Asian Journal of Scientific Research

ISSN 1992-1454

Asian Journal of Scientific Research 8 (2): 165-181, 2015
ISSN 1992-1454 / DOI: 10.3923/ajsr.2015.165.181
© 2015 Asian Network for Scientific Information

Automated Animation of Quadrupeds Using Procedural Programming Technique

Zeeshan Bhatti, Asadullah Shah, Ahmad Waqas and Mostafa Karbasi
Kulliyyah of Information and Communication Technology, International Islamic University, Malaysia

Corresponding Author: Zeeshan Bhatti, Kulliyyah of Information and Communication Technology, International Islamic University, Malaysia

ABSTRACT

In this research a simple yet efficient means of generating quadruped animation using trigonometric based mathematical equations with expressions were discussed. The technique proposed through this research is based on two tier animation control curve. The base simulation is driven through procedural model using trigonometric sinus and cosine functions to generate a sinusoidal motion curve with control parameters. The trigonometric sinus and cosine functions are used with control attributes to produce involuntary, procedurally generated base animation. In the top layer, animation is manipulated and customized by the animator through custom user controllable manipulators. These manipulators are created through an automated procedural rigging technique, based on forward and inverse kinematics controls. The User is provided with various attributes to control the locomotion gaits and create varying types of gaits from walking, running and trotting with complete custom mechanism to easily extend the base simulation as per requirements through an intuitive Graphical User Interface (GUI).

Key words: Quadruped animation, simulation, procedural animation, trigonometric expressions

INTRODUCTION

The use of quadruped character animation has become immensely popular and common since last decade in the field of Entertainment, Games, Movies, Advertisement, Scientific and Architectural Visualizations. Feature Films like the Shrek, Madagascar series, Chronicles of Narnia, Lord of The Rings etc., feature the quadruped motion with extreme accuracy and believable gait patterns. The procedure of bringing a virtual character to life using various animation techniques is long and wearisome process.

Animation in 3D applications usually happens in two primary ways, through Key-frame animation and Motion capture. Key-framing is the oldest and traditional technique which is immensely popular and still being widely used in the animation industry. The process of animating a character using Key-frame technique is very long and tedious. In this the animator has to manually move and reposition each body part through certain controllers and manipulators, positioning them with respect to time and keying their attributes. This process is lengthy and monotonous. The other technique used for animation is through Motion capture. There are various downsides to animating through Motion capture (mocap). First the cost of mocap technology is huge which can be around thousands of US dollars, thus making it very difficult for an ordinary production studio or freelance 3D animators to implement. Secondly, the animation artist then has to learn the tools and techniques of importing the mocap data and incorporating it onto the 3D character rig. Thirdly, problem with motion capture is that the mocap animation of quadrupeds is very difficult and practically impossible to get in various situations. The major downside of using motion capture based animation is the obtained result from sensory data is far from perfection and eventually the animator has to perform various clean-up operations to remove jitter and noise from the animation.

Procedural animation technique is quite old and vastly discussed amongst researchers, yet it has failed to achieve attention of mainstream production houses which still rely on Key-frame and Motion capture based animation techniques. The key advantage of using procedural animation is that motion is generated instantaneously and efficiently. Various software tools are available commercially, that can produce quadruped gaits procedurally. But mostly, such type of systems lack the accuracy and realism of quadruped motion and are usually observed to be robotic and mechanical (Skrba *et al.*, 2009; Cureton, 2013). A procedural technique based on trigonometric sinusoidal signal have been used to generate smooth motion curves. We implemented this technique inside Maya through creating expressions for each body part to generate their motion over time. The Expressions inside Maya are extremely powerful mechanism to control the motion of an object over time. Expressions are generally written using Maya Embedded Language and each expression code is automatically executed at every frame.

Another key element of our project is the automatic rigging of quadruped characters. Numerous tools and plugins are available in the market implementing diverse kinds of skeletal rigs for animating virtual characters. The tools set available in such type of software's are very efficient with vide variety but usually they seem to fail in satisfying the computer animator or riggers need. So, a character animator normally ends up building a custom skeletal rig for the ease of animation (Bhatti and Shah, 2012; Allen and Murdock, 2008). The process is also known as Character rigging. We have developed our own automatic character rigging system to facilitate our procedural animation setup.

QUADRUPED CHARACTERISTICS

Quadruped motion has always been an integral part of character animation and simulation. Various techniques and approaches have been used to understand the realistic motion gaits and parameters of quadrupeds.

Kinematics: In 3D computer graphics area, the Inverse Kinematics (IK) based approach for leg transformation is used widely as discussed by Kokkevis *et al.* (1995) and Torkos and van de Panne (1998), where the placement of the foot is decided first and then physical based models drives the body.

Controller based simulation: Whereas an advance technique of producing real-time interaction based simulation of quadrupeds, is through controller-based approach as discussed by Marsland and Lapeer (2005) and Coros *et al.* (2011).

Physics and dynamics: 3D forward dynamic simulations of quadruped gaits are introduced by Raibert and Hodgins in 1990 and 2008, respectively who developed control strategies for trotting, bounding and galloping gaits for a robot quadruped with a rigid body and extensible legs (Raibert, 1990). Similarly the game of Spore by Hecker *et al.* (2008), developed methods for generating procedural animation for arbitrary legged creatures, including locomotion patterns (Hecker *et al*., 2008). Conversely Wampler and Popovi (2009) developed a two-level optimization procedure for physics-based trajectories of periodic legged locomotion and use it to explore connections between form and function. Whereas Kwon and Shin (2005) have analyzed the center-of-mass trajectory of human walking and running motions to segment unlabeled motion sequences into motion half-cycles. On the other hand Coros *et al*. (2011) in their study discussed physics based simulation centered on integrated set of gaits and skills covering a wide range of motion repertoire of a dog.

Video based hybrids: However Skrba *et al*. (2008) in their study “Quadruped Animation” discussed various ways for achieving realistic quadruped motion, through “video-based acquisition, physics based models and inverse kinematics,” or some combination of the above (Skrba *et al*., 2008, 2009).

The major issue with current system of generating procedural animation is of controllability. With physics based system of procedural animation, the character’s motion cannot be controlled and modified or altered directly, as it is driven through virtual forces and torques. Further, as each individual character has its own style and personality which is usually shown through their motion i.e., walking style. This can also reflect the mood of character and his feeling or thought. For example if he is sad, then he may walk slowly with depressed look and feel to his walking style. This property cannot or is hardly ever addressed in physics based procedurally generated dynamic motions.

MATERIALS AND METHODS

The automation of quadruped animation is achieved through trigonometric sinus and cosine functions with control parameters generating sinusoidal waves for motion gaits as discussed by Bhatti *et al*. (2013). Initially a rigorous quadruped motion analysis was done to understand the gait patters of each motion type. For this study Horse and Lion characters were concentrated and same algorithm expressions were applied on both characters types.

Motion analysis: At first video reference was used to extract frame by frame motion footage of each character and key poses were segregated to identify where each character changes its pose. For example, from stance phase where each feet is planted on ground to swing phase, where each leg starts its swing motion as shown in Fig. 1 with lion motion.

Then a footfall diagram was crafted to understand and implement the each foot placement timings and patterns for each individual character as shown in Fig. 2 for horse walk and Fig. 3 shows the foot fall pattern of lion walk motion.

From the foot fall phase diagram of each character, a gait graph was generated to provide the timing information of each leg during motion as shown in Fig. 4 illustrating the gait graph of lion walk motion. Further results were obtained from Liszewski (2011), who discussed various quadruped gaits, along with research work of Huang *et al*. (2013) and McLaughlin (2012) were also used for motion analysis.

It was analyzed from this study that the normal quadruped motion follows periodic gait patterns, where each joint rotates with periodic signal. This means that the angle value of each joint changes periodically over time. Thus the sinus and cosine functions are used to generate simple sinusoidal curve representing periodic quadruped motion.

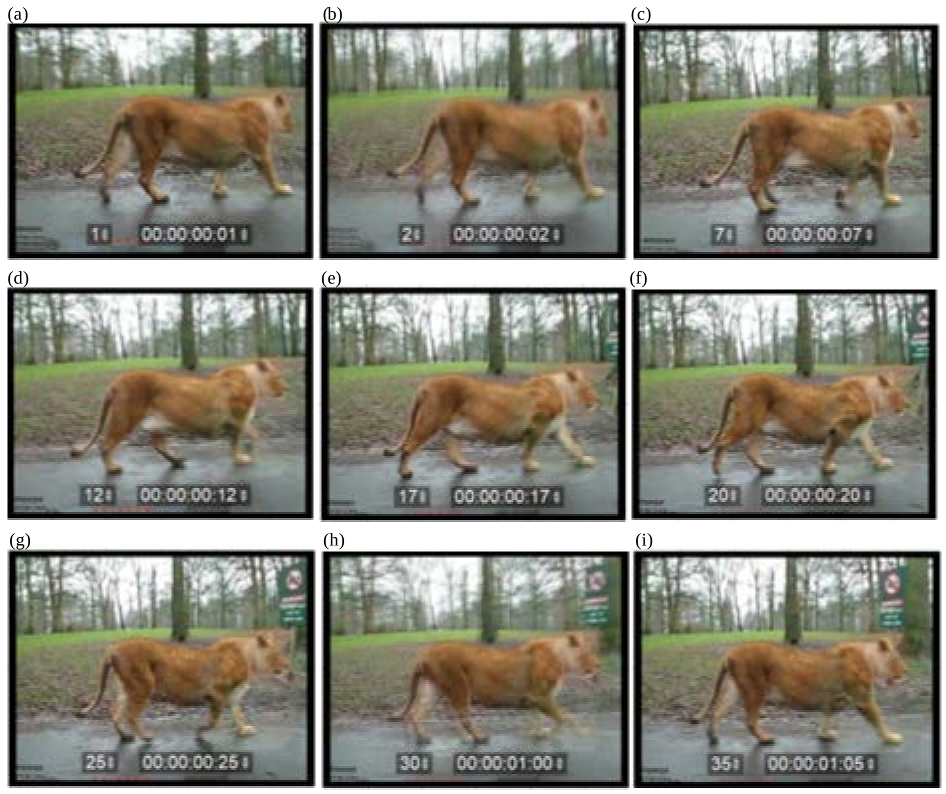


Fig. 1(a-i): Frame by frame breakdown of lion walk motion as represented in Frame (a) 1, (b) 2, (c) 7, (d) 12, (e) 17, (f) 20, (g) 25, (h) 30 and (i) 35

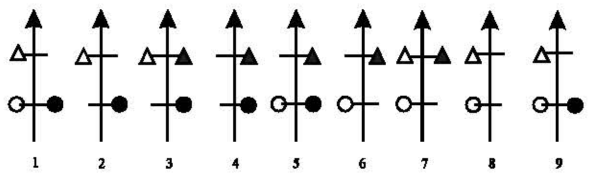


Fig. 2: Footfall phases of horse walk motion

Overview of system: The desired system has been divided into several key components as shown in Fig. 5. The mathematical expression consists of trigonometric equations that consists of sinus and cosine function with control parameters. The expressions are applied on each body part individually to generate periodic motion for each gait. The user input is provided through the Graphical User Interface (GUI) that derives and controls the gait behavior during runtime. The resulting motion curve is summed over and applied on the dynamic dual layer based quadruped rig for final output.

Quadruped rig model: Primarily an abstraction of the quadruped skeletal structure is proposed with single joint chain representing each of the four legs of the character facilitating each foot placement calculations independently as proposed by Coros *et al.* (2011). This proposed model was further enhanced and added few additional joints for various body parts. A flexible abstract spine

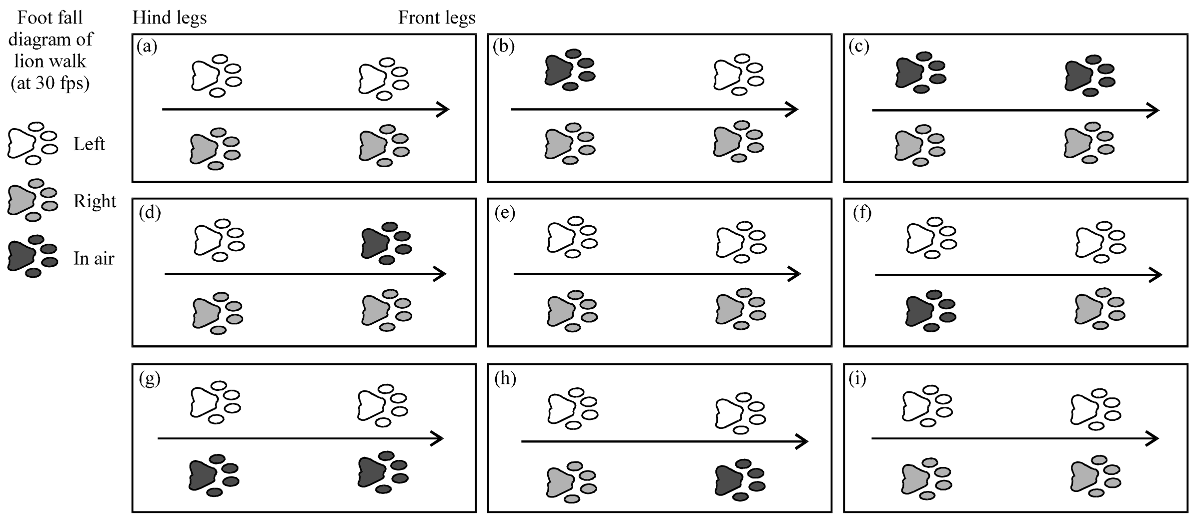


Fig. 3(a-i): Footfall phases of lion walk motion at Frame (a) 1, (b) 2, (c) 7, (d) 12, (e) 17, (f) 20, (g) 25, (h) 30 and (i) 35

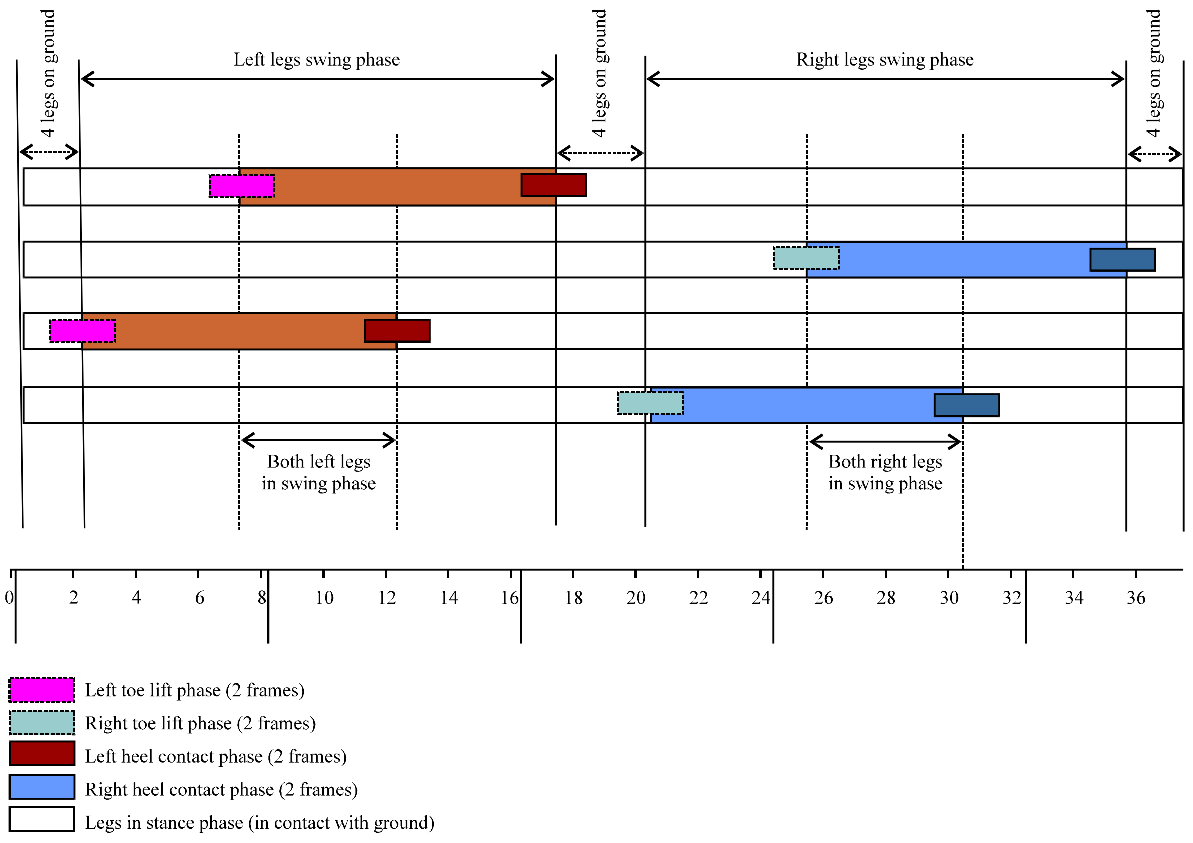


Fig. 4: Gait graph of lion walk

multi-joint structure was introduced which helps in achieving more natural motion during fast gaits such as galloping. The flexible spine connects front and back leg frames which together form a dual leg frame model with each leg frame making independent decisions regarding the footstep

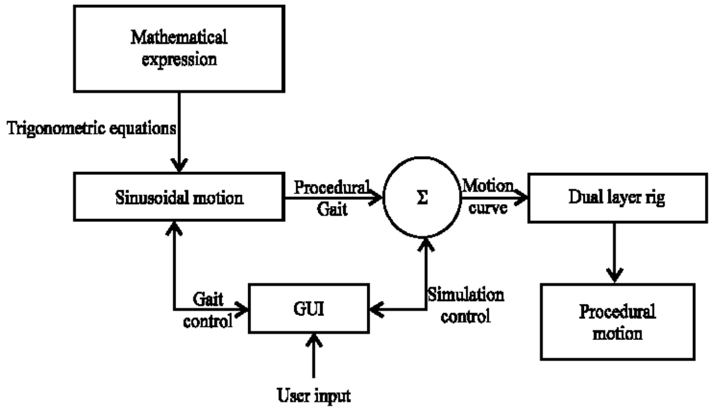


Fig. 5: Overview of procedural animation system

placement, height control and pitch control (Coros *et al.*, 2011). In tail and neck few additional joint were added to produce accurate rotational arcs during animation. Once the abstract model based mathematical algorithm has been achieved, then more specific and standard model is derived and equations are modified to produce simulation for a standard skeletal model which was developed using the techniques discussed in the study of Bhatti and Shah (2012). This skeletal joint chain is then duplicated and constrained to the original base skeleton, thus we now have two identical skeletal structures providing a dual layer based approach. The base skeletal structure is used for rigging whereas custom controls are provided for user manipulation according to the animators need. The second skeleton is used for procedural simulation and is driven through the trigonometric equations. This system of dual layer quadruped model is illustrated in Fig. 6.

The rigging process is illustrated in Fig. 7, where this process is broken down into 4 steps. First steps involves creating an abstract quadruped skeletal structure based on widgets as discussed in Bhatti and Shah (2012) and Bhati *et al.* (2013). The widgets are then manipulated by the user to fit into any quadruped character type like horse, lion, dog, etc. Then the system automatically generates the skeletal joints and finally also rigs the character with custom controls. All this process is done using procedural rigging techniques.

Mathematical modeling: It has been observed that motion gaits of most of the quadruped characters have same symmetrical patterns across both right and left legs. The difference however is that the gait pattern between both right and left legs occurs with different time shifts and phases (Griffin *et al.*, 2004; Marhefka *et al.*, 2003). Thus consequently same motion equation is shared between legs containing varying time shift parameters.

The mathematical expression generator in our system plays the key role in producing automated motion gaits and is derived from Bhatti *et al.* (2013) study. As we know that the periodic sinusoidal motion is driven through the time variable, so the time in 'frames (t_f)' has been used with 24 frame-per-second (fps). The mathematical model developed for this system is segregated in three main parts as shown in Fig. 8. First part involves various control parameters that effect the behavior of entire quadruped motion, such as Motion frequency denoted by M_{freq}, Body position (B_p) and Body oscillation (B_{osc}). Motion frequency determines the overall speed of motion and can be altered during runtime to affect the behavior of animation dynamically. The B_p parameter determines the position of the body from ground whereas B_{ocs} controls the oscillation of body. Both

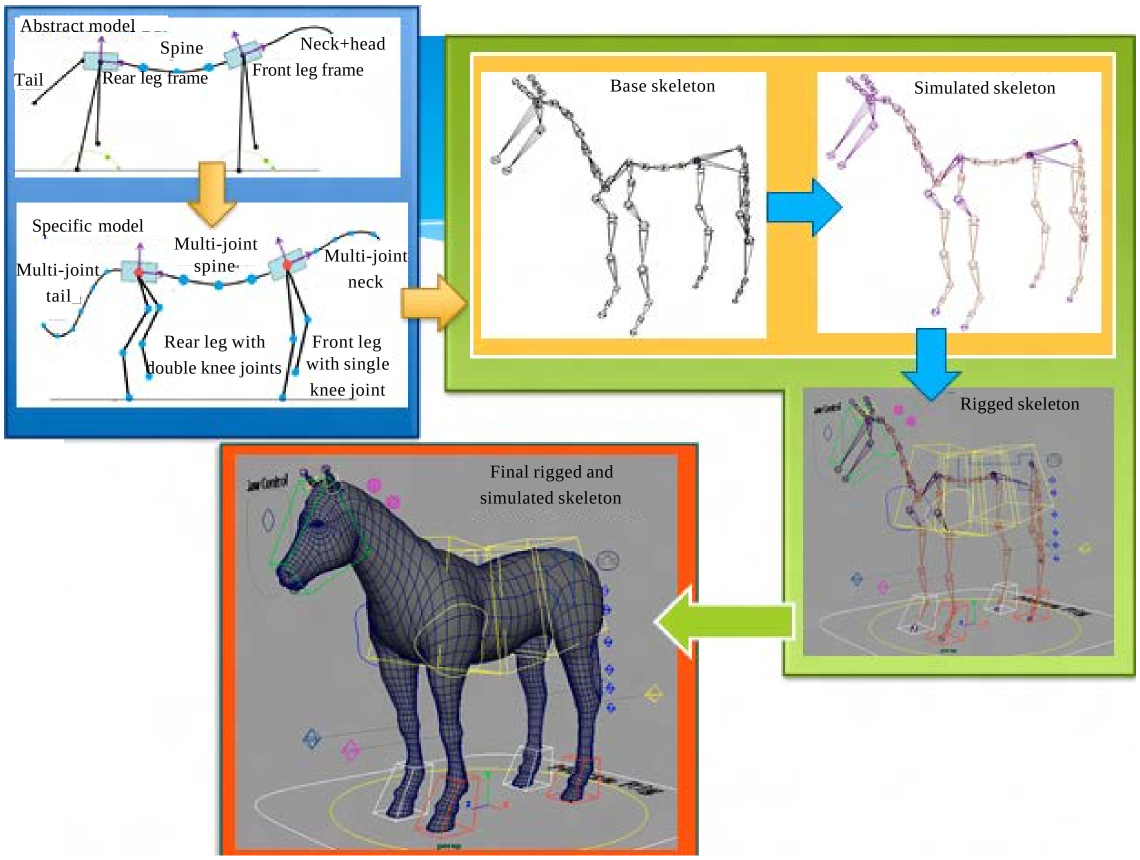


Fig. 6: Quadruped dual layer rigged character model

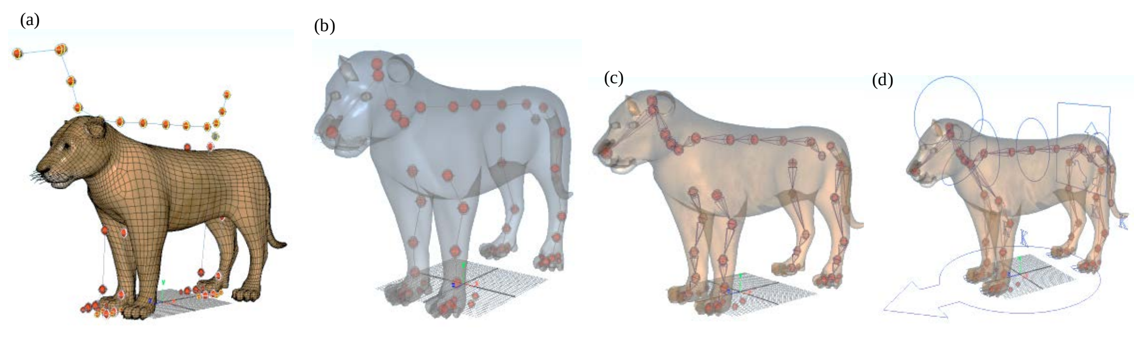


Fig. 7(a-d): Procedural rigging process of quadruped, (a) Step 1, (b) 2, (c) 3 and (d) 4

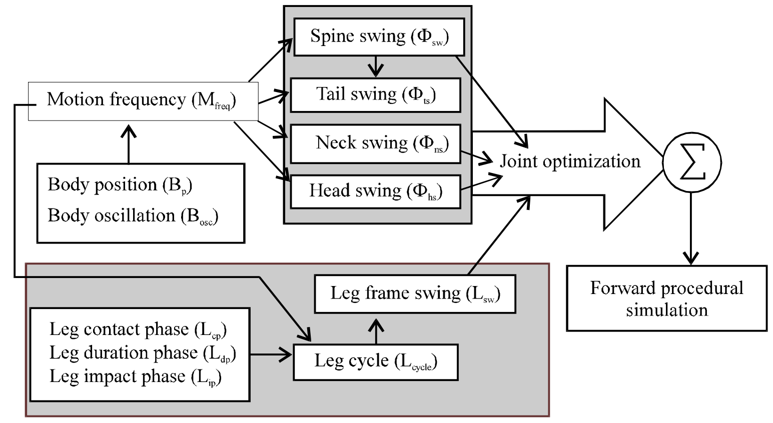

Spine swing (Φ_{sw})
Motion frequency (M_{freq})
Tail swing (Φ_{ts})
Neck swing (Φ_{ns})
Joint optimization
$\sum$
Head swing (Φ_{hs})
Body position (B_p)
Body oscillation (B_{osc})
Leg frame swing (L_{sw})
Forward procedural simulation
Leg contact phase (L_{cp})
Leg duration phase (L_{dp})
Leg impact phase (L_{ip})
Leg cycle (L_{cycle})

Fig. 8: Overview of controller based mathematical model

these can be modified at runtime through GUI. The second part involves parameters that individually affect the spine, neck, head and tail parts. All these body parts have separate trigonometric equations that drive their periodic gaits through sinusoidal motion curves. The last part involves various parameters that are used to control each leg motion exclusively. These parameters are applied on each leg separately and regulate various gait characteristics of each leg during motion.

Finally, the overall control loop is driven through Motion frequency (M_{freq}) and sinus function with control parameters, provide torques to a forward dynamics simulator at every time step. The motion parameters of each body part is calculated and generated individually at each frame and applied on that body part. The example of spine (sw) motion equation used to control the torso region of the character is calculated using the sine function as shown in Eq. 1:

$$\phi_{Sw} = \sin (t_f \times M_{freq} \text{ x } C_{error}) \times B_{osc} \tag{1}$$

where, the current time in frame (t_f) is obtained from the timeline and is always $t_f > 1$. C_{error} is a constant used for counter gait error for optimization. the B_{osc} is >0 and maximum value is 10 and represents the natural bounce in the body during motion. The head and tail motion curves are also obtained by using the same trigonometric sine function with motion frequency and optimizing with joint error (J_{error}) constant as given in Eq. 2:

$$\Phi_{ht} = \sin (t_f \times M_{freq} \times J_{error}) \tag{2}$$

The joint chain of each leg is initially constrained and attached to Inverse Kinematics (IK) based controller for achieving realistic motion. Each leg motion is calculated individually with leg impact phase (L_{dp}) and leg contact phase (L_{cp}) controlling the gait position while leg duration phase (L_{dp}) determines the time the leg is in motion. The IK based translation of the each leg is obtained using Eq. 3. The FPS is a constant variable used for the frame per second count and L_{cycle} is calculated from number of cyclic leg motion of a horse per second:

$$\phi_{lsa} = \frac{P_{id} - 2\times\pi\times Jps + L_{cycle} - L_{rp}}{L_{cycle}} \tag{3}$$

As discussed that the gait of quadruped motion is symmetrical which means that the motion patterns of left and right legs are equivalent but vary in time shifts with phase difference of 50% (Yamamoto, 2000).

For each leg the phase period was calculated using Fourier series as time shift in cosine function, illustrated in Eq. 4 obtained from the thesis of Cureton (2013). When each stride is at midway, the time shift is considered. The stride length and time was determined during the motion analysis.

$$\begin{aligned} &\text{Left leg}: \sum_{n=1}^{251} \text{Amplitude}\times\cos(\omega_s\times(n)\times t+\text{Phase}) \\ &\text{Right leg}: \sum_{n=1}^{251} \text{Amplitude}\times\cos(\omega_s\times(n)\times(t-t\_\text{shift})+\text{Phase}) \\ &t\_\text{shift} = \frac{1}{2}\times(\text{stride length}) \end{aligned} \tag{4}$$

The sternum joints of the horse skeleton are moved in the corresponding 'z' direction using the cosine trigonometric function with counter gait error (C_{error}) and Joint error optimization parameters multiplied with the desired leg oscillation as given in Eq. 5:

$$\begin{aligned} \tau_{leg} &= L_{lgt}\times\cos(\phi_{lsa}\times\pi)+L_{pos} \\ \tau_z &= \cos(t_f\times M_{freq} - \left(\frac{\pi}{4}\right))\times L_{osc}\times C_{error} \end{aligned} \tag{5}$$

Leg motion cycle: The motion cycles considered here are cyclic walk or amble cycles, where the character is walking with a constant velocity. The same applies to motion with turning, where a character is turning around while walking (Johansen, 2009). There can be any number of legs and the movements of the different legs are analyzed completely independently from each other. For each leg a heuristic is used to find a stance time (The point in time in the motion cycle when the leg is standing most firmly and neutrally on the ground).

Time is defined cyclically for the motion cycles with a value of 0.0 denoting the start of the cycle and the end of cycle is represented by 1.0, after which the cycle starts over again. In addition to the overall motion cycle, each leg has its own leg cycle as shown in Fig. 9.The start and end of these leg cycles are marked by the stance times of the respective legs, the start times of the leg cycles are not arbitrary at all (Johansen, 2009).

RESULTS AND DISCUSSION

The use of simple trigonometric functions and algebra provided us with a basic set of equations that can be easily used for generating high end quadruped motion. Figure 10 shows the skeletal model of a quadruped character (lion) built on inverse kinematics with elementary set of rigging controls. The skeletal joint hierarchy was used to create the dual layered based animation setup.

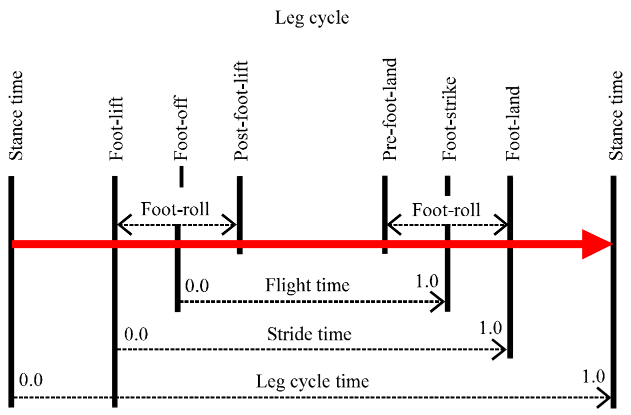


Fig. 9: Leg motion cycle from stance to motion to stance (Johansen, 2009)

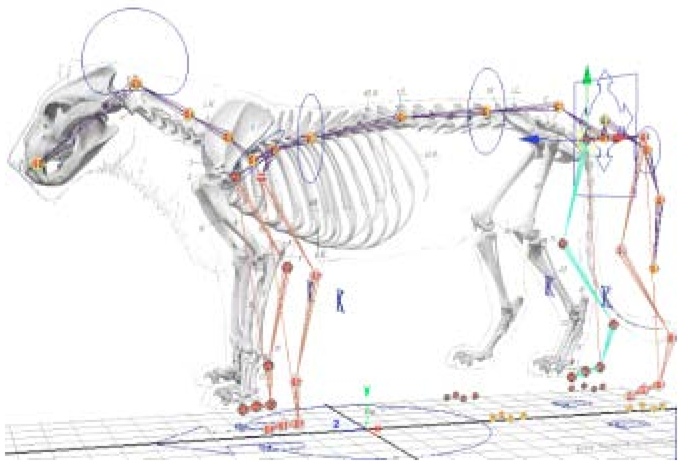


Fig. 10: Skeletal joint chain of horse character

The first layer of joints chain is controlled by the trigonometric expressions and the second layer provides user controls for custom manipulation of the animated character. Using this model we can produce several animation types for quadruped characters including walking, trotting, ambling and racking. The gaits transition was achieved using the Leg Impact Phase (L_{ip}) and duration (L_{dp}) of contact with the ground.

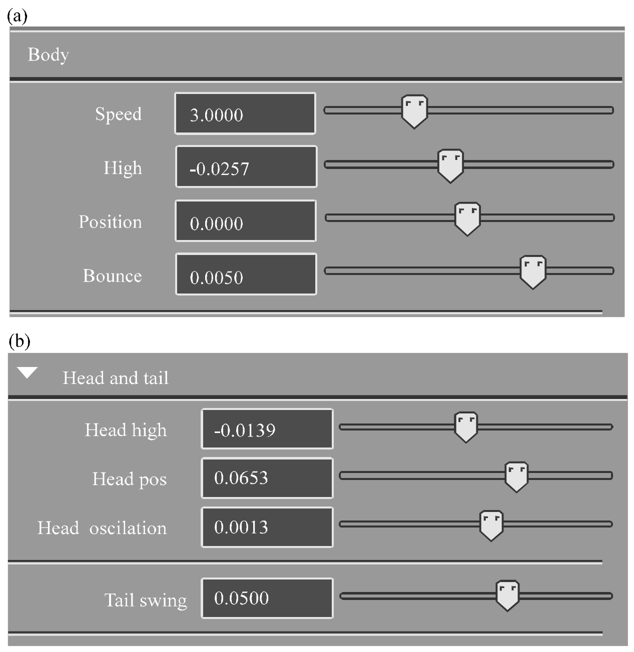
Inverse Kinematics (IK) system is used to control joint chains of the spine, tail and neck individually. Each IK is driven through expressions that define the oscillation motion curves according to the gaits. For leg joint each IK were used with expressions controlling the translating attribute. The joint of the foot are separately controlled using simple forward kinematics.

An easy to use graphical user interface was designed in MAYA that provided the basic control over the simulation.

Figure 11a shows the parameters used to control the motion of the body. The 'Speed' attribute determines the overall motion frequency of the characters. The 'High' attribute is the height of the head, neck and tail with respect to the spine and hip joint of the character during motion. The 'Bounce' attribute controls the overall bounce in the characters body during any type of motion.

Figure 11b shows the head and tail attributes. The head motion is controlled by three parameters; ‘Head High’ controls the height of head with ‘Head pos’ determines the position of head with respect to its orientation from neck. The ‘Head oscillation’ controls the number of oscillations of head joints during any type of motion. Similarly the tail swing attribute controls the swing frequency of the tail during motion. All of these attributes can be changed dynamically during runtime to change the behavior of the quadruped motion.

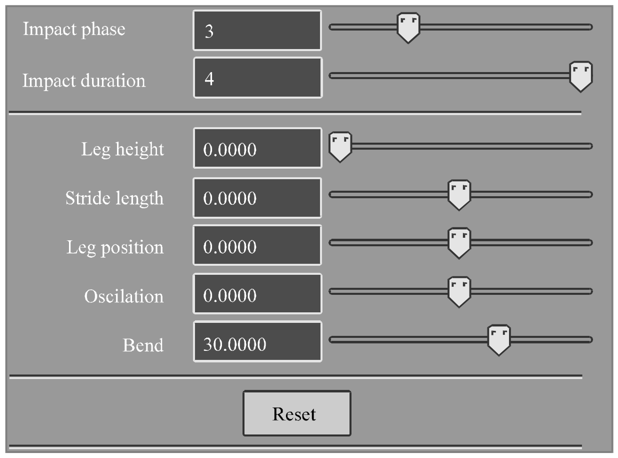
The parameters for the leg motion are shown in Fig. 12. Each leg is controlled individually by set of same parameters. All for legs have different values for different type of motion gaits. The leg impact phase and impact duration controls the main parameter between various different gaits. These attribute determines the time it takes for each leg to touch the ground and the duration for which each leg remains in contact with the ground.



Speed 3.0000
High -0.0257
Position 0.0000
Bounce 0.0050

Head high -0.0139
Head pos 0.0653
Head oscilation 0.0013
Tail swing 0.0500

Fig. 11(a-b): GUI for (a) Main body and (b) Head and tail parameters



Impact phase 3
Impact duration 4
Leg height 0.0000
Stride length 0.0000
Leg position 0.0000
Oscilation 0.0000
Bend 30.0000

Fig. 12: GUI for leg parameters

The results of changing the impact phase attribute for each leg; through the GUI, is shown in Fig. 13. The impact phase value (between 0 to 8), alters the timing of each stride impact and provides control in time shift. By modifying this Impact phase parameter of each leg individually, we control and generate various gaits of quadruped motion.

The Table 1 shows the Impact phase values of lion and horse character in walk and run gaits. As it can be observed that the walk motion of lion and horse is quite similar with very little difference in impact phase, however the run motion is some variable amount of differences in timing of leg impact.

Similarly the Fig. 14a shows the ghosting effect of leg height parameter when changed at runtime. When the leg height is set to 0 the feet stays at the ground but as the height parameter is increased the height or lift of leg during swing phase is increased, producing varying form of gaits.

The Fig. 14b shows the effect of stride length during motion. This is the key parameter that controls the length of each legs stride during motion.

The animation is created by modifying various attributes through GUI, according to the characteristic of each motion gait during animation. A preset library was developed containing values of impact phase, impact duration, leg height, Stride Length and position for multiple motion gaits i.e., walk, run, gallop. Figure 15a shows the phases of walk motion of lion in frame by frame mode, generated procedurally using our technique. Whereas Fig. 15b shows the walk phases of horse skeleton, again generated using the same technique.

Table 1: Leg impact phase (L_{ip}) for walk run motion comparison of lion and horse

	Front right leg		Front left leg		Hind right leg		Hind left leg	
Impact phase	Horse	Lion	Horse	Lion	Horse	Lion	Horse	Lion
Walk	3	3	7	7	1	2	5	6
Run	1	1	5	2	7	6	3	7

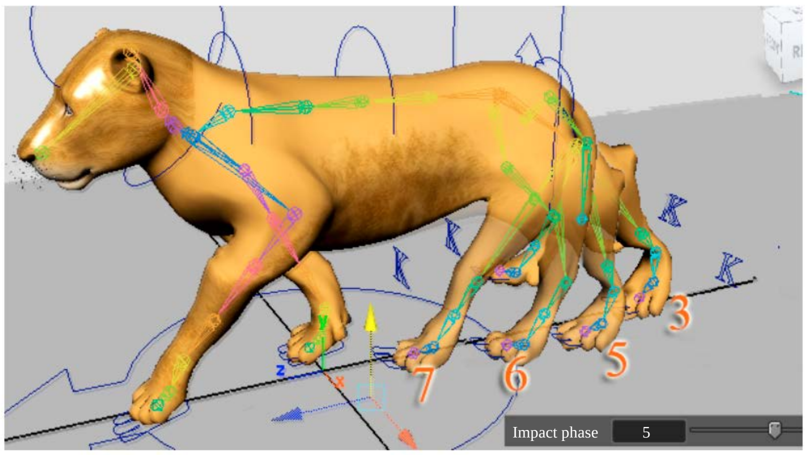


Fig. 13: Results of varying impact phase of left hind leg

Figure 16 shows the procedurally generated animation of (a) Lion and (b) Horse during run motion. For horse run motion the motion frequency was set to 4 with impact duration of

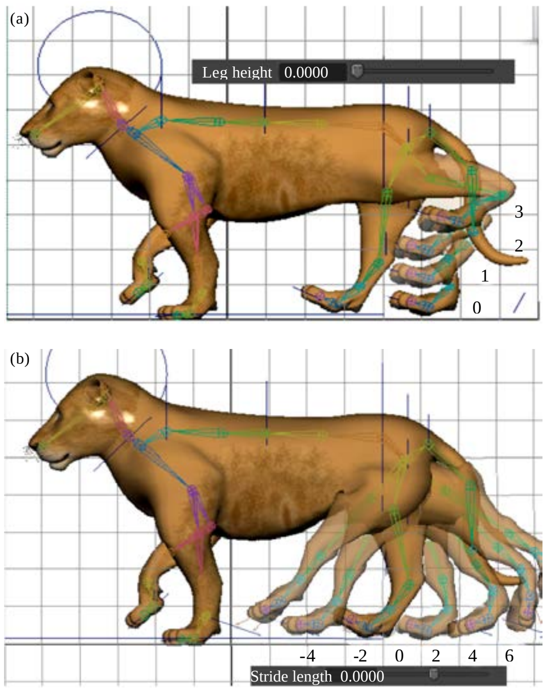


Fig. 14(a-b): (a) Front left leg and (b) Back left leg parameters

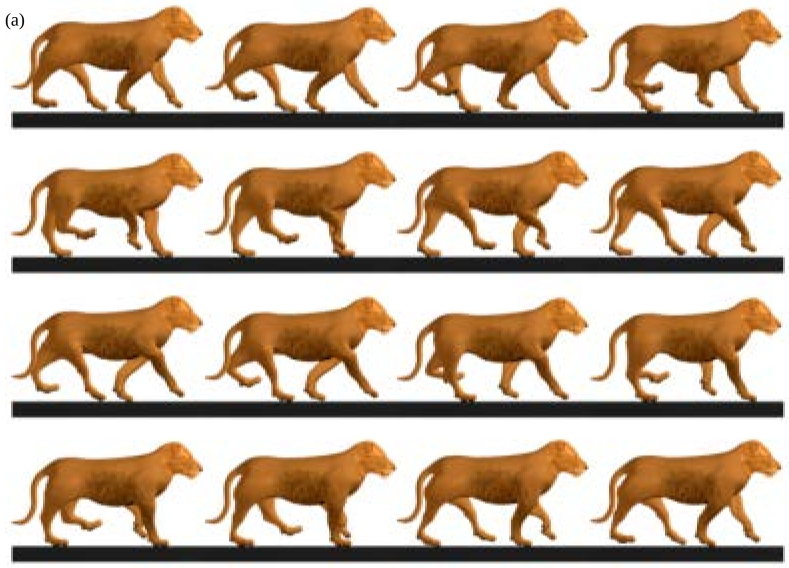


Fig. 15(a-b): Continue

(b)

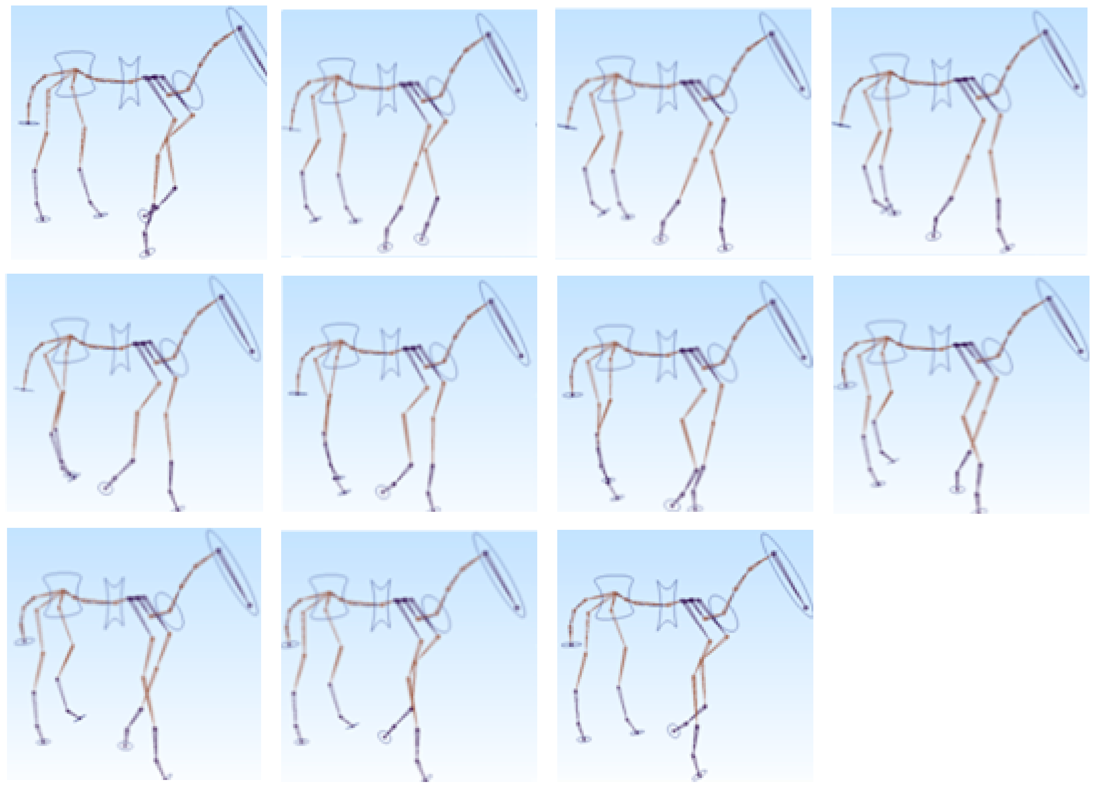


Fig. 15(a-b): Frame by frame procedurally generated (a) Lion walk motion and (b) Horse walk motion

(a)

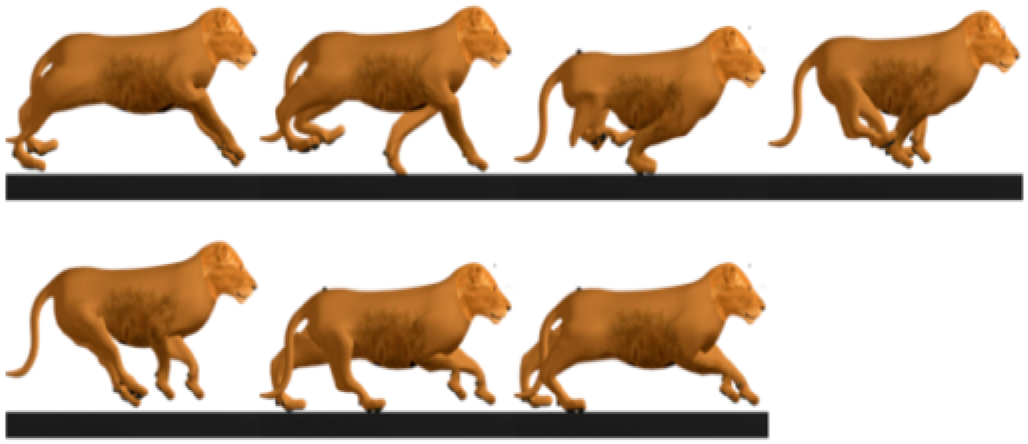


Fig. 16(a-b): Continue

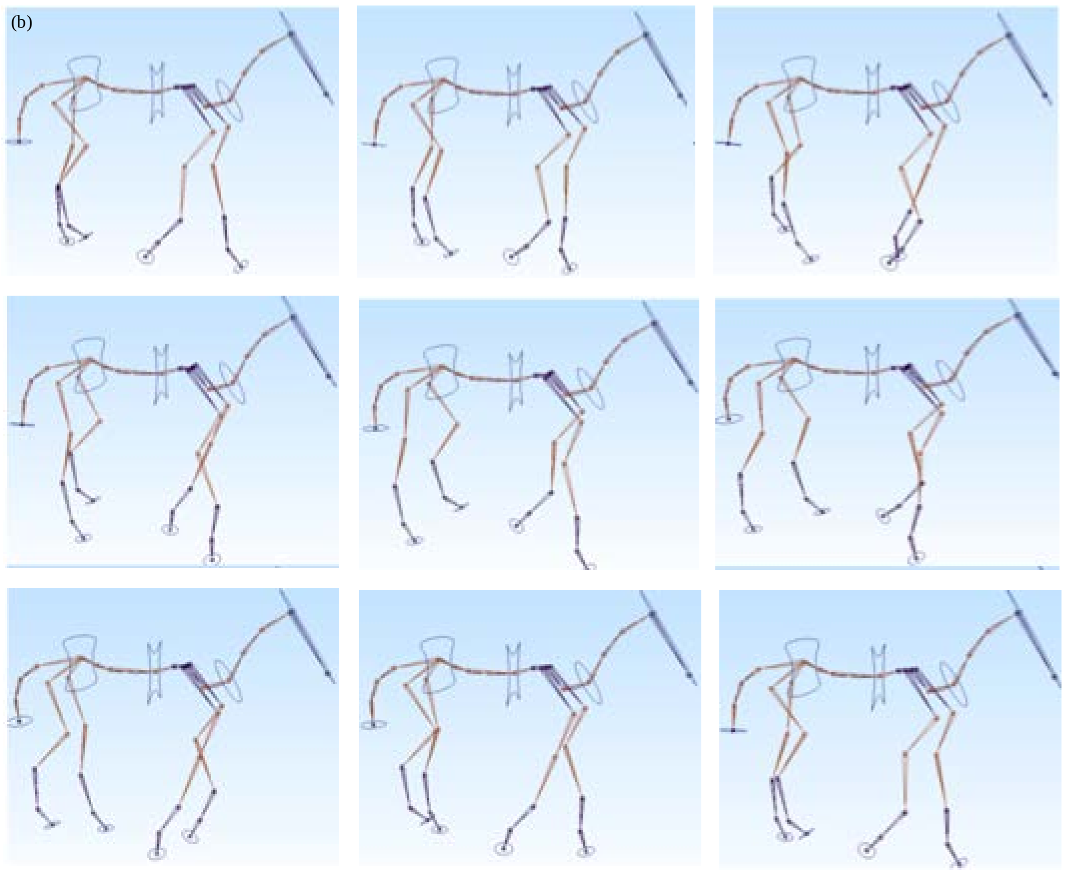


Fig. 16(a-b): (a) Frame by frame run motion of lion and (b) Amble motion of horse

each leg set to 3. The impact phase of each leg was set to 1 for front right leg, 5 for front left leg, 7 for back right leg and 3 for back left leg.

CONCLUSION

In this study a simple yet effective prototype technique of dual layer based animation model was discussed with base layer driven through procedural expressions using trigonometric functions and the top layer is controlled manually by custom user controlled manipulators. This work develops a widget based automatic rigging system which can be easily generalized to fit various other quadruped characters like Lion, Tiger, Dog, Zebra, etc. Through mathematics exact animation of character gaits, locomotion behavior and various motion processing aspects are directly incorporated into the control algorithms. The motion is then controlled by an easy to use animator friendly GUI with simplified attributes that internally effect the expression values driving the articulated character skeleton. Through this technique the process of animating quadrupeds is simplified and the base motion is generated easily. After the motion is achieved the animator then can easily grab the custom manipulators and modify the animation as per need and requirement. This approach will greatly enhance the quality and reduce the time consumed for creating animation from scratch.

This study needs to further improve the behavior and pattern of quadruped motion cycle. The control mechanism for terrain elevation and change in motion direction is also covered in this project which further needs to be researched and implemented. A rigorous testing mechanism has to be implemented to ensure the generated motion is absolutely accurate and realistic. More motion gaits have to be analyzed and incorporated into the algorithm with advanced set of custom control manipulators to produce high quality animation.

ACKNOWLEDGMENT

This study is partially funded by Ministry of Higher Education Malaysia (MOHE) under Commonwealth Scholarship and Fellowship Program, Ref: KPT.B.600-6/3 (BP3173731) 2012-2014.

REFERENCES

Allen, E. and K.L. Murdock, 2008. Body Language: Advanced 3D Character Rigging. 1st Edn., John Wiley and Sons, New York, USA., ISBN-13: 978-0470173879, Pages: 396.

Bhati, Z., A. Shah, A. Waqas and H.A.M. Malik, 2013. Template based procedural rigging of quadrupeds with custom manipulators. Proceedings of the International Conference on Advanced Computer Science Applications and Technologies, December 23-24, 2013, Kuching, Malaysia, pp: 259-264.

Bhatti, Z. and A. Shah, 2012. Widget based automated rigging of bipedal character with custom manipulators. Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, December 2-4, 2012, Singapore, pp: 337-340.

Bhatti, Z., A. Shah, M. Karabasi and W. Mahesar, 2013. Expression driven trigonometric based procedural animation of quadrupeds. Proceedings of the International Conference on Informatics and Creative Multimedia, September 4-6, 2013, Kuala Lumpur, pp: 104-109.

Coros, S., A. Karpathy, B. Jones, L. Reveret and M. van de Panne, 2011. Locomotion Skills for Simulated Quadrupeds. ACM SIGGRAPH, New York, USA.

Cureton, S.M., 2013. Using fourier analysis to generate believable gait patterns for virtual quadrupeds. Master's Thesis, Texas A and M University, Texas, USA.

Griffin, T.M., R.P. Main and C.T. Farley, 2004. Biomechanics of quadrupedal walking: How do four-legged animals achieve inverted pendulum-like movements? J. Exp. Biol., 207: 3545-3558.

Hecker, C., B. Raabe, R.W. Enslow, J. DeWeese, J. Maynard and K. van Prooijen, 2008. Real-time motion retargeting to highly varied user-created morphologies. ACM Trans. Graph., 27: 27-27.

Huang, T.C., Y.J. Huang and W.C. Lin, 2013. Real-time horse gait synthesis. Comput. Animat. Virtual Worlds, 24: 87-95.

Johansen, R.S., 2009. Automated semi procedural animation. Ph.D. Thesis, Aarhus University, Denmark, Europe.

Kokkevis, E., N.I. Badler and D. Metaxas, 1 995. Autonomous Animation and Control of Four-Legged Animals. University of Pennsylvania, Pennsylvania, USA.

Kwon, T. and S.Y. Shin, 2005. Motion modeling for on-line locomotion synthesis. Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, July 29-31, 2005, Los Angeles, CA., USA., pp: 29-31.

Liszewski, E., 2011. Gaits of the quadruped. http://emg-zine.com/item.php?id=779.

Marhefka, D.W., D.E. Orin, J.P. Schmiedeler and K.J. Waldron, 2003. Intelligent control of quadruped gallops. IEEE/ASME Trans. Mechatron., 8: 446-456.

Marsland, S.A. and R.J. Lapeer, 2005. Physics-based animation of a trotting horse in a virtual environment. Proceedings of the 9th International Conference on Information Visualisation, July 6-8, 2005, London, UK., pp: 398-403.

McLaughlin, T., 2012. Quadrupedal gait footfall patterns. http://www.viz.tamu.edu/research/pbanimation/data.shtml

Raibert, M.H., 1990. Trotting, pacing and bounding by a quadruped robot. J. Biomech., 23: 79-98.

Skrba, L., L. Reveret, F. Hetroy, M.P. Cani and C. O'Sullivan, 2008. Quadruped animation. Proceedings of the 29th Annual Conference of the European Association for Computer Graphics, April 14-18, 2008, Hersonissos, Greece, pp: 7-23.

Skrba, L., L. Reveret, F. Hetroy, M.P. Cani and C. O'Sullivan, 2009. Animating quadrupeds: Methods and applications. Comput. Graph. Forum, 28: 1541-1560.

Torkos, N. and M. van de Panne, 1998. Footprint-based quadruped motion synthesis. Graph. Interf., 98: 151-160.

Wampler, K. and Z. Popovi, 2009. Optimal gait and form for animal locomotion. ACM Trans. Graph., Vol. 28. 10.1145/1576246.1531366

Yamamoto, S., 2000. Effective implementations of multi-dimensional radix-2 FFT. Comput. Phys. Commun., 125: 1-7.