Construction of Animal Models and Motion Synthesis in 3D Virtual Environments using Image Sequences

Costas Panagiotakis and Georgios Tziritas
Department of Computer Science, University of Crete
P.O. Box 2208, Heraklion, Greece
E-mails: {*cpanag,tziritas*}@*csd.uoc.gr*

Abstract

In this paper, we describe a system that can build 3D animal models and synthesize animations in 3D virtual environments. The model is constructed by 2D images captured by specific views. The animation is synthesised by using physical motion models of the animal and tracking data from image sequences. Finally, the user selects some points of the 3D world and a smooth and safe motion path, which passes by these points, is created.

The main assumption of the 3D modelling is that the animal could be divided into parts whose normal sections are ellipses. Joints and angles between skeleton points are used in order to decrease models complexity. Using the above methodology, a snake, a lizard and a goat are reconstructed.

1. Introduction

1.1. Previous Work

The analysis of image sequences containing moving animals in order to create a 3D model of the animal and physical motion is a difficult problem because of unpredicted and complicated (most of the time) animal motion. In many cases, the construction of an accurate 3D model of an animal is a tough task because of its complicated shape. There are many techniques and methods in character modelling. We can distinguish them in canned animation from motion capture data and numerical procedural techniques such as Inverse Kinematics (IK) ([8]). Canned animations are more expressive but less interactive from numerical procedural techniques which often appear "robotic" and require parameter tweaking by hand. Motion capturing could be performed by means of image analysis. Ramanan and Forsyth [13] present a system that build appearance models of animals from image sequences. Grammalidis and Strintzis [5] proposed a method of head detection and tracking using multiview sequence, in which they model the head as 3-D ellipsoid estimated by least squares techniques. Apart from image analysis, many systems use direct motion capture technology. Dontchena, Yngve and Popovic [3] introduce an acting based animation system for creating and editing character animations at interactive speed using motion capture technology.

Until now, a lot of work has been done in 3D animal modelling. A.J. Ijspeert [6], [7] designed the 3D model of Salamander using neural controllers. Miller [12] simulates muscle contractions of snakes and worms by animating the spring tensions. The dynamical model includes directional friction due to the surface structure. Terzopoulos and Rabie [15] emulate the appearance, motion and behaviour of real fishes. Ferree and Lockery [4] derived a linear neural network model of the chemotaxis control circuit. The muscle tension is described by using Newtonian force equations. Wu and Popovic [16] describe a physics-based method for synthesis of bird flight animations. It is computed a set of wing-beats that enables a bird to follow the specified trajectory. The most recent works try to model the muscle system of the character getting realistic results. Albrecht, Haber and Seidel [1] present a deformation algorithm to create hand animations from images using muscle models. Seron, Rodriguez, Cerezo and Pina [14] present the definition and management of synthetic actors in real-time applications. They propose two types of nodes designed to move real-time synthetic actors and to integrate them within any animation system.

The basic purpose of this work is to create simple 3D models and animations of animals. The minimization of the model parameters without accuracy cost is an innovative part of our algorithm. The precision of image analysis data suffices to give realistic modelling.

We have animated a snake, a lizard and a goat (agrimi [1]). Finally, we add them in the virtual environment. We use digital images in order to construct the 3D model and to synthesize the 3D motion sequence of the animal. We have used algorithms from digital image analysis and signal processing. In the next section we are going to present the proposed methodology.

[1] Agrimi is a type of goat that lives in the Samaria of Crete.

Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'04)
0-7695-2223-8/04 $ 20.00 IEEE

2. Methodology

2.1. Overview

In this section we examine the main methodology of creating 3D animal animations. The overal method which compute the 3D animation of an animal, could be divided in the following steps.

- Creation of a 3D Animal Model
- Motion Synthesis
- Path Planning in Virtual Enviroment
- Motion Transformation in Virtual Enviroment

In the first step the 3D animal model is produced using segmented images created by background subtraction method. The 3D model is defined by some points (major points) like joint points and skeleton points. In the second step these major points are tracked in the image sequence. Finally, the 3D motion sequence is generated by motion tracking data and physical model. In the next step the user gives some points of the 3D world and a smooth and safe motion path is generated that passes from these points. The motion sequence is transformed to follow this path using a prototype motion transformation.

2.2. Creation of a 3D Animal Model

In this section we describe the first step of our method, the 3D animal model construction. The perfect 3D reconstruction using image analysis is a difficult task because of noisy image data and the animal's complicated shape. We use the following model in order to reduce image noise effects and to minimize parameters without accuracy cost. The animal could be divided into parts whose normal sections are ellipses. For example, the 3D-snake model is trivial, since it is composed of only one part.

The skeleton points are defined below. Let as define the function $d(p)$ (p is a snake point) as the Euclidean distance of p from the snakeskin. Then the snake skeleton could be defined as the set of snake points where the $d(p)$ has local maximum (Figure 1). The trace between the plane, which is vertical with the snake direction in a skeleton point, and the snakeskin could be approximated by an ellipse. So, the set of these ellipses could be an accurate 3D model (Figure 1, 5). These ellipses can be computed using the skeleton points p (ellipse center), the $d(p)$ and the eccentricity of the ellipse.

The 3D model is constructed from the animal image and its background image (Figure 2). In the first step, the skeleton points and the distance of each of them from the snakeskin are computed. The snake border is extracted using the Background Subtraction method (Figure 3). Bobick, Liu and Ivanov [2] represent a lighting independent background subtraction method. In our implementation we just subtract the animal image from its background image, then we use a threshold in order to compute the animal border (Figure 3). A constant number of skeleton points are computed using the border. An original iterative method is developed, whose input is the number of output skeleton points and its output is the skeleton points (Figure 4), having the next property: The Euclidean distance between two sequentially skeleton points is the same for every pair of sequentially skeleton points. Finally, for every skeleton point p is trivial to compute the $d(p)$.

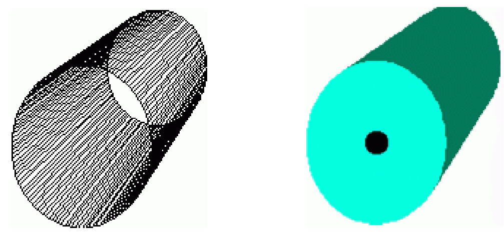

Figure 1: The 3D model for a snake part. The black point correspond to the skeleton point.

Figure 2: A snake image and its background image.

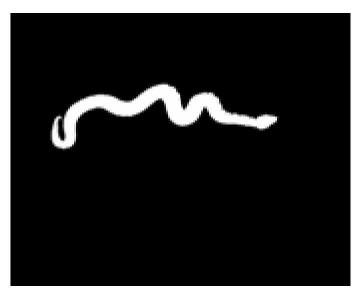

Figure 3: A snake bitmap.

2.3. Motion Synthesis

The motion synthesis algorithm uses as input the image tracking data. The 3D models are defined by skeleton

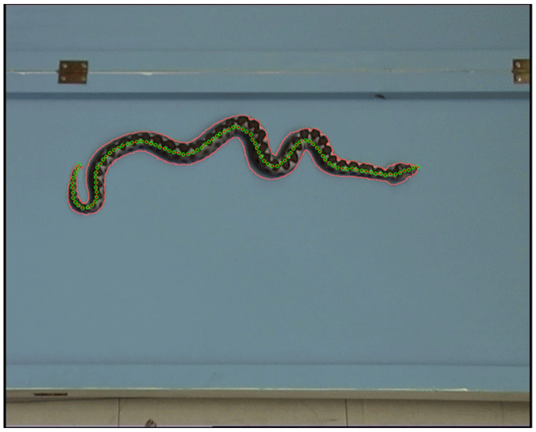

Figure 4: The segmented snake, the skeleton points (green points) and the snake border (red line).

Figure 5: The wireframe of a 3D snake model.

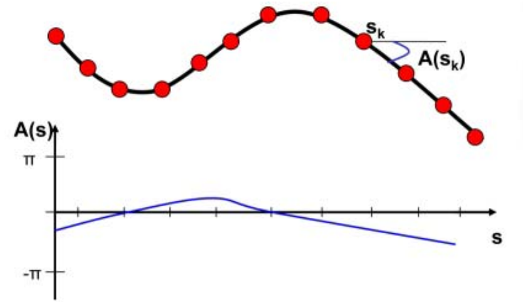

Figure 6: An example of skeleton points and the function A(s) computed in these points.

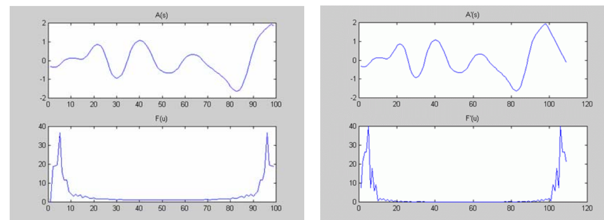

Figure 7: The $A(s)$ and $F(u)$ on the left, the $\hat{A}(s)$ and $\hat{F}(u)$ on the right.

points. In the case of articulated motion, like leg motion, the motion parameters could be reduced to joints points, because the skeleton points can be computed by joints points, using linear interpolation between two sequentially joints.

2.3.1 Non articulated motion

First, we analyze the case of non articulated motion. Examples of that motion are the snake motion and the lizard body motion. Using the above method we can compute the snake skeleton points in every frame of the image sequence. The sequence contains limited number of frames, so, it is important to develop a synthetic motion method. Let $A(s_k)$ be the skeleton angle computed in skeleton point s_k (Figure 6). This is an efficient representation because the distance r of each pair of sequentially skeleton points is constant. Consequently, the skeleton points reconstruction can be done by using just $A(s_k)$ and r.

Let $F(u)$ be the Fast Fourier Transform of the $A(s)$ sequence. The repetition of $A(s)$ may not give a periodic signal (Figure 7 on the left). The periodicity can be achieved by adding some samples at the end of $A(s)$ sequence which can be computed as the interpolation of the last and first term of $A(s)$. Let $\hat{A}(s)$ be the signal $A(s)$ after expansion and $\hat{F}(u)$ be the Fast Fourier Transform of sequence $\hat{A}(s)$ (Figure 7 on the right).

Because of snake shape, it can be proven that the first 8 FFT coefficients could reconstruct the $A(s)$ sequence. The first 8 FFT coefficients of $F\hat{(u)}$ contain the 99% of the signal energy (Figure 7 on the right), while the first 8 FFT coefficients of $F(u)$ contain only the 94% of signal energy (Figure 7 on the left). The first FFT coefficient gives the snake rotation, so we can ignore it. Let $C(u)$ be the first 7 FFT coefficients of $\hat{F}(u)$ ignoring the first one.

The advantages of this signal representation are the following. It is independent of snake translation, rotation and scaling. Also the number of parameters is minimal, just 7 complex numbers instead of 100 angles of $A(s)$. Also the signal noise is decreasing in reconstruction because just the low frequency band are passed. The same model can be used for lizard body.

Let $A(t,s)$ be the $A(s)$ at time t and $C(t,u)$ be the first 8 FFT coefficients of $A(s)$ at time t. Every sample $C(u)$ has its symmetrical $C_s(u)$ (equation (1)). Let $C_n(u)$ be the set of these samples.

$$|C_s(u)| = |C(u)| \quad and \quad \angle C_s(u) = -\angle C(u) \tag{1}$$

A set of random states of skeleton $R_k(u)$ can be generated by using the set of $C_n(u)$ samples (equations (2), (3), (4), (5)). The r, g are random numbers in $[0,1]$.

$$C_{min}(u) = \min_{n}(|C_n(u)|) \quad (2)$$

$$C_{max}(u) = \max_{n}(|C_n(u)|) \quad (3)$$

$$|R_k(u)| = C_{min}(u) + r \cdot (C_{max}(u) - C_{min}(u)) \quad (4)$$

$$\angle R_k(u) = 2\pi \cdot g \quad (5)$$

A states graph is created by connecting close states of skeleton. The graph is Probabilistic Roadmap ([10],[11]). Let $[X_c(t), Y_c(t)]$ be the position of the center of the snake mass. We model its motion using cubic splines. The velocity vector of $[X_c(t), Y_c(t)]$ gives the head snake direction ($\angle(\overrightarrow{s_1 s_0})$). A motion sequence could be produced by starting from a state (node) of the graph and selecting randomly the next states using the graph edges. In the state selection process we can use the following criterion. Let s be a state and $E(s)$ be the number of edges that pass from s. The unvisited states and the states that have many edges ($E(s)$) are selected. The maximization of $W(s)$ (equation (6)) gives the next state s and the n show how many times the state s has been visited. So, unvisited states are chosen making the motion more unpredictable.

$$W(s) = \frac{1}{2^n} \cdot E(s) \quad (6)$$

The median states between two states could be produced by using linear interpolation between the FFT coefficients. The produced snake motion gives the motion sequence of water snakes. The motion of the snake that lives in land is characterized by the following rule. The tail motion follows the head motion with a phase delay. Let $A(t, s)$ be the angle of the s point of the skeleton at time t. $T(t, 0)$ corresponds to the end point of the head. It is observed that the snake tail motion follows the head motion, this means that the snakes motion is characterized by the following rule $A(t, s) \approx A(t-1, s-1)$. Let $A_c(t, s)$ be the computed angle by our method (Figure 8). The following algorithm is used in order to convert the $A_c(t, s)$ to real snake motion. Let $A_f(t, s)$ be the output of the following algorithm (equation (8)) (Figure 8). The function $m(s)$ (eq. (7)) returns a real value in range of $[0, 1]$ for the skeleton point s according to its distance from the end of head. It returns one, If the point s is close to end of the head, else if the point s is close to end of the tail it returns zero. The constant c determines the skeleton point p from which on the rule $A_f(t, s) \approx A_f(t-1, s-1)$, $s > p$ holds, N is the number of skeleton points. The c can be changed randomly over the time ($c \le N$).

$$m(s) = 1 - e^{-(c \cdot \frac{s}{N})^6} \quad (7)$$

$$A_f(t, s) = m(s) \cdot A_c(t, s) + (1 - m(s)) \cdot A_f(t-1, s-1) \quad (8)$$

The snake tongue model is described below. The snake tongue is modelled by a cylinder whose height varies over time. The direction of the snake tongue is almost the same as the snake head. Let $H_{sb}(t)$ be the height of the cylinder at time t. We used image tracking data in order to estimate the probability $P(H_{sb}(t) = x / H_{sb}(t-1) = y,\ H_{sb}(t-2) = z)$. If this probability is known, it is trivial to compute a sequence of $H_{sb}(t)$.

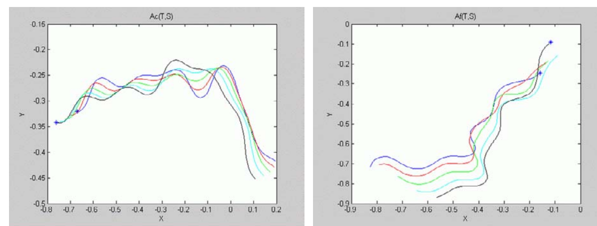

Figure 8: Sequentially skeleton positions computed by $A_c(t, s)$ and $A_f(t, s)$. The first state is the blue skeleton and the last one is the black skeleton. The * point corresponds to head.

2.3.2 Articulated motion

Figure 9: Two frames of a goat motion sequence.

In this section we analyse the articulated motion synthesis. We have implemented our methodology in lizard and goat motion synthesis. In this type of motion we have to compute the order of motion of the legs. The legs 1, 4 are stable and the legs 3, 2 are moving on the left image of

Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'04)
0-7695-2223-8/04 $ 20.00 IEEE

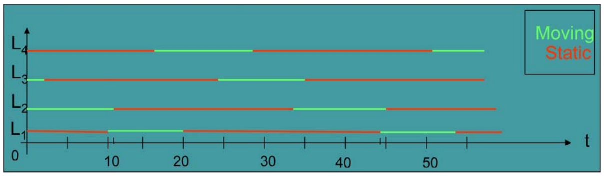

Figure 10: The red lines corresponds to static legs time periods, while the green line corresponds to moving legs time periods.

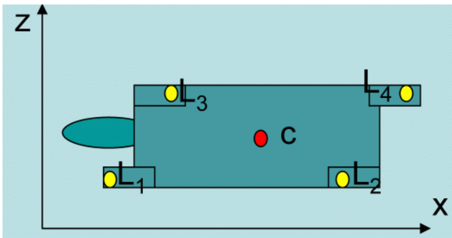

Figure 11: The goat legs model.

Figure 9. The legs 2, 3, 4 are stable and the leg 1 is moving on the right image of Figure 10. The table of Figure 10, where the legs states L_1, L_2, L_3, L_4 (moving or static) are shown over the time (in frames), has been computed using the whole image sequence. Let $L_i(t)$ be 0 if leg i is static at time t and 1 if leg i is moving at time t. The legs motion is periodic with period T. The following equations are derived by the data of Figure 10. $L_3(t) = L_1(t - \frac{T}{2})$, $L_4(t) = L_2(t - \frac{T}{2})$. The minimum and maximum values of two joints angles per leg can be estimated by video data, these values represent the motion limits. The mathematical model of motion is described below.

The animal will be moved along the X axis, in XZ plane. Let $C(t)$ be the X coordinate of the body mass center at time t, let $X_i(t)$ be the X coordinate of end of leg i at time t. The best balance is achieved if $c(t) = E(X_i(t))$. Let $u(t)$ be the velocity of mass center (C) and $v_i(t)$ be the velocity of leg i. Let $N(t)$ be the number of static legs at time t.

$$X_i(t) = \begin{cases} X_i(t-1) & , \textit{leg i static} \\ X_i(t-1) + u(t) + v_i(t) & , \textit{else} \end{cases} \tag{9}$$

$$\sum_{i=1}^{4} v_i(t) = N(t) \cdot u(t) \tag{10}$$

A static leg at time $t-1$ will be static at time t, if it satisfies the conditions derived by the motion limits. Else the leg at time t will be moving. The rule is the same for the moving case.

2.4. Path Planning in Virtual Enviroment

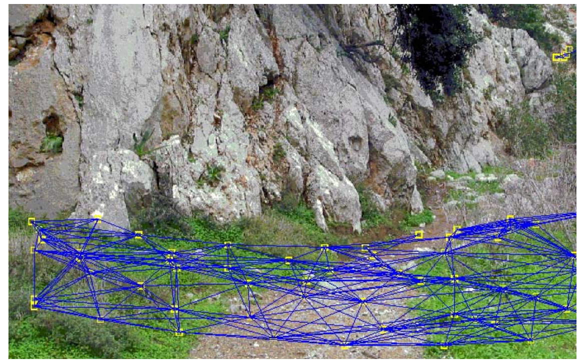

Figure 12: The 3D points graph.

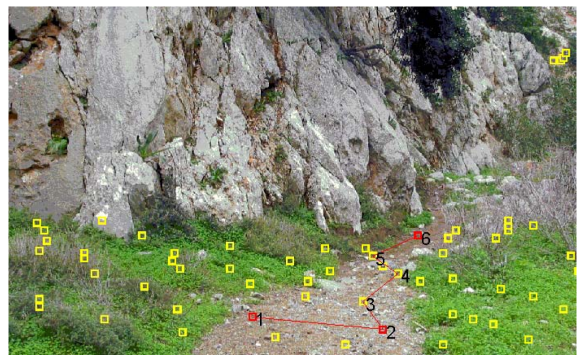

Figure 13: The final path in virtual environment.

The virtual world consists of 3D points. The animal is moved in the graph that is created by a subset of these points (Figure 12). The user gives a sequence of points $(\{U_1, \cdots, U_l\})$. The path planner creates a safe, smooth and short path in the graph that passes from user defined points. The meaning of the safe path is that the animal should avoid obstacles and holes. In the Figure 13 the final path is shown, the user gave the red points (1, 2, 6) and the method computes a path that passes from these points.

Now we are going to examine the method of path computation. Let $W = \{K_1, \cdots, K_n\}$ $(K_i \in \Re^3)$ be a set of 3D points of the virtual environment. Each point K_i can be written as $K_i = [X(K_i), Y(K_i), Z(K_i)]^T$. Let L_a be the animal length. The set S $(S \subset W)$ contains the points of the world in which the animal can stand safely (equation 11). The constants $c_1 \in (1, 2), c_2 \in (0, 0.5)$ depend on the animal.

$$P_i \in S \Leftrightarrow E_{|P_i - K_j|_2 < c_1 \cdot L_a}(|Y(P_i) - Y(K_j)|) < c_2 \cdot L_a \tag{11}$$

Many points of set S are very close each other, so, the following procedure is executed in order to decrease $|S|$. The method subtracts points from set S until the distance between any pair of points $P_i \in S, P_j \in S$ exceeds a threshold. The threshold is proportional to L_a. The animal can stand safely in every point of S. The next step is the computation of graph $G = (S, E)$. The edges correspond to the paths that the animal can follow. Let $D(p, ab)$ be the minimum distance of point p from the segment ab. The following equations (12), (14), (13) define how the edges of G are created.

$$Y_m(P_i, P_j) = \max_{D(K_g, P_iP_j) < L_a} (Y(K_g)) \tag{12}$$

$$W(P_i, P_j) = \sum_{D(K_g, P_iP_j) < L_a} |Y(K_g) - Y_m(P_i, P_j)| \tag{13}$$

$$P_i \sim P_j \Leftrightarrow W(P_i, P_j) < c_3 \cdot L_a \tag{14}$$

Let $W(P_i, P_j)$ be the edge P_iP_j weight. The edge weight is the elevation variability along the edge. A path is safer as the sum of weights along the edges is decreasing. So, we use Dijkstra's algorithm to find the path between U_i and U_{i+1} with the minimum sum of edge weights.

2.5. Motion Transformation in Virtual Environment

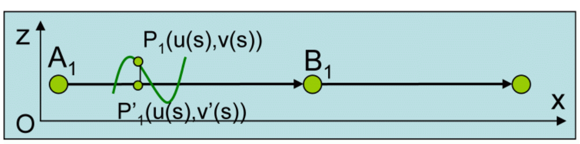

Figure 14: An example of animal moving in straight path.

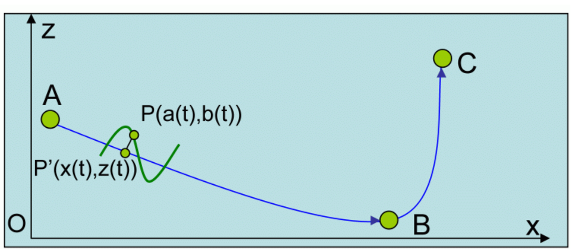

Figure 15: The transformed animation in real path.

The method of articulated animal animation, which is described in section 2.3.2, computes a prototype motion in straight path (Figure 14) of horizontal XZ plane. This motion is transformed in order to follow the real animation path (Figure 15), which is determined by the points sequence computed in section 2.4. We use cubic interpolation of these points sequence in order to get a smooth and continuous path (blue line of Figure 15). In the last step, we compute the Y coordinates of animation using the 3D model of virtual environment.

The mathematical formulation is described bellow. Moreover, the symbols are shown in Figures 14, 15. Let $u(s)$ be the initial path between points A_1 and B_1.

$$u(s) = A_1 + s \cdot (B_1 - A_1), \;\; s \in [0, 1] \tag{15}$$

Let $P_1(u(s), v(s))$ be a point of animal at time s. The projection of this point to path A_1B_1 is $P_1(u(s), v'(s))$, where the $v'(s)$ is the z coordinate of A_1 or B_1. Let A, B be the correspondence of points A_1, B_1 in real path $(c(t) = [x(t), z(t)]^T)$. The coordinates of A, B are known (see section 2.4). As it is described above, the real path $(c(t))$ is defined by a cubic spline (equation (16)).

$$[x(t), z(t)] = [\sum_{k=0}^{3} a_k t^k, \sum_{k=0}^{3} b_k t^k], \; t \in [0, 1] \tag{16}$$

The parameters a_k, b_k can be estimated by the coordinates of points A, B and the path direction in these points $(c(0) = A, c(1) = B)$. Let $c'(t) = [x'(t), z'(t)]^T$ be the first derivative of $c(t)$. The transformation of $u(s)$ to $c(t)$ is not linear so we have to correspond the s to t. The criterion is that the proportion of distance that the animal has covered at time t in real path should be s. This is defined by the equation (17).

$$\int_0^t |c'(u)| \mathrm{d}u = s \cdot \int_0^1 |c'(u)| \mathrm{d}u \tag{17}$$

Let $P(a(t), b(t))$ be the projection of P_1 in real path. Let $P'(x(t), z(t))$ be the projection of P in real path $c(t)$. The vector $w = [x'(t), z'(t)]^T$ shows the curve direction at time t. The vector $w_k = [-z'(t), x'(t)]^T$ is vertical to w. The point P is defined by the equation (19).

$$sign(x) = \begin{cases} 1 & , x \geq 0 \\ -1 & , x < 0 \end{cases} \tag{18}$$

$$P = P' + |P_1P_1'| \cdot sign(v(s) - v'(s)) \cdot \frac{w_k}{|w_k|} \tag{19}$$

Using the above equation, the animation of straight path is corresponded to animation of real path. In the last step, it is estimated the Y coordinates of motion according to the 3D virtual environment. We use a least mean square criterion in order to compute an approximated plane P_g of the

3D points of virtual environment that are close to animal position in XZ plane (local ground plane). The animal is rotated and translated so that the animal motion plane will be the estimated plane P_g. The rotation center is the mass center of the animal. The above procedure is repeated in every frame of motion sequence. On the left frame of Figure 18 the estimated plane P_g is almost horizontal while on the right frame the P_g grade is about 20 degrees. In both cases, the motion plane has adapted to local ground plane.

3. Results

Figure 16: Frames of snake animation.

Figure 17: Frames of lizard animation in virtual environment.

We have animated a snake, a lizard and a goat. Finally, the animals were placed in the 3D virtual environment of Samaria gorge. The Figures 16, 17 and 18 show some frames of these animations. In the Figure 16 we can see two frames of snake animation. As far as the motion synthesis algorithm is concerned the more unvisited and close states are selected. Consequently, the snake motion algorithm passes from many states producing smooth and unpredictable motion, which are the major characteristics of the realistic snake motion. In the bottom left of Figure 17 the lizard has just turned and it is moving straight, while, in the top and in the bottom right of the same figure the lizard is now trying to turn. In the Figure 18, the goat has adopted to local ground plane.

The method was implemented in C and Matlab. The results are extracted in VRML 2.0. The computation time for the creation of 3D animal model using 1.5 megapixels images was 26 seconds. This step is executed only once. The next steps produce the animation. The computation time for the motion synthesis of 300 frames, path planning in virtual world of 10.000 nodes and motion transformation in virtual world was 22 seconds. The animation duration was about a minute. So, our method can be implemented in real time. For our experiments, we use Pentium 4 at 2.8 Ghz.

4. Discussion

In this paper, we have presented a method for 3D animal model building and animation synthesizing using image sequences as input. The 3D model is created by ellipses whose centres are defined by the skeleton points. These points are tracked with pixel accuracy in the whole image sequence. The tracking accuracy has been improved by using stable camera and high contrast background. The 3D model parameters of articulated motion are the animal joints. In the case of non articulated parts, we reduce the noise of image data and the 3D model parameters by using the FFT coefficients of the angles sequence between sequentially skeleton points. The path planner creates a safe, smooth and short path in the 3D virtual environment that passes from specific points defined by the user. An original non linear motion transformation method has been developed in order to transform a straight motion in any curved motion. By experiments, we can see that if the maximum curvature of the real path is lower than the maximum curvature that the animal body can achieve, then the transformed motion result will be realistic. This is done because the real path curvature is transferred to the body curvature by the

Figure 18: Frames of agrimi animation in virtual environment.

motion transformation algorithm.

The real time implementation of this system will be one possible extension. In this system, each animal can act into the 3D environment according to the position of other animals, including the position of a virtually moved user. So, the animal interact with other creatures, trying to avoid or track the human observer and the other animals. Extending the above methodology, we can reconstruct more animals.

5. Acknowledgements

This work was supported by DXH project.

References

[1] I. Albrecht, J. Haber and H. Seidel. Construction and Animation of anatomically Based Human Hands Models. SIGGRAPH Symposium on Computer Animation., 2003

[2] A. Bobick, J. Liu, Yuri Ivanov. Lighting Independent Background Subtraction. International Journal of Computer Vision., 1998

[3] M. Dontchena, G. Yngve, Z. Popovic. Layered Acting For Character Animation. *Proccedings of the IEEE Int. Conf. on Computer Vision (ICCV 2003)*, pp. 409-416, 2003

[4] T. Ferree and S. Lockery. Computational rules for chemotaxis in the nematode C. elegans. *Journal of Computanional Neuroscience*, vol. 6, pp. 263-277, 1999

[5] N. Grammalidis and M. Strintzis. Head Detection and Tracking by 2-D and 3-D Ellipsoid Fitting. *IEEE Computer Graphics International Conference (CGI'00)*, Geneva, Switzerland, 2000

[6] A.J. Ijspeert. A 3D biomechanical model of the salamander. *http://rana.usc.edu:8376/ ijspeert/.*

[7] A.J. Ijspeert. Design of artificial neural oscillatory circuits for the control of lamprey- and salamander-like locomotion using evolutionary algorithms. PhD thesis. Department of Artificial Intelligence, University of Edinburgh, 1998

[8] M.P. Johnson. Exploiting Quaternions to Support Expressive Interactive Character Motion. PhD thesis. Massachusetts Institute of Technology, 2003

[9] K. Liu and Z. Popovic. Synthesis of Complex Dynamic Character Motion from Simple Animations.

[10] L. Kavraki, P. Svestka, J. Latombe and M. Overmars, Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces. IEEE Transactions on Robotics and Automation. *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566-580, 1996

[11] L. Kavraki, S. LaValle and J. Yakey. A Probabilistic Roadmap Approach for Systems with Closed Kinematic Chains. *Proccedings of the IEEE Int. Conf. on Robotics and Automation*, 1999

[12] G. Miller. The Motion Dynamics of Snakes and Worms. *Computer Graphics*, vol. 22, no. 4, pp. 169-178, Aug. 1988

[13] D. Ramanan and D.A. Forsyth. Using Temporal Coherence to Build Models of Animals. *Proccedings of the IEEE Int. Conf. on Computer Vision (ICCV 2003)*, vol. 2, 2003

[14] F. Seron, R. Rodriguez, E. Cerezo and A. Pina. Adding Support for High-Level Sekeltal Animation. *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 4, pp. 360-372, 2002

[15] D. Terzopoulos and T.F. Rabie. Animat Vision: Active Vision in Artivicial Animals. *Journal of Computer Vision Research*, vol. 1, no. 1, pp. 2-19, 1997

[16] J. Wu and Z. Popovic. Realistic Modeling of Bird Flight Animations. *ACM Transactions on Graphics (SIGGRAPH)*, 2003

[17] A. Zisserman, A. Fitgibbon and G. Cross. VHS to VRML: 3D Graphical Models from Video Sequences. *IEEE International Conference on Multimedia Computing and Systems*, vol. 1, 1999