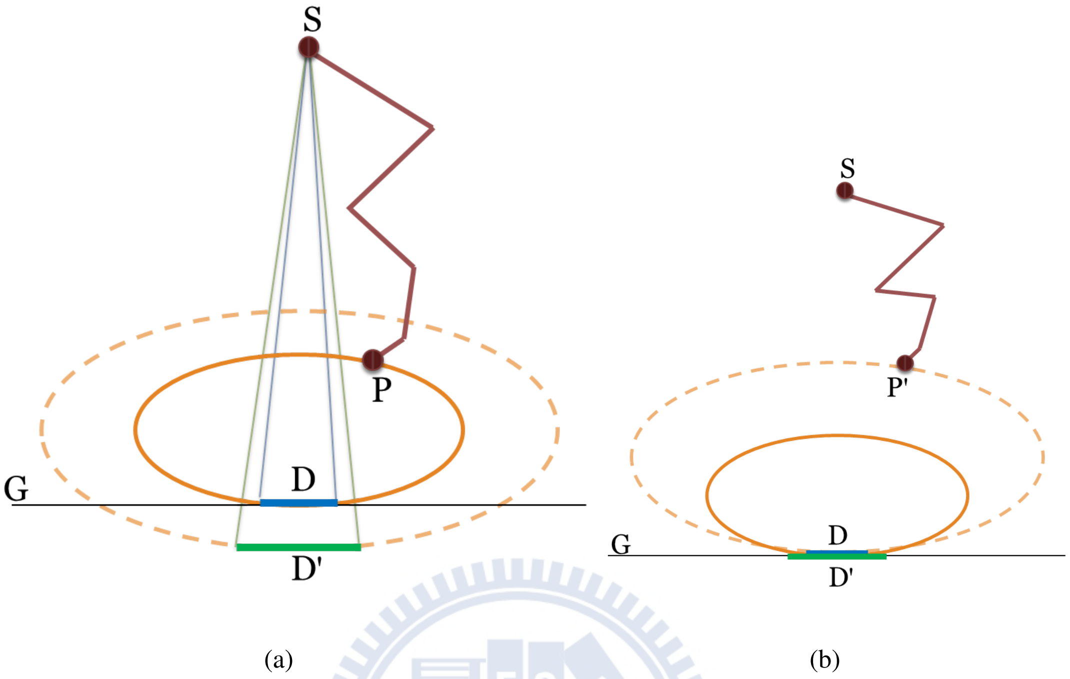


Figure 3.11: *Time of Contact and Step Length*



(a) (b)

Figure 3.12: *Adjust foot position trajectory*

step lengths D and D' as one side of a triangle. We use a similar triangle method to generate the target foot position trajectory of the new stride. Figure 3.12 (a) demonstrates the similar triangle method we use. We adjust the x component of P by Equation 3.3, and the y component of P by Equation 3.4. P' is the new foot position and G is an offset ensuring that the stance leg would not penetrate or leave the ground. Figure 3.12 (b) shows the adjustment. We use Jacobian-Inverse IK method to recompute q^1, q^2, q^3 and q^4 to alter the horse four legs foot positions from P to P'.

$$P'_x = (P_x - S_x) \times \frac{D'}{D} + S_x, \tag{3.3}$$

$$P'_y = (P_y - S_y) \times \frac{D'}{D} + S_y + G, \tag{3.4}$$

After we adjust the foot positions of all the frame of the new motion, we also have to warp the motion to conform to its speed. Using the stride length of the new motion and the frame rate we use in the system, we reset the total frame number of the new motion and warp the original motion to fit in with it.

3.3 Gait Transition

Our system uses limited horse motion data to build a realtime system which generates horse locomotion at different speeds and directions. When a horse changes its speed rapidly, it will change its gait. It is important to preserve the smoothness at gait transitions.

It is needed to set the keytimes such as left foot heel-strikes, left foot toe-offs, right foot heel-strikes, and right foot toe-offs, in the conventional time warping method. Besides, to blend the various speed motions it has to guarantee that all motions consist of the same sequence of keytime phases.

A horse changes its gaits at different speeds. In different gaits, a horse has different step orders. If we set keyframes as left forelimb heel-strikes, left forelimb toe-offs, right forelimb heel-strikes, right forelimb toe-offs, left hindlimb heel-strikes, left hindlimb toe-offs, right hindlimb heel-strikes, and right hindlimb toe-offs, we will not find the same sequence of keytime phases in different gait motion. With four legs, we cannot blend these legs at the same time. Nevertheless, if we blend each foot individually, it will not converge to the same posture in target motion. Therefore, we propose an asynchronous time warping method to deal with our horse gait transitions.

In this section we will show the result of blending each foot individually by conventional time warping method. Then we will illustrate our asynchronous time warping method.

3.3.1 Synchronous Time Warping

We borrow the notations in [24]. We denote the actual keytime by $K_i, 1 \leq i \leq N_k$, where N_k is the number of the keytimes. Based on the keytimes, we can defined time warping as a piecewise linear mapping of actual time $T \in [K_1, K_{N_k}]$ onto generic time $t \in [0, 1]$. The relationship between actual time T and generic time t are the Equations 3.5-3.8.

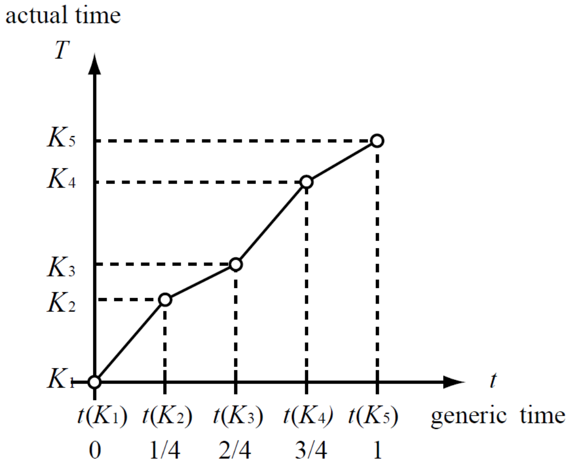


Figure 3.13: *Time Warping*

$$t\left(K_m\right)=\frac{m-1}{N_k-1}, \tag{3.5}$$

$$t\left(T\right)=\left(m-1+\frac{T-K_m}{K_{m+1}-K_m}\right)\times\frac{1}{N_k-1}, \tag{3.6}$$

$$T\left(t\right)=K_m+\left[\left(N_k-1\right)\times t-\left(m-1\right)\right]\times\left(K_{m+1}-K_m\right), \tag{3.7}$$

$$T\left(t\right)=\sum_{i=1}^{N_e}w_iT_i\left(t\right), \tag{3.8}$$

Where w_i and $T_i\left(t\right)$ are the weight value and the actual time of the ith example at the generic time t, respectively. N_e represent the number of the motions. Figure 3.13 from [24] illustrates the piecewise linear mapping of actual time T onto generic time t.

We tried to use Equation 3.8 and blended four legs independently. Using the same generic time t for each leg at every time frame let each leg has separate actual time T. And four actual time T will not converge at the end of the blending. We cannot find a posture at target motion for our horse to transit to. So we find the synchronous time warping method is not working. We use asynchronous time warping method to process the gait transition.

3.3.2 Asynchronous Time Warping

Using synchronous time warping method and individually blending each leg cause the four legs not able to converge on the target motion. We observed horse gait transitions in horse locomotion video, and we found that a horse need not transits their legs in the same rate. When transiting from the original gait motion to the target gait motion, some feet will speedily transit to target motion and the other feet will speed up to catch up with them. So we do not need to use the same generic time t for all four legs at every time frame. We can let some feet blend faster and some feet blend slower.

The basic idea of our asynchronous time warping method is that we use an actual time frame T of the original motion M_1, then we map the actual time to its four different generic time t of each leg. When blending, every time frame we increase a small amount of generic time which is difference for each leg. With the new generic time we can map to each leg's actual time of the original motion M_1 and target motion M_2. We blend each leg independently. We choose the adding generic time carefully and make sure that four legs will converge in the target motion M_2. If the user changes the horse progressing speed so rapidly that the horse needs to change its gaits, we start our gait transition process.

At the time the user decides a new speed, we save the current actual time frame T_1 of the original motion. Then we use T_1 to check four leg's generic time $t_1^j, 1 \leq j \leq 4$. During blending, the incremental of the generic time per frame smoothly changes from Δt_1 to Δt_2. Δt_1 and Δt_2 represent the average incremental generic time per actual time frame in M_1 and M_2. We assign a leg as the reference leg and at every time frame letting its generic time increases Δt^*. This is shown in Equation 3.11. The weights w_1(i) and w_2(i) control the proportion of M_1 and M_2 during the blending process. $w_2(i) = \frac{i}{B}$, $w_1(i) = 1 - w_2(i)$ where i denotes the ith frame in the transition.

$$\Delta t_1 = \frac{1}{L_1}, \tag{3.9}$$

$$\Delta t_2 = \frac{1}{L_2}, \tag{3.10}$$

$$\Delta t^* \left(i\right) = w_1 \left(i\right) \times \Delta t_1 + w_2 \left(i\right) \times \Delta t_2, \tag{3.11}$$

σ^* is the length of transition in generic time by summing up the incremental in each frame. Since every actual time frame is mapping to four different generic time for each leg. We observed from the video and Muybridge's photographs that a horse changes its gaits within two to three stride cycles. Therefore, we set the total blending generic time as the length of two and a half strides. We precompute the length of total blending generic time for the reference leg. Equation 3.14 demonstrates that. B is the number of actual time frames of the blending process.

$$\sum_{i=1}^{B} \left(\left(1 - \frac{i}{B} \right) \times \Delta t_1 + \frac{i}{B} \times \Delta t_2 \right) \cong 2.5, \tag{3.12}$$

$$B = \frac{5 + \Delta t_1 - \Delta t_2}{\Delta t_1 + \Delta t_2}, \tag{3.13}$$

$$\sigma^* = \sum_{i=1}^{B} \left(\left(1 - \frac{i}{B} \right) \times \Delta t_1 + \frac{i}{B} \times \Delta t_2 \right), \tag{3.14}$$

With σ^* we can find the ending posture $M_2(t_2^*(0) + \sigma^*)$ of our blending process in M_2. Where $t_2^*(0)$ is the generic time of the reference leg when the transition begins. We use the actual time of the ending posture $M_2(t_2^*(0) + \sigma^*)$ to calculate the incremental generic time Δt^j for the other legs besides the reference leg.

At first, we decide the number of stride cycles every leg going to blend. Using $t_1^j(i)$ of the original motion at the start posture of blending, and $t_2^j(i)$ of the target motion at the end posture of blending, we calculate $\sigma^j, j \neq *$ by finding a number of stride cycles which let σ^j be closest to the σ^*.

$$x' = \underset{x}{\operatorname{argmin}} \left| \left[\left(t_2^j(i) + x \right) - t_1^j(i) \right] - \sigma^* \right|, x = 2 \; or \; 3 \tag{3.15}$$

$$\sigma^j = \left(t_2^j(i) + x'\right) - t_1^j(i), \tag{3.16}$$

At every time frame, the incremental generic time for each leg besides the reference leg is Δt^j. We let the horse four legs blend with distinctive generic time.

$$\Delta t^j\,(i) = \Delta t\,(i)^* \times \frac{\sigma^j}{\sigma^*}, \tag{3.17}$$

We first compute the blended posture of legs at the ith frame of the transition,

$$q^j(i) = w_1(i) \cdot q_1^j(t_1^j(i)) + w_2(i) \cdot q_2^j(t_2^j(i)), \tag{3.18}$$

where q_1^j and q_2^j are the postures of the jth leg in the current gait and the new gait. The generic time $t_1^j(i)$ and $t_2^j(i)$ are computed as follows

$$t_1^j(i) = t_1^j(0) + \sum_{f=1}^{i} \Delta t^j(f) \tag{3.19}$$

$$t_2^j(i) = t_2^j(0) + \sum_{f=1}^{i} \Delta t^j(f), \tag{3.20}$$

where $t_1^j(0)$ and $t_2^j(0)$ are the generic time of the jth leg in the current and new gaits when the transition begins.

The **Body** postures q^b of M_B can be obtained by simply blending the body postures corresponding to the generic time of four legs. We use slerp to blend the **Body** postures by

$$q^b(i) = \frac{1}{4}[w_1(i) \sum_{j=1}^{4} q_1^b(t_1^j(i)) + w_2(i) \sum_{j=1}^{4} q_2^b(t_2^j(i))],$$

where q_1^b and q_2^b are the body posture of M_1 and M_2, respectively. The translation of the root p^r will be explained in the following section.

3.4 Root Trajectory Generation

Our system works in on-line computation. In order to let the user change the horse's speed and direction instantaneously, we cannot preplan root trajectory. At every time frame, we need to calculate the translation and orientation of the root joint. Since we get our motions from images, and we try to preserve the reality of the original motion, we do not choose to use path editing to control the direction of the horse. We consider foot striking position before we translate the horse root.

Figure 3.14 shows the flowchart of the way we plan the root trajectory. In realtime, we determine the foot phase of the posture every time frame. We estimate which leg is on the ground and which leg is in the air. Legs on the ground are in charge of root translation. Legs in the air have no influences to root translation. If the user changes the horse's direction, we have to rotate the horse's body and recompute its foot prints. Finally, we need to calculate foot's movement to obtain the translation of root.

3.4.1 Determine Foot Phase

In realtime, our system deals with a horse posture every time frame. Owing to plan root trajectory, we need to determine the horse leg's foot phase of the posture. It is important to know which leg is on the ground and which leg is in the air. The horse root translation is based on the amount of movement of its legs which are on the ground. The movement of the legs in the air have no influence to root translation. So we need to distinguish the legs on the ground from a horse posture. We use some comparative method to determine the foot phase for each leg.

We use three parts of a horse to decide if a leg is on the ground. The notations we use are shown in Figure 3.7. **Pastern** is the bottom part of a horse leg. It plays an important role of being a spring or a cushion to lessen the concussion from the ground. When a horse foot strikes on the ground, to lighten the force, **Pastern** commences to bend. As the leg becomes vertical, depending on the concussion, **Pastern** could becomes horizontal. While the leg gradually away

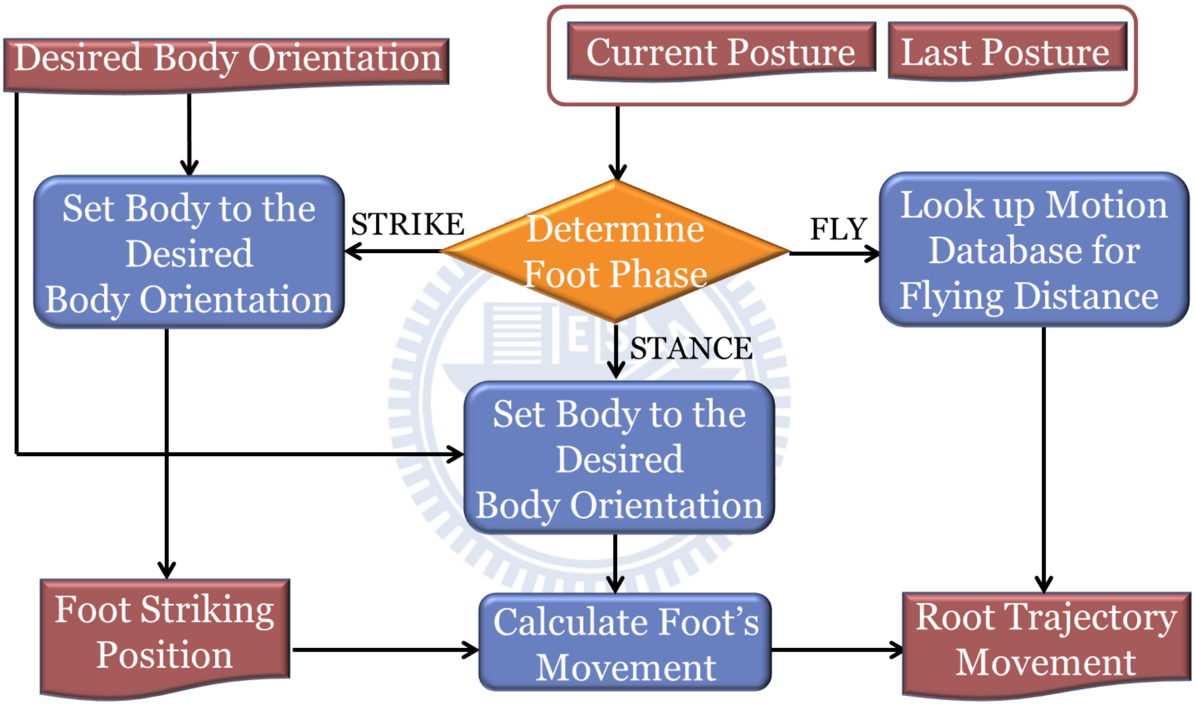

Desired Body Orientation
Current Posture
Last Posture
Set Body to the Desired Body Orientation
STRIKE
Determine Foot Phase
FLY
Look up Motion Database for Flying Distance
STANCE
Set Body to the Desired Body Orientation
Foot Striking Position
Calculate Foot's Movement
Root Trajectory Movement

Figure 3.14: *The flowchart of our root trajectory planing system. The red modules represent data, the blue modules represent computation and the orange modules represent decisions*

from the ground, **Pastern** bends from horizontal to vertical then bends over to another direction. After the leg leaves the ground, **Pastern** keeps bending in the air, and then progressively stretches to the front.

Because of the function and characteristics of **Pastern**, we pay attention to the angle of ***Fetlock Joint Angle*** which is connected to it. When a leg's ***Foot Position*** is in the front of its ***Hip/Shoulder Position***, we can tell that this leg is about to land on the ground or is just landing on the ground. In this situation, we check this leg's ***Fetlock Joint Angle*** to see the bending degree of pastern. If pastern bends beyond a threshold, we mark this leg as on the ground. We analyzed Muybridge's photographs to set the threshold at $\{158, 152, 165, 169\}$ degrees for the **pastern** in the left forelimb, right forelimb, left hindlimb and right hindlimb. On the other hand, when a leg's ***Foot Position*** is in the back of its ***Hip/Shoulder Position***, it is preparing to leave the ground or just leaving the ground. Again, we check the leg's ***Fetlock Joint Angle***. If **Pastern** bends over too much cross a threshold we set, we determine this leg already leaves the ground and is in the air. The threshold we obtained from Muybridge's photographs is $\{181, 181, 180, 180\}$ degrees. We add another two constraints to assure our comparative method got the correctest result. We compare the height distance between a leg's ***Hip/Shoulder Position*** and ***Foot Position***. When their height are too close, which means that this leg must be in the air. In our system, when we build a motion we also mark the frame which four legs strike on the ground. And we save the striking position of the foot step of each leg. So in realtime, when determining if a leg has landed on the ground, we also compare its ***Foot Position*** with the striking position we saved to make sure the ***Foot Position*** is stepping behind the striking position.

3.4.2 Adjust Root Movement and Body Orientation

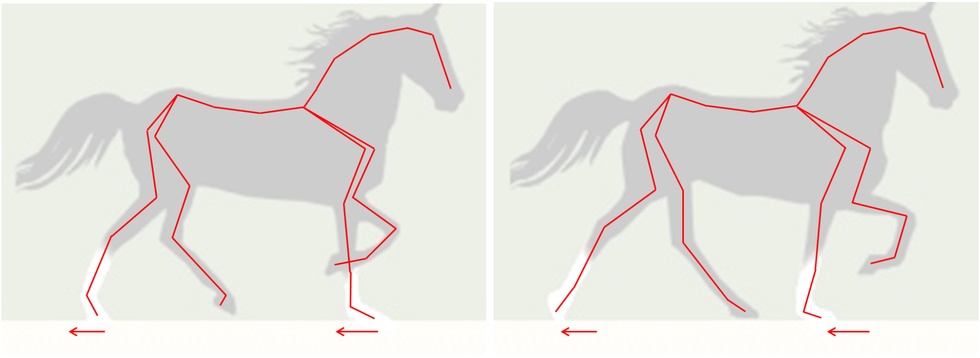
In this section we will explain the method we use to handle the horse's orientation when the user changes the horse's direction. We will also expand on the way we calculate root translation using the movement of the legs which are standing on the ground. When the horse moving forward without changing direction, we only need to deal with root translation and we do not need to alter the horse posture; However, when a user changes the horse's direction, we need to rotate the horse body to follow the new direction. The user can turn the horse's direction to left or right with 15 degrees at a time. When the user specifies a turning degree to the horse, the horse **Body** begins to turn a quantity degrees every time frame from its **head** joint, followed by its **neck** joints, finally to its **spine** joints. The turning degree per frame is decided by the horse's speed. When the horse moves at slow speed, we let it turns 0.01 degrees every time frame. In contrast, when the horse moves at faster speed, we set the turning degrees to 0.1 and 0.2. Every time frame, the system will assign a *Desired Body Orientation* to the posture.

In realtime, after we determine a posture's foot phase, we will also check if the leg on the ground is just striking the ground. It could simply be estimated by comparing the posture with the last posture from last time frame. We called a leg just striking the ground *Striking Foot*. For the other legs who stands on the ground but not the *Striking Foot*, we called them *Standing Foot*. To avoid foot sliding, during advancing or turning, our horse should stick its foot at the position where it strikes the ground.

We set the current striking position K as our reference position. For those postures which are moving forward without changing direction, we calculate the distance between the standing legs foot position P and the striking position K, and then we use the distance to compute root translation in the *XZ* plane. Figure 3.15 shows a horse feet's direction of movement. The red arrows denote the offset d^j of each foot. Equation 3.21 illustrates the computation. Where $p^r(i)$ is the root position at frame i and $p^r(0)$ is the initial root position when the landing foot first strikes the ground. N is the number of legs standing on the ground.

$$p^r(i) = p^r(i-1) + \frac{1}{N}\sum_{j=1}^{N} d^j, \tag{3.21}$$

Separately, for those turning postures, when a leg is a *Striking Foot*, we rotate its **Body** to the *Desired Body Orientation*. Then we set its foot position as the striking position K. Figure 3.16 illustrates the *Desired Body Orientation* and the striking position K. The dotted



(a) First Posture (b) Next Posture

Figure 3.15: *When the horse moving forwards, its legs moving backwards*

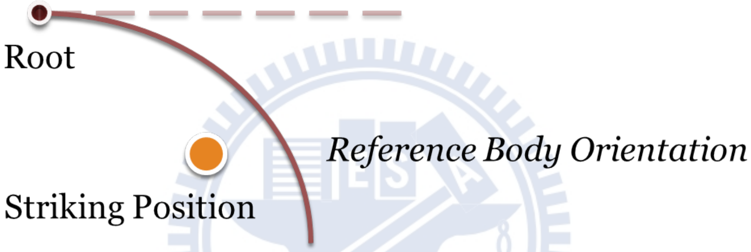


Figure 3.16: *The Reference Body Orientation and the Striking Position*

line demonstrates the original posture we obtained from our motion database.

For the *Standing Foot*, after we rotate the horse **Body** to the *Desired Body Orientation*, we calculate root translation by compute the distance between those foot positions P and their striking positions K. The computation is illustrated in Equation 3.21 and is shown in Figure 3.17. The dotted orange circle represents the striking position K, and the green circle is the foot position P. The **Body** has turned to the *Desired Body Orientation*.

In our method, we use the original posture's foot information to generate our root translation. Therefore, we preserve the reality of the motion and gait style. We use the standing legs y component for root's height. As to those postures having no legs on the ground, we add an average flying quantity at the advancing direction for their root to translate. In [24], they force the foot position to be fixed while the foot strikes on the ground. In contrast to their method, we calculate the foot's goal position before we plan the root trajectory. Our method uses the

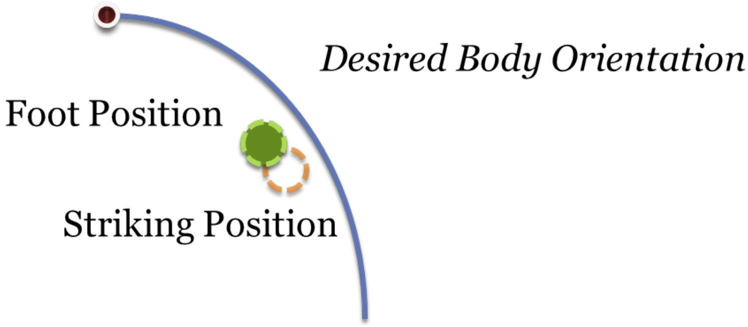


Figure 3.17: *Calculating root translation by the striking position and the foot position*

displacement of the horse standing feet to estimate the root translation.

CHAPTER 4

Experimental Results

In this chapter, we demonstrate the results of our horse locomotion system. We show the transition between horse gaits, and the turning process when we change horse direction. We will also exhibit some extended motions as horse leaping, climbing up and walking down a slope, and the motions of different skeleton and size horse. The images we used from Muybridge's photographs contain horse locomotion with rider or without rider. In this thesis, we did not consider the difference between these two situations. Nevertheless, a horse walking with rider adds the balancing problem to the horse. The horse should consider the weight of the rider and still remain balance in its locomotion. Figure 4.1 shows some snapshots of two examples of gait transition: *Cantor* to *Gallop* and *Trot* to *Rack*.

Gait Transition: In our system we demonstrate six types of horse locomotion gaits at different velocities. In Figure 4.2-Figure 4.6 we illustrate a process of our horse transition from the slowest speed to the fastest speed. In this case, we transit our horse gaits in speed order begin with *Walk* passing *Amble*, *Trot*, *Rack*, *Canter* to *Gallop*, but our system allow user to arbitrarily transit between any gaits, do not need to follow the speed order. In Figure 4.7, we show our horse transition from *Gallop* to *Amble*.

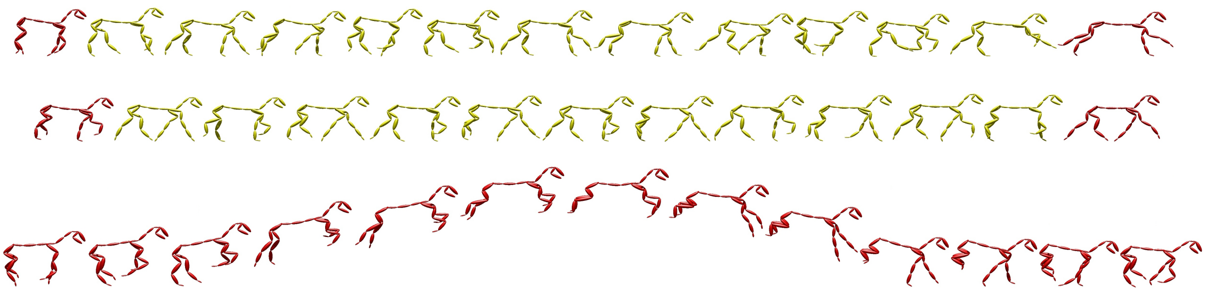


Figure 4.1: Top: snapshots of gait transition from *Cantor* to *Gallop*; Middle: snapshots of gait transition from *Trot* to *Rack*; Bottom: snapshots of a leap motion. Yellow color denotes that the horse is under gait transition.

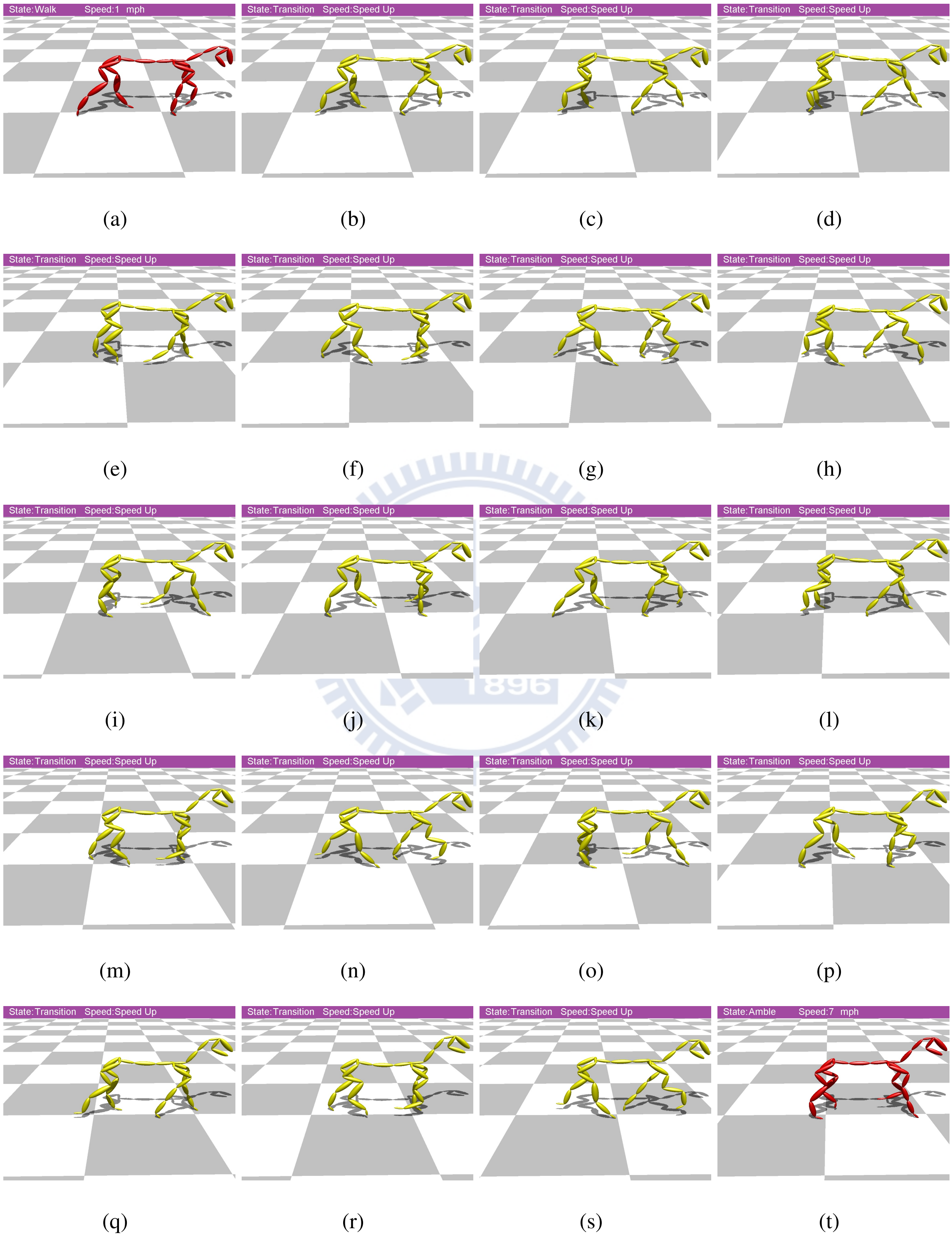
Turning: Figure 4.8 illustrates the turning process of our system. According to the fact that the horse uses slower gait proceeding a smaller distance per time frame, the horse also turns a smaller degree. Our horse turns 0.01 degree every time frame at the speed under 5 mph, 0.1 degree at speed between 6 mph to 25 mph, and 0.2 degree at speed above 25 mph. It turns 0.05 and 1.15 degree while the transition between different speed. In our system, the horse can turn direction during the gait transition and jumping.

Leaping: We also captured a leaping motion from Eadweard Muybridge's work [23]. Our system compares a section of the begin of the leaping motion and search the motion data we built to find the most similar section. We will mark the gait with the most similar section as the leaping-start gait. When a user claims to jump, our system will first let horse transit its gait to the leaping-start gait, and use linear interpolation to blend the similar section. The horse will change to the leaping motion after the blending part is over, and perform a leap. In the same way, our system also compares a section of the end of the leaping motion and search for the most similar section from our motion data, letting our horse blend to the destination gait. Figure 4.9 illustrates the leaping motion of our horse.

Climbing Up and Walking Down a Slope: We demonstrate this result by modifying our turning system. At the initial state our horse heading the global X direction, and the global Y

direction pointed to the sky. We modify the turning system by letting the horse rotate its *Body* from the global Y direction to the global Z direction. Figure 4.10 illustrates our horse climbing up and walking down a slope.

Changing Skeleton: We experiment our system by changing horse skeleton. We perform a horse with very short legs and the other horse with extremely long legs. This experience demonstrates that our root trajectory planning system can suit various horse skeletons and still preventing the horse from foot sliding. Figure 4.11 shows our results of different skeletons horse locomotion.


State:Walk Speed:1 mph
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Amble Speed:7 mph

(a) (b) (c) (d)

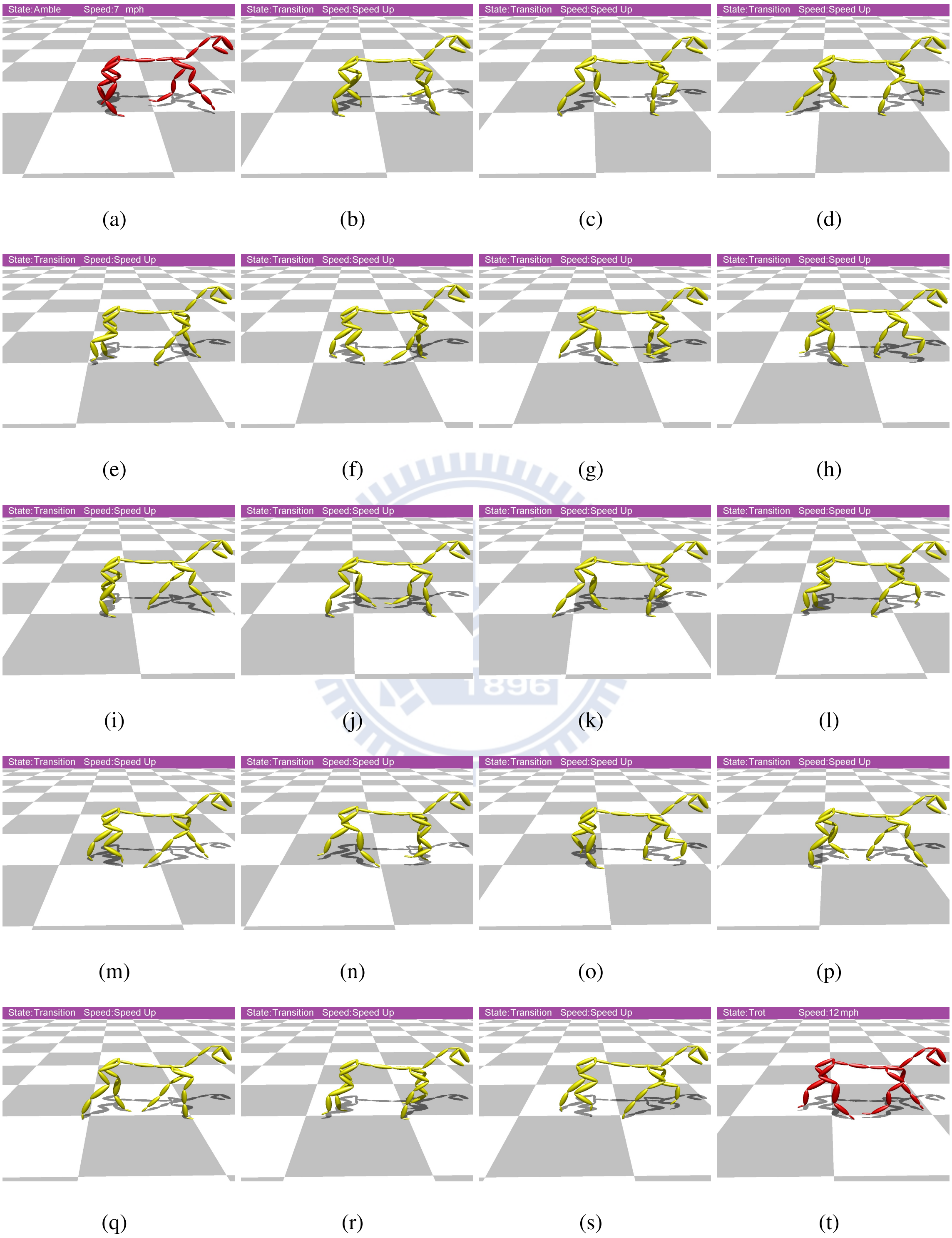
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 4.2: *Transition from Walk to Amble. The transition begin from* (d) *and end at* (r)


State:Amble Speed:7 mph
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Trot Speed:12mph

(a) (b) (c) (d)

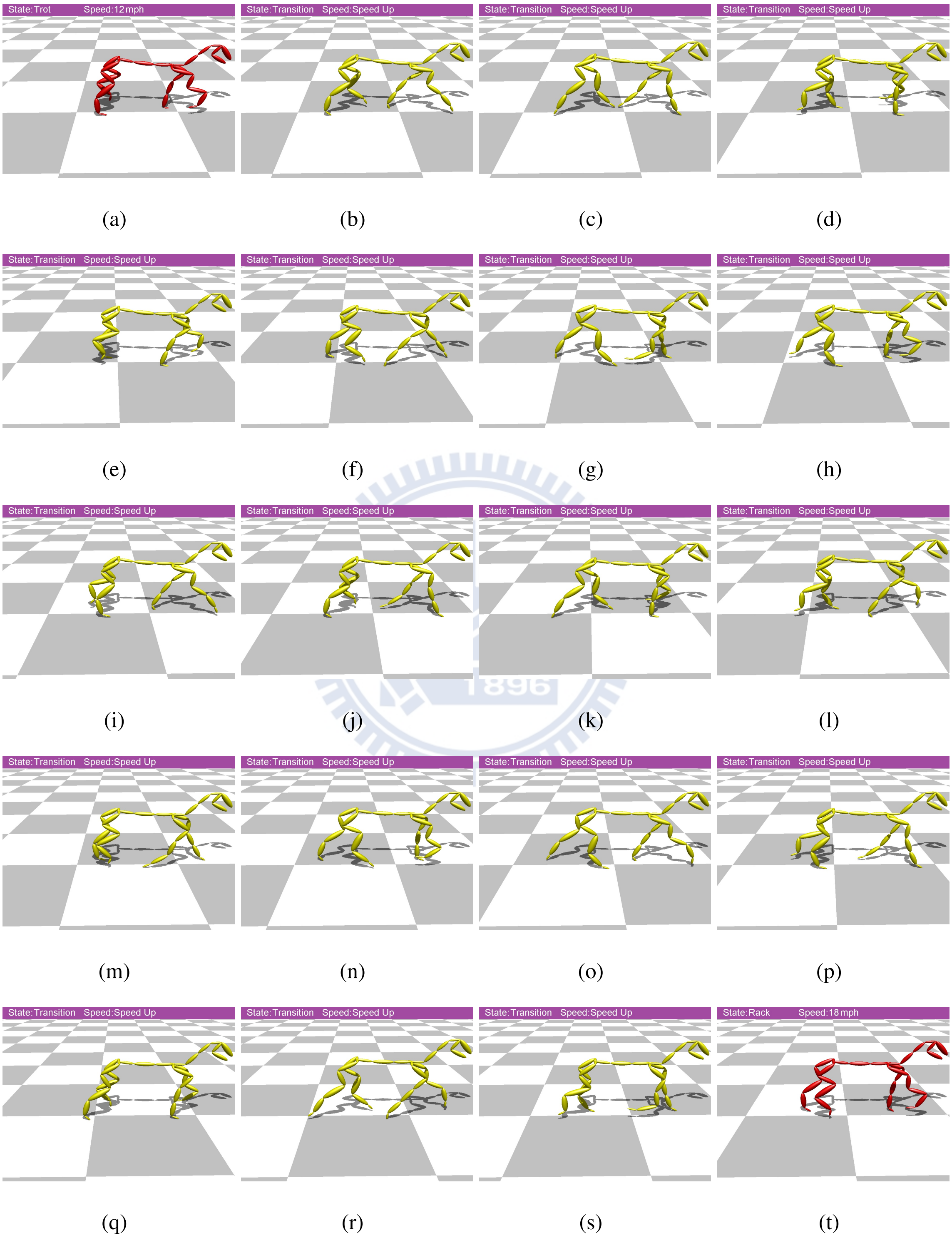
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 4.3: *Transition from Amble to Trot. The transition begin from* (d) *and end at* (q)


State:Trot Speed:12mph
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Rack Speed:18mph

(a) (b) (c) (d)

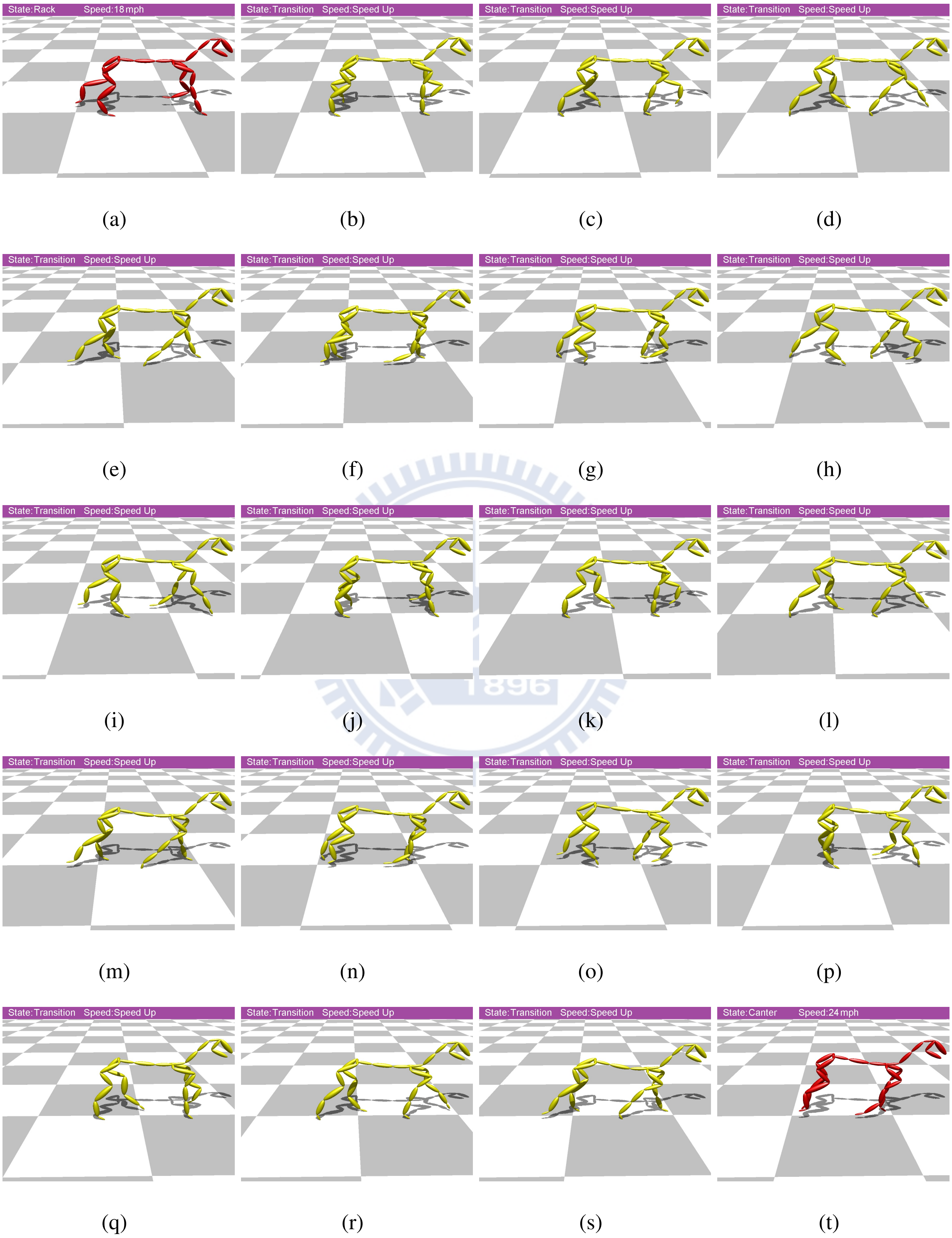
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 4.4: *Transition from Trot to Rack. The transition begin from* (d) *and end at* (s)



(a) (b) (c) (d)

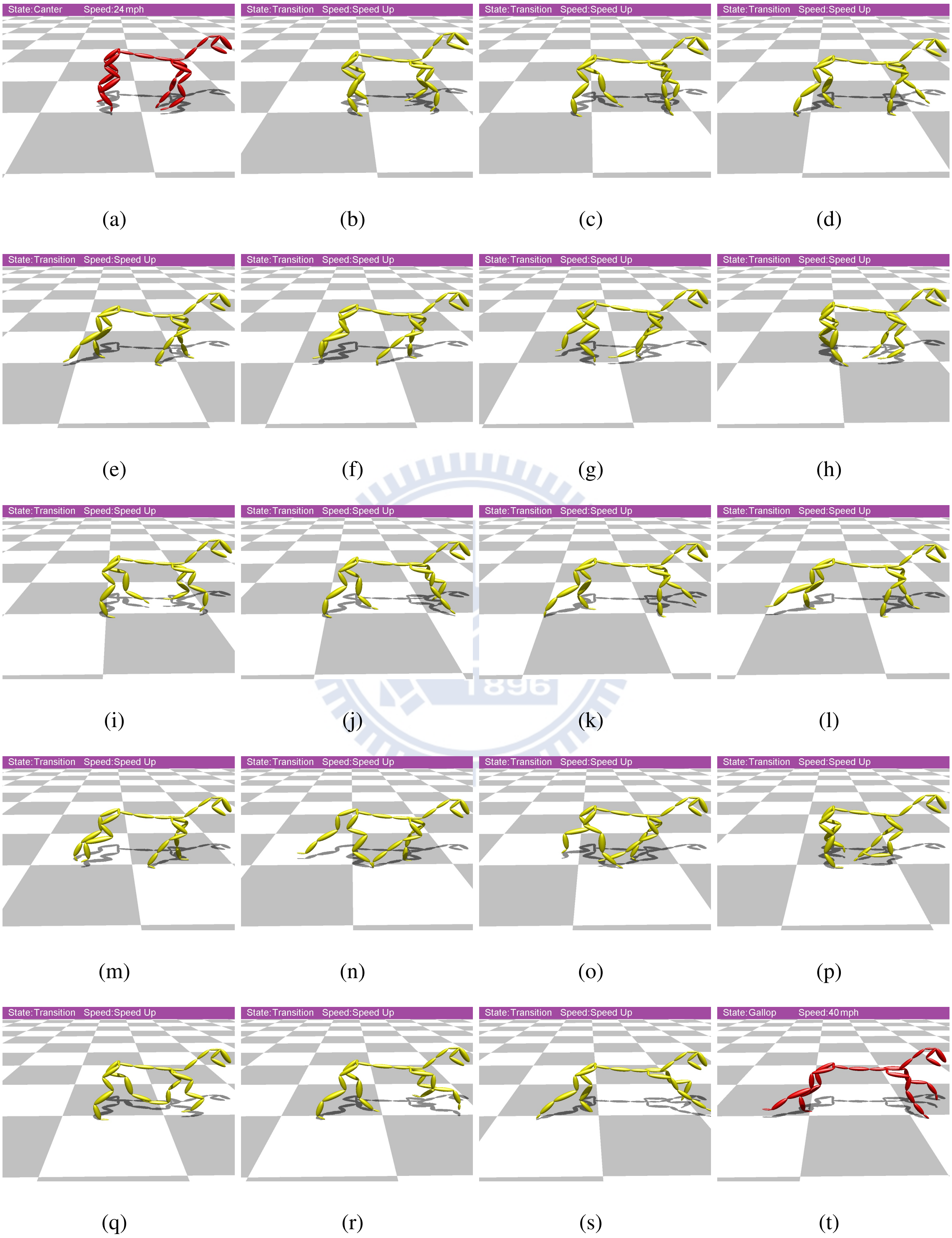
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 4.5: *Transition from Rack to Canter. The transition begin from* (d) *and end at* (s)


State:Canter Speed:24 mph
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Transition Speed:Speed Up
State:Gallop Speed:40 mph

(a) (b) (c) (d)

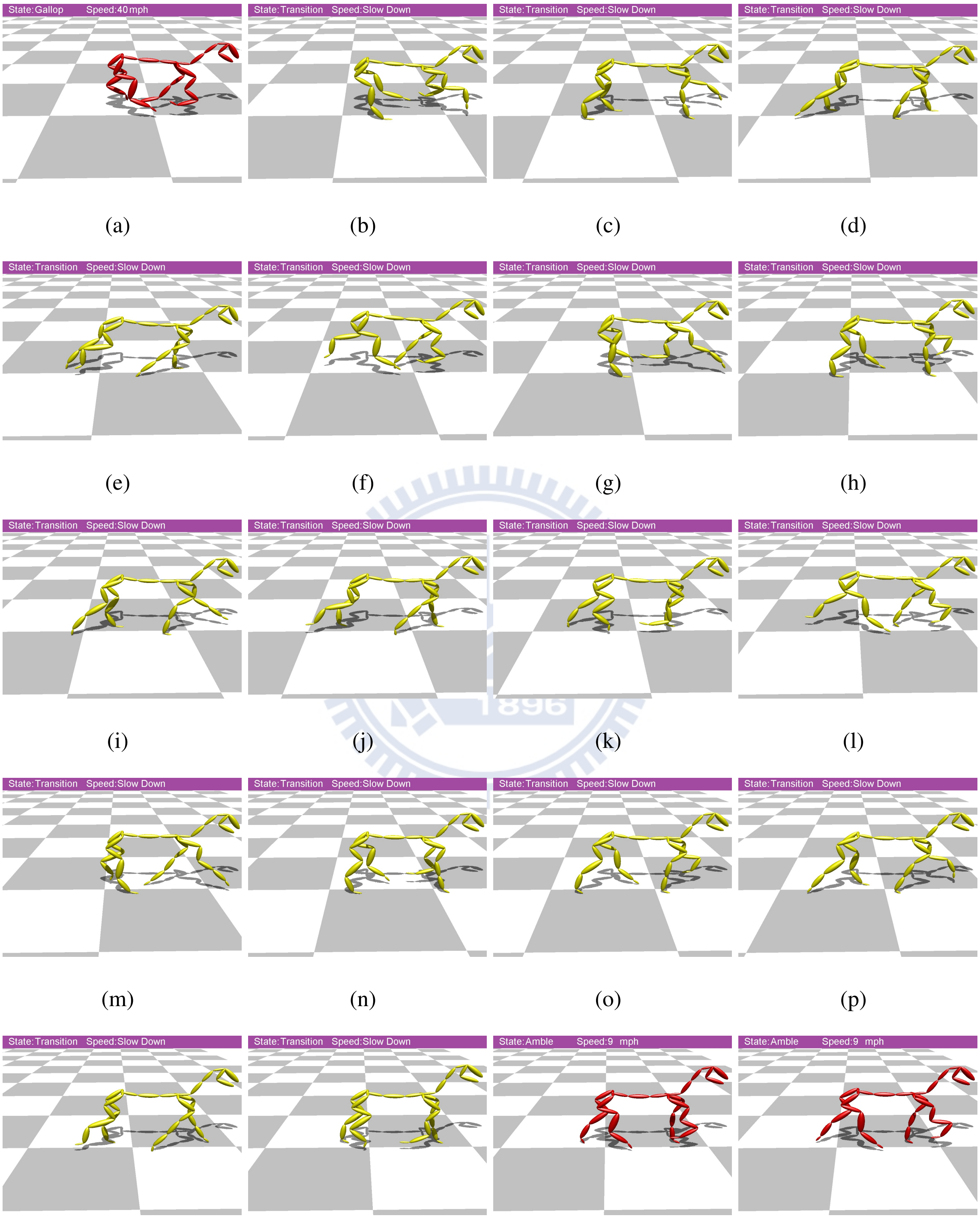
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 4.6: *Transition from Canter to Gallop. The transition begin from* (d) *and end at* (s)


State:Gallop Speed:40 mph
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Transition Speed:Slow Down
State:Amble Speed:9 mph
State:Amble Speed:9 mph

(a) (b) (c) (d)

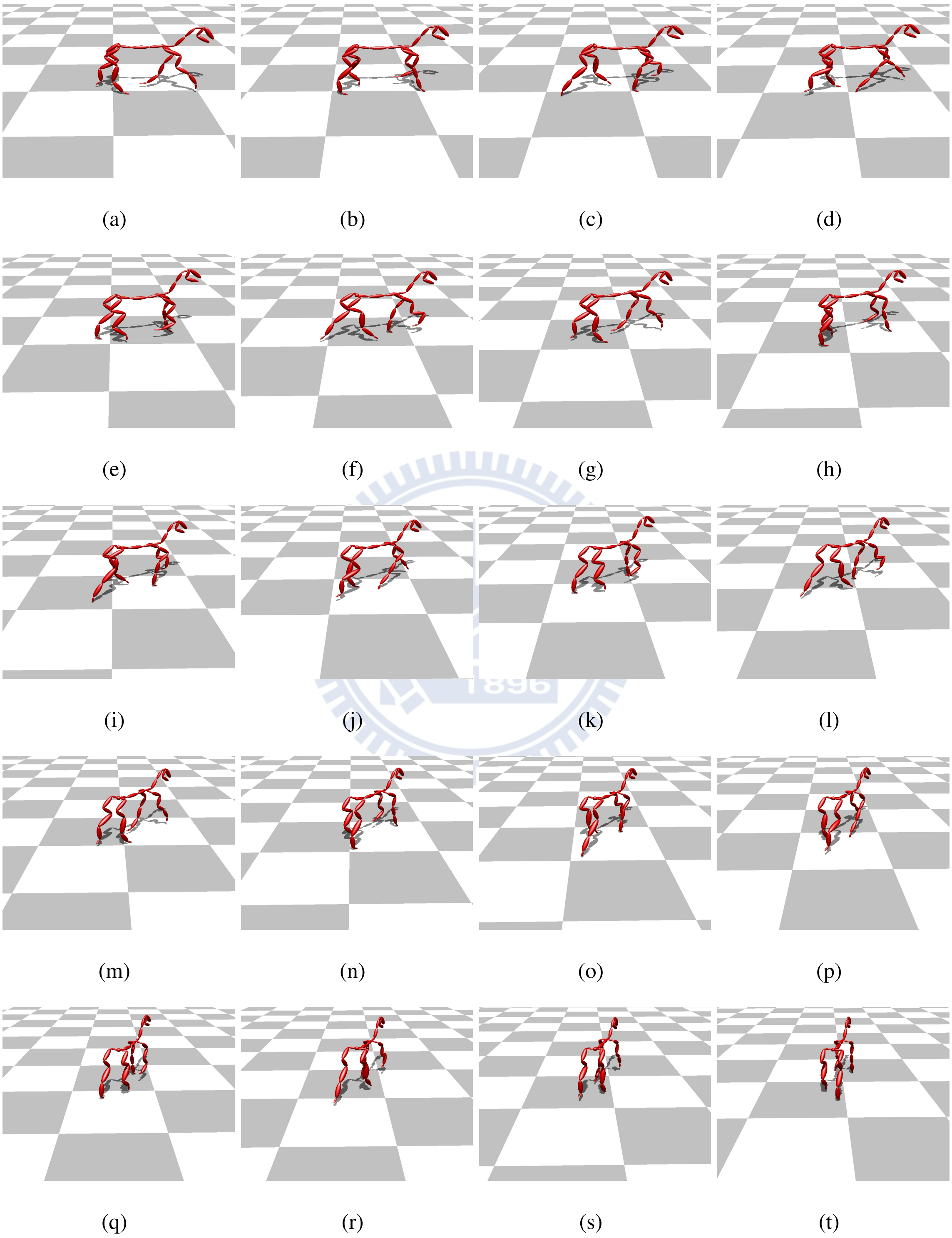
(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 4.7: *Transition from Gallop to Amble. The transition begin from* (d) *and end at* (s)


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 4.8: *Horse changing direction*

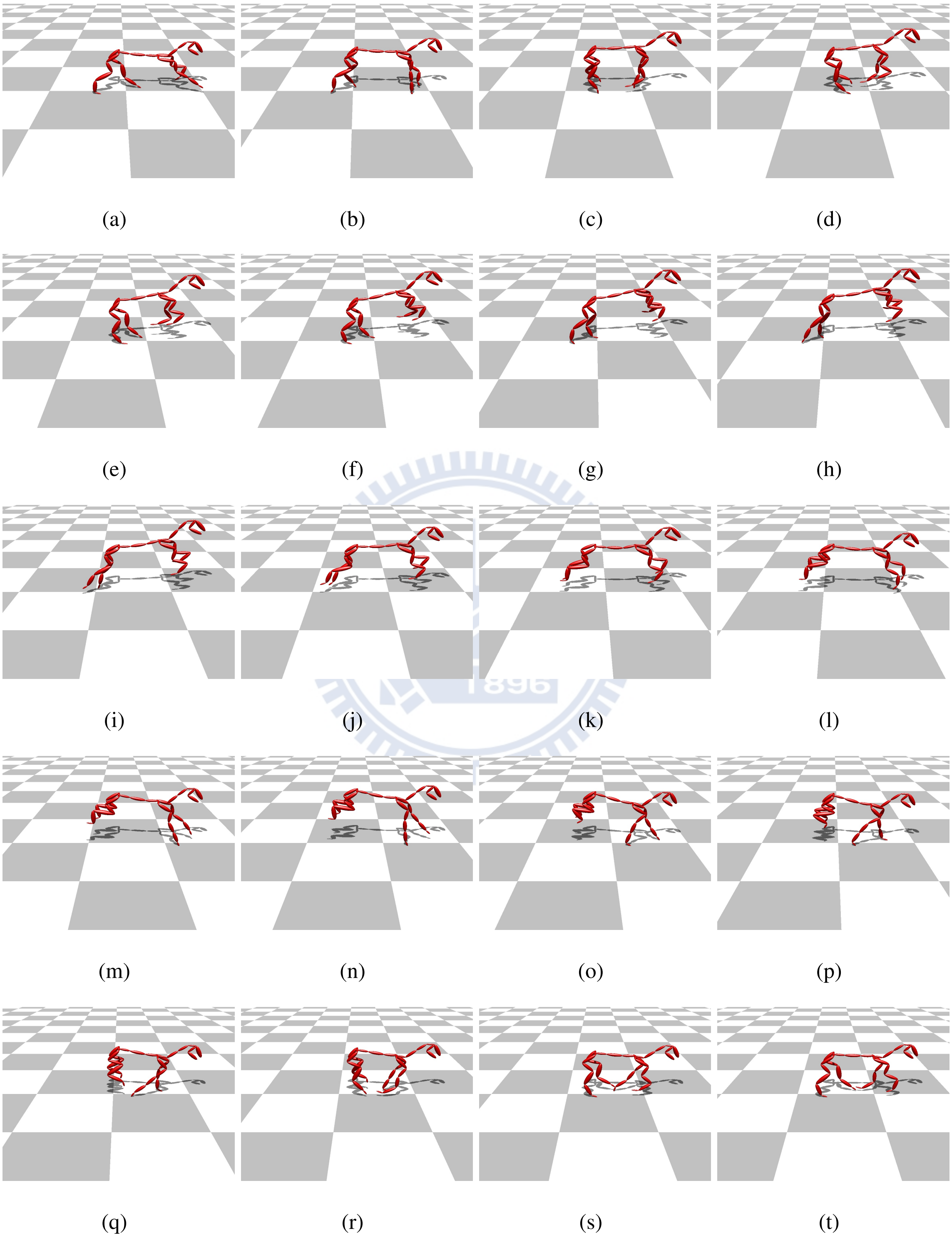


Figure 4.9: *Horse jumping*

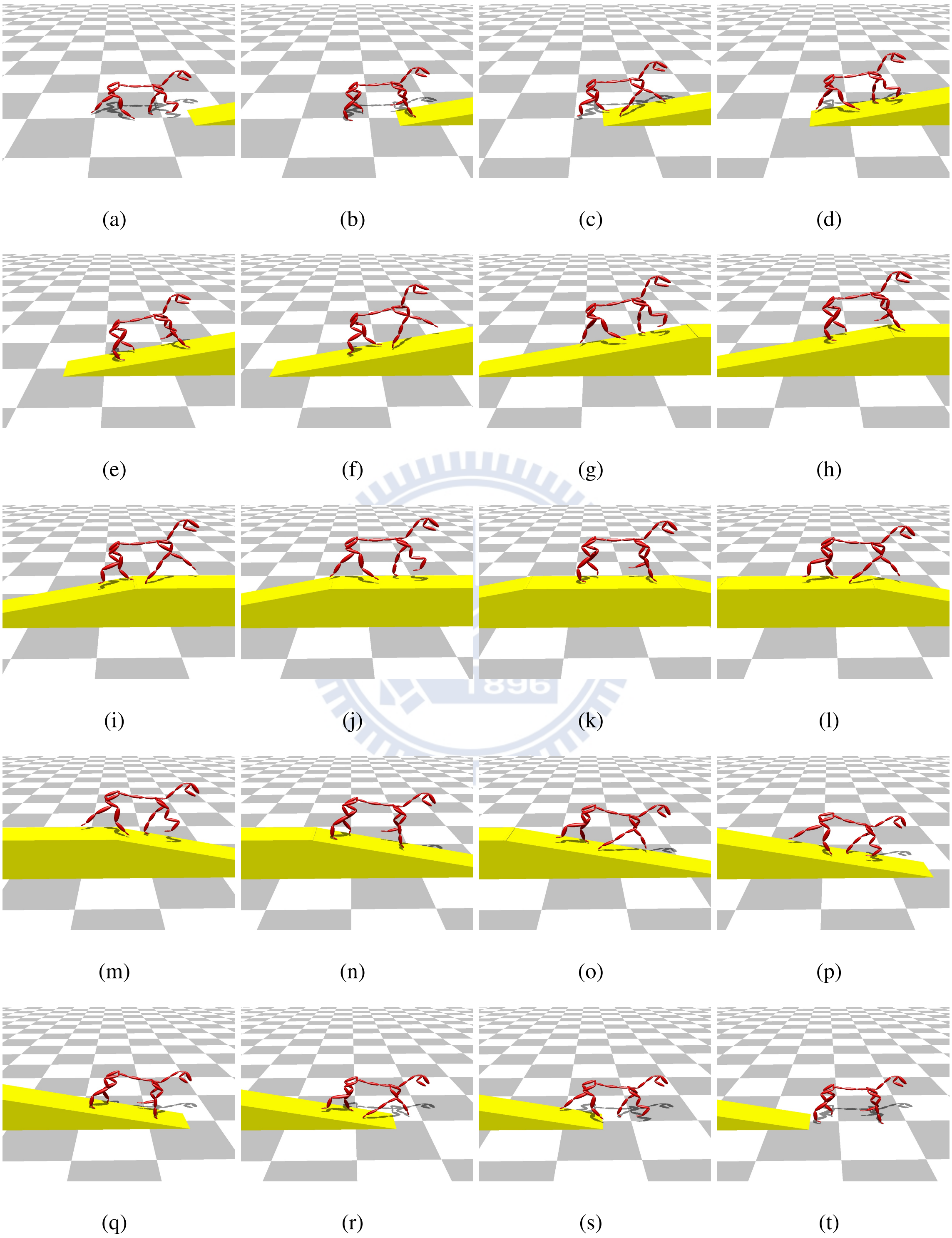
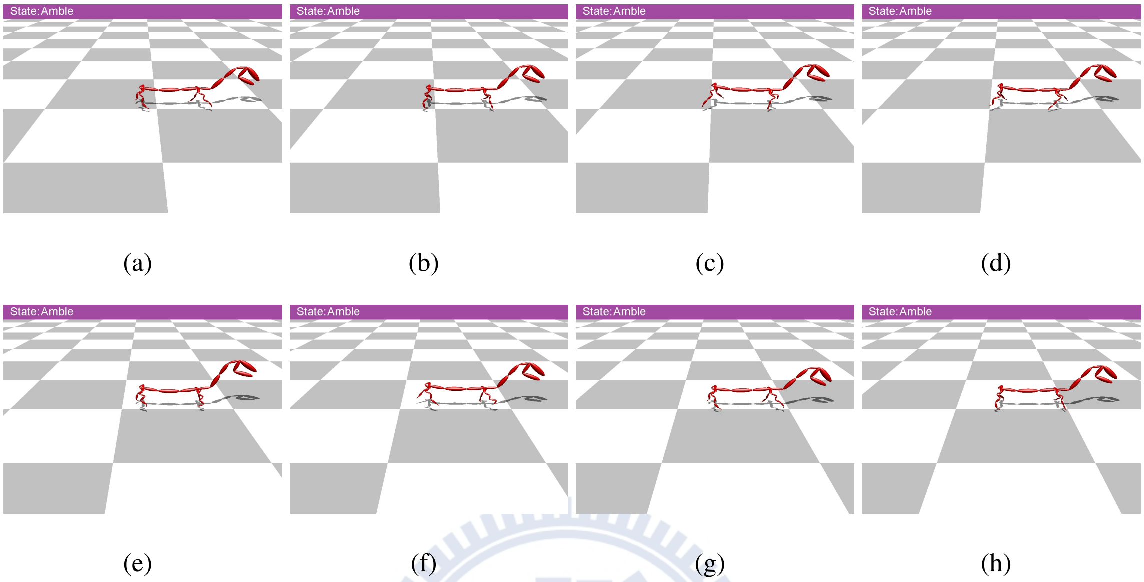


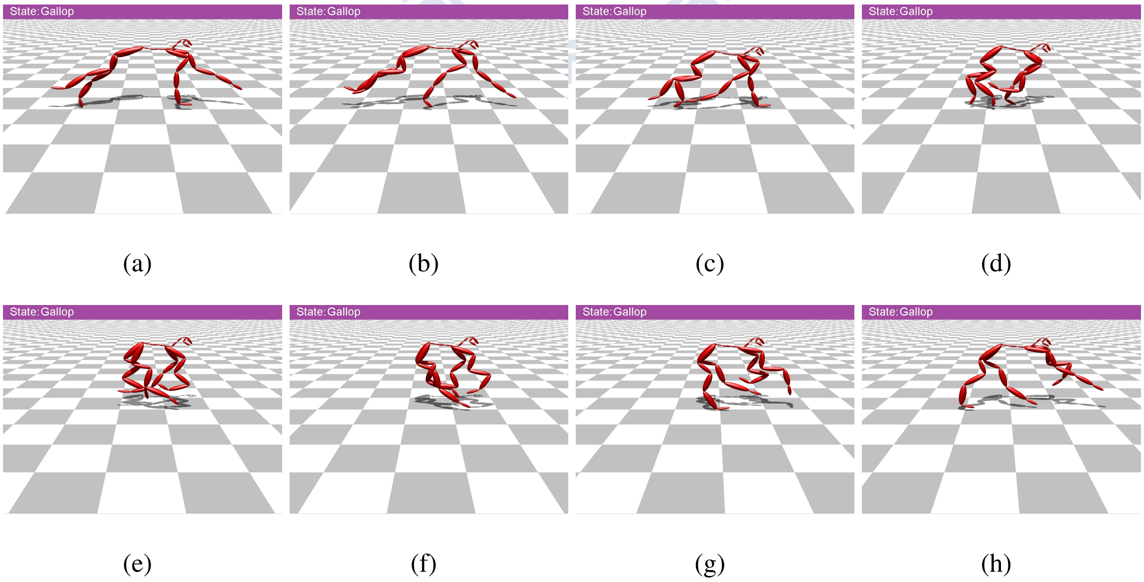
Figure 4.10: *Horse climbing up and walking down a slop*



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.11: *Horse with short legs*



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.12: *Horse with long legs*

CHAPTER 5

Conclusion and Future work

5.1 Conclusion

We present a realtime horse locomotion system using manually capture motion data to generate a wide speed range of horse motions. When a user changes our horse advancing speed, our system will automatically select the proper gait for our horse to use and deal with the transition between gaits. We also plan the root trajectory in realtime. Controlling our horse translation and rotation. To achieve this result, we manually capture a series of Eadweard Muybridge's horse locomotion photographs in MAYA. We also propose an asynchronous time warping method to translate motions between gaits. Our horse uses *Walk*, *Amble*, *Trot*, *Rack*, *Canter* and *Gallop* at different speeds. In every time frame we determine if the horse's legs are standing on the ground and compare the horse's moving distance to last time frame. Using the moving distance of the stance legs we can calculate the translation of horse root. When turning, we preserve the reality of the original motion. In the results, we also exhibit the motion of horse leaping. Furthermore, we modify our turning system to let our horse climbing up and walking down a slope. In the end, we change our horse skeleton to short legs and long legs to examine the effects.

5.2 Limitation and Future work

Our system manually captures animal locomotion through MAYA. Although this is an effective way to capture horse motion, it is not a procedural method. We need to spend a lot of time for every motion. In the future, we should improve our motion capture system to the automatic generating animal locomotion data via videos. In that case, we could capture more sets of motion data in a short time. Abe et al [1] proposed an optimization based algorithm which can let the captured motion satisfy high-level user constraints while preserving physical realism. Their method simulate a volleyball slam example by using a realtime inverse control mechanism. After a user specifies the position of the player's hand in mid-flight, the system can determine the correct linear interpolation of the sampled motions to meet the positional constraint on the hand. We could use their method to adjust our system to let users determine the height of the leaping motion. In the results, we present the motion of a horse climbing up and walking down a slope. Johansen [17] has proposed a method to analyze foot movement. He introduces a concept of a footbase, which is a constraint that combined heel and toe to retain the information about the alignment of a foot relative to the ground. His method could predict the place where the feet should land on the ground; calculate the trajectories and alignments of the feet when taking steps; and adjust the legs and hip height to accommodate the new foot position. In the future, we would like to add this function into our system so that our horse can not only changing direction but also walking on different terrains. We will also apply our system to simulate other quadrupedal locomotion, e.g. lions, pigs or elephants. Furthermore, we hope our system can be extended to handle multi-pedal locomotion. In the end, we will be able to build a zoo with different kinds of animals walking arbitrarily in it.

Bibliography

[1] Y. Abe, C. K. Liu, and Z. Popović. Momentum-based parameterization of dynamic character motion. *Graph. Models*, 68:194–211, 2006.

[2] R. Alexander. *Animal mechanics*. University of Washington Press, 1968.

[3] R. Alexander. The gaits of bipedal and quadrupedal animals. *The International Journal of Robotics Research*, 3(2):49–59, 1984.

[4] R. Alexander. *Optima for animals*. Princeton University Press, 1996.

[5] R. Alexander and A. Jayes. A dynamic similarity hypothesis for the gaits of quadrupedal mammals. *Journal of Zoology*, 201(1):135–152, 1983.

[6] D. Biery, T. M. Lenehan, and D. C. V. Sickle. *Textbook of Small Animal Orthopaedics*. 1985.

[7] A. Biewener, J. Thomason, A. Goodship, and L. Lanyon. Bone stress in the horse forelimb during locomotion at different gaits: A comparison of two experimental methods. *Journal of Biomechanics*, 16(8):565 – 576, 1983.

[8] B. Blumberg and T. A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. pages 47–54, 1995.

[9] M. Buehler, R. Playter, and M. Raibert. *Robots step outside*. 2005.

[10] S. Coros, A. Karpathy, B. Jones, L. Reveret, and M. van de Panne. Locomotion skills for simulated quadrupeds. *ACM Transactions on Graphics*, 30(4):59:1–59:12, 2011.

[11] L. Favreau, L. Reveret, C. Depraz, and M.-P. Cani. Animal gaits from video. In *Symposium on Computer Animation (SCA) 2004*, pages 277–286.

[12] M. Girard. Interactive design of 3-D computer-animated legged animal motion. In *Interactive 3D graphics*, pages 131–150, 1987.

[13] D. Golubovic, B. Li, and H. Hu. A hybrid software platform for sony aibo robots. In *Lecture Notes in Computer Science*, pages 478–486, 2003.

[14] C. Hecker, B. Raabe, R. W. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. In *ACM SIGGRAPH 2008 papers*, pages 27:1–27:11, 2008.

[15] D. F. Hoyt, S. J. Wickler, and E. A. Cogger. Time of contact and step length: The effect of limb length, running speed, load carrying and incline. *Journal of Experimental Biology*, 203:221–227, 2000.

[16] D. L. James and C. D. Twigg. Skinning mesh animations. *ACM Trans. Graph.*, 24:399–407, 2005.

[17] R. S. Johansen. Automated semi-procedural animation for character locomotion. Master's thesis, Aarhus University, 2009.

[18] E. Kokkevis, D. Metaxas, and N. I. Badler. Autonomous animation and control of four-legged animals. In *Graphics Interface*, pages 10–17, 1995.

[19] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng. A control architecture for quadruped locomotion over rough terrain. *IEEE International Conference on Robotics and Automation*, pages 811–818, 2008.

[20] P. Kry, L. Revéret, F. Faure, and M.-P. Cani. Modal locomotion: animating virtual characters with natural vibrations. *Comput. Graph. Forum*, 28(2):289–298, 2009.

[21] J. Laszlo, M. van de Panne, and E. Fiume. Interactive control for physically-based animation. In *Proceedings of SIGGRAPH 2000*, pages 201–208, 2000.

[22] S. A. Marsland and R. J. Lapeer. Physics-based animation of a trotting horse in a virtual environment. In *Proceedings of the Ninth International Conference on Information Visualisation*, pages 398–403, 2005.

[23] E. Muybridge. *Animals in Motion.* Dover Publications, Mineola, NY, 1957.

[24] S. I. Park, H. J. Shin, and S. Y. Shin. On-line locomotion generation based on motion blending. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 105–111, 2002.

[25] B. M. PLAYTER, R. and M. RAIBERT. Bigdog. In *Proceedings of SPIE*, 2006.

[26] I. Poulakakis, J. A. Smith, and M. Buehler. Modeling and experiments of untethered quadrupedal running with a bounding gait: The scout ii robot. *International Journal of Robotics Research*, 24:239–256, 2005.

[27] M. Raibert and J. Hodgins. Animation of dynamic legged locomotion. *ACM SIGGRAPH*, 25:349–358, 1991.

[28] M. H. Raibert. Trotting, pacing and bounding by a quadruped robot. *Journal of Biomechanics*, 23, Supplement 1:79 – 98, 1990.

[29] R. Ringrose. Self-stabilizing running. Technical report, In Proceedings of IEEE International Conference on Robotics and Automation, 1997.

[30] X. Shi, K. Zhou, Y. Tong, M. Desbrun, H. Bao, and B. Guo. Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics. *ACM Trans. Graph.*, 26, 2007.

[31] M. Simmons, J. Wilhelms, and A. V. Gelder. Model-based reconstruction for creature animation. In *Symposium on Computer animation*.

[32] K. Sims. Evolving virtual creatures. In *SIGGRAPH 1994*, pages 15–22, 1994.

[33] L. Skrba, L. Revéret, F. Hétroy, M.-P. Cani, and C. O'sullivan. Animating quadrupeds: Methods and applications. *Computer Graphics Forum*, 28(6):1541–1560, 2009.

[34] B. Tomlinson and B. Blumberg. Alphawolf: Social learning, emotion and development in autonomous virtual agents. In *In Proceedings of First GSFC/JPL Workshop on Radical Agent Concepts*, pages 35–45, 2002.

[35] N. Torkos. Footprint based quadruped animation. In *Master's thesis*, 1997.

[36] N. Torkos and M. van de Panne. Footprint-based quadruped motion synthesis. In *Graphics Interface*, 1998.

[37] M. Tsai, T. Lu, and T. Chen. Physically based motion adaptation for diverse articulated figures. *International Journal of Digital Content Technology*, 5(2), 2011.

[38] M. van de Panne. Parameterized gait synthesis. *IEEE Comput. Graph. Appl.*, 16:40–49, 1996.

[39] M. Walter, A. Fournier, and D. Menevaux. Integrating shape and pattern in mammalian models. In *SIGGRAPH '01*, pages 317–326, 2001.

[40] K. Wampler and Z. Popović. Optimal gait and form for animal locomotion. In *ACM SIGGRAPH*, pages 60:1–60:8, 2009.

[41] J. Wilhelms and A. V. Gelder. Anatomically based modeling. In *SIGGRAPH '97*, pages 173–180, 1997.

[42] M. Zucker, J. A. Bagnell, C. G. Atkeson, and J. Kuffner. An optimization approach to rough terrain locomotion. In *IEEE International Conference on Robotics and Automation*, pages 3589–3595, 2010.