A PROPOSED APPROACH FOR 3-D ANIMATIOn GENERATION BASED ON MOTION CAPTURE DATABASE

Nowadays 3-D animated human models are widely used in the computer games, films, systems for modeling sports, dance and other areas. Movements of virtual humans are required to seem realistic, but animation, which is modeled with standard mathematical functions and made by computer, doesn’t provide realistic results. So there is a need to improve animation of movements and make them more realistic and natural.

An approach, which is proposed in this work, is based on human movement database, obtained with the motion capture technology. A large database of human movements is collected, then a neural network is trained on this dataset and then performs a generation of human movement function. A brief review of related works is presented below. The review includes articles with description of similar self-trained methods, which apply various statistical models, such as PCA or HMM, and also articles, related to neural networks, motion capture data processing and algorithms for solution of the inverse kinematics task.

General plan

1. ~~Introduction. Formulation of the problem~~
2. Solutions for animation generation. Things to pay attention to
   1. Comparison with the proposed approach
3. Compilation of previous methods, suitable for proposed approach
4. Proposed approach

Motion capture data: data handling and retrieval

Animation with motion capture

1. Inverse kinematics approach to animation simulation
2. Neural network approach to animation simulation
3. Statistical approach to animation simulation

Inverse kinematics related tasks

Pure neural networks related tasks

Things to pay attention to in animation generation solutions.

Tasks:

1. To handle the motion capture data – form an appropriate dataset, suitable for neural network
2. Build a neural network
3. Train neural network

Being critical 10/10

Being cautious 10/10

Classify 5/5

Compare 4/5

* 1. Real time animation generation from mocap data

Motion capture data needs to be collected, proceeded and handled for further use in any computer graphics system. Motion capture markers can provide only small amount of information about 3-D model itself, so it needs to be additionally generated from mocap data markers. It’s not simple an obvious task, for example, system sometimes isn’t able to differ markers from right and left hands and etc[2]. In this paper[2] a system called MocapRender for real-time processing motion capture data as virtual 3-D models is presented. Motion capture system, which has been described in the article, retrieves data in form of a set of body markers. The system of the real-time mocap rendering to virtual 3-D models has distributed architecture and can handle information from any motion capture device. Since the MocapRender is flexible, it’s impossible to bind the whole mesh to whole human body, and so body is divided into parts. Also MocapRender applies an estimation to retrieved mocap markers, comparing them with desired markers’ positions. However, models, which are generated via this system, have no skeletons. MocapRender can be fast, but is inconvenient for application to tasks, which require skeleton data.

* 1. Motion capture assisted animation: texturing and synthesis

This paper[3] describes a method of animated movements generation via motion capture. The method starts from the idea, that keyframed animation looks unnatural and not realistic. Keyframed data based on e.g. key frames – the consecutive snapshots of human movement. In the article two plots of keyframed data function and motion capture function are presented for comparison: keyframed function looks like more formal and repeating, while function, obtained from motion capture, has more individual character, which cannot be simply described via one or two formulas.

The basic idea of the method is matching keyframed data and motion capture data in order to obtain more precise results and more realistic animation synthesis. A dataset is divided into matching regions of motion capture data and keyframed data. Then the region with the highest variety is chosen. Then this regions is divided into fragments in the points, where the first derivative of this region’s function changes a sign. Both for the fragments of mocap and keyframed data and their derivatives respectively a comparison and matching process is performed. Then the matching matrix can be obtained. Matching is necessary to define the similarity of the data fragments, and then their conjunction to one function. As there are several suitable ways to form a consecutive realistic movement function, the best (or the first best) is chosen, basing on the neighbor weight matrixes of every fragment. Fragments join via selecting the middle points of the function in the gap and then joining edges of the fragments with this point.

The method is good, but, however, constant joining of the fragments and mixing fragments can cause errors in data. Search of the optimal path and joining the fragments requires time and memory to store matrixes, so it’s quite inconvenient for high-loaded computer games or databases with the large dataset.

* 1. Realistic synthesis of novel human movements from a database of motion capture examples

In [4] a generation of the animation is performed by self-learned algorithm. Firstly, the motion capture database is collected. A dataset presented as the set of joints and rotation angles. This is called angle-axis representation, and it has been applied because of Euler angles model’s disability to compute several parameters with. Then the K-means clustering algorithm is applied to denote different clusters of joints. A learning algorithm itself is based on two main parts: a Markov chain level and Hidden Markov Model level. Each state of the Markov chain denotes cluster of joints. A learning process is presented by counting probabilities of transitions between the states of Markov chain. A second level contains motion examples and complements the states of the first level with probabilistic estimation of every example and it’s attachment to the certain cluster via Bayesian formula. After the model is built, the motion synthesis is performed between two keyframes, as in [1]. The path, which consists of the transitions between Markov chain states is found by application of the dynamic programming methods, what is similar to the path finding task in [3].

* 1. Physically-valid statistical models for human motion generation

This paper[5] similar to [4] and [3] and based on the idea of trained statistical models for animation generation. It’s also has statistical model, which is trained on motion examples, but the trained model also considers various physical constraints while learning on examples. The authors suppose, that the set of physical constraints will make the generated animation more realistic, and model will become more adaptive to various physical conditions, which can appear in real live. So the main feature of the proposed model is that it consciously considers physical conditions, which causes different movements. Kinematic motion is generated via non-linear probabilistic function, which modeled with Gaussian process. Physical constrains of the model are represented by a set of special equations. Several experiments, such as walking with the heavy shoe are presented.

* 1. PCA-based walking engine using motion capture data

In this paper[6] another system for human movement generation is presented. It uses marker motion capture system to retrieve the data and then a skeleton representation is formed from the marked data array. A statistical model called PCA is applied to generation of movements. PCA is able to transform a set of variables to their linear combination, so so as to account for most of the variance of an original dataset. Therefore, PCA spaces composed of various walking motion sequences that only differ in speed values are established, leading to one space per subject[6]. The presented walking engine is driven by two high-level parameters: speed and human size. First, interpolation and extrapolation are achieved in the PCA spaces of each subject to generate new motion according to a speed value. Secondly, a time warping method allows to handle the human height parameter.

It’s not enough just to collect motion capture data: it is also needed to be handled, divided into sensitive parts and etc. The following articles describes methods, which can be applied to simplify the usage of motion capture dataset.

1.7 Phase-Functioned Neural Networks for Character Control

This article[21] proposes a novel approach of character’s motion generation. This approach is based on neural network, which generates new weights per each frame of a motion and provides adaptiveness of a model to various terrain surfaces. Learned on the motion capture database of motion in different terrains, the neural net is capable to predict the next frame of the character motion, depending on terrain and user control inputs. The set of neural network’s parameters also includes directions of character’s joints and environment parameters.

1.8 A Deep Learning Framework for Character Motion Synthesis and Editing.

This article[22] also proposes a generation method for smooth motions, based on terrain track. Because of lack of phase function and user control inputs influence it is different from [21] and can be applied only to a prepared sequence of steps, which character should perform. The neural network, which takes joint positions as parameters, is presented, but the conversion between high-level parameters and joint angles is required to retrieve a motion frame. But the plus is ability of the model to generate certain motion frame on demand.

1.9 Synthesizing Physically Realistic Human Motion in Low-Dimensional, Behavior-Specific Spaces

This paper[25] proposes a method of utilizing an existing motion capture database to find a low-dimensional space that captures the properties of the desired behavior. The approach implies the fact that joint movements in human motion are highly correlated, because of natural cohesion of human bones. The input data are a rough sketch of the desired motion and the constraints that should be enforced. The animator also selects the motions used to define the lowdimensional space for the desired motion. Constrained optimization is then used to automatically find a motion that minimizes some objective function subject to satisfying the user-specified and physics constraints. The SNOPT library is used to solve the non-linear optimization problem applying to described task.

1. MOTION CAPTURE RELATED TASKS
   1. Efficient content-based retrieval of motion capture data

This article[10] presents a method for sequential motion capture keyframe chains retrieval from a large dataset. In case if motion capture dataset is large, it can be hard to find an appropriate place to divide it. The authors introduce various kinds of features describing relations between special body points of a pose and the following performance of these features. Via these features consecutive poses are retrieved, and then the document database, where every document is a set of motion capture keyframes, is composed.

* 1. Keyframe extraction from human motion capture data based on a multiple population genetic algorithm

The keyframe extraction techniques applied to motion capture data extract representative frames from the motion data based on the entire movement sequence[11]. This article presents a human motion capture data keyframe extraction method based on a multiple population genetic algorithm (MPGA). An idea of genetic algorithm comes from natural evolution process, mutations and birth of new generations by mixing “chromosomes” (parts of the dataset on the current iteration). MPGA breaks with the standard genetic algorithm (SGA) framework that relies on the evolution of a single group of genes by introducing multiple populations to optimize the search process. In the algorithm the population is presented as motion sequence, and all frames in it are labeled “keyframe” and “not keyframe” labels. Then the algorithm evolves, according to proposed fitness function.

* 1. Entropy-based motion extraction for motion capture animation

In this paper[16] a new segmentation solution for extracting motion patterns from motion capture data by searching for critical keyposes or “keyframes” in the motion sequence is presented. The method is based on entropy metrics, specifically the mutual information measure.

Firstly keypose extraction problem is solved by using a mutual information method to identify critical keyposes from a given motion sequence. Keyposes determine a substantial directional change in the motion pattern. An entropy measure is used to quantify the amount of information that is passed from one frame to another. The algorithm ranks the keyposes based on their corresponding mutual information levels. These levels are related to the significance in the directional change of the motion.

2.4 Learning Motion Manifolds with Convolutional Autoencoders

In this paper[23] a problem of finding a sequences of data in the motion capture sets, which correspond to valid and biologically correct human motions. The approach applies convolution neural networks to learn projection and inverse projection operators to produce valid human motions. Learned models can be used for filling missing peaces of data, fixing corrupted data and motion interpolation to make the character perform smoother movements.

2.5 Learning an Inverse Rig Mapping for Character Animation

This article[24] proposes real-time solution for mapping character’s rig to its skeleton. The task is to map skeleton posing to corresponding mesh, basing on given rig pose postures. The approach is to take initial skeleton parameters and then apply inverse rig mapping and non-linear regression to generate an appropriate rig. Rig function’s derivative is applied to increase precisenesss of a method.

1. INVERSE KINEMATICS TASK

A dataset, which has been collected from motion capture sensors, needs to be handled in order to obtain functions of joints, where an argument is a time parameter and value is the angle, which every joint rotates to. Obtaining of the parameters, which are necessary to place the kinematic system to preferred position is called an inverse kinematics task. As it’s not ordinary task, it’s being still explored. Below several research works, describing various solutions, are presented.

* 1. Neural network based inverse kinematics solution for trajectory tracking of a robotic arm

This paper[8] presents inverse kinematics task solution, which uses neural network as a key mean to solve the problem for a robotic arm. A robotic arm itself is a kinematic system, and proposed approach can also be applied to another kinematic system which has similar parameters, for example, a leg or even entirely human body. The arm is presented by connected links and angles. The main idea of the approach is to learn the feed forward neural network with random joint data, which meets the constraints of forward-kinematics equations. The process of finding a solution to the inverse kinematics problem, in this case, can be seen as an inversion problem for neural networks. This means, finding the inputs (joint parameters) that yield a desired output.

* 1. Redundant inverse kinematics: experimental comparative review and two enhancements

This article[9] presents a review of inverse kinematics algorithms: Jacobian Pseudoinverse, Jacobian Transpose, Selective Damping and etc. An analysis of various characteristics of these algorithms is performed: convergence, numerical error, joint limit avoidance and etc. Also a few enhancements to existing algorithms were produced. The first is to filter the singular values of the Jacobian matrix before calculating its pseudoinverse in order to obtain a more numerically robust result. The second is to combine a continuous task priority strategy with selective damping to generate smoother trajectories. The enhancements performed good results.

* 1. Fabrik: a fast, iterative solver for the inverse kinematics problem

In this article[13] a completely novel method of inverse kinematic task solution is presented. The method called Forward And Backward Reaching Inverse Kinematics (FABRIK). FABRIK uses a forward and backward iterative approach, finding each joint position via locating a point on line. The proposed method starts from the last joint of the chain and works forwards, adjusting each joint along the way. Thereafter, it works backward in the same way, in order to complete a full iteration. This method, instead of using angle rotations, treats ﬁnding the joint locations as a problem of ﬁnding a point on a line; hence, time and computation can be saved. The graphical presentation of the algorithm is given and also models with multiple end effectors (joints with no children) are reviewed. Also algorithms is compared to another algorithms and as a result performs better.

* 1. Inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver

This paper is an extension to [13], where FABRIK solver is briefly described. This paper also include a wide and proper review of the existing inverse kinematics technologies, which include Jacobian inverse, such as the Jacobian Transpose, Damped Least Squares (DLS), Newton Method, Sequential Monte Carlo Method and Cyclic Coordinate Descent. However, many of these algorithms produce unrealistic solutions and that the motions of human joints are limited, implies that joint restrictions need to be added. Generally this paper gives a wider understanding of inverse kinematics task. Also several algorithms are compared to each other on parameters of poses naturalness, number of end effectors, computational cost and other.

* 1. A constrained inverse kinematics technique for realtime motion capture animation

In this paper[14] a constrained inverse kinematics algorithm for real-time motion capture in virtual environments is presented. It is an iterative algorithm for finding the solutions of the constrained equations of motion. It is based on a numerical algorithm called SHAKE. The algorithm is an iterative numerical integration scheme and is designed to reduce computation time by increasing the time step to solve constrained motions of rigid bodies with a large number of constraints. Algorithm considers every constraint and iterates over all the constraints until they are all within a given local tolerance and global tolerance is also satisfied.

* 1. A neuro-genetic approach to the inverse kinematics solution of robotic manipulators

In this paper[15], a neuro-genetic approach is proposed for the inverse kinematics problem solution of robotic manipulators. The proposed solution method is based on using neural networks and genetic algorithms in a hybrid system. Firstly, three Elman neural networks are trained for the inverse kinematic solution. The next step is genetic algorithm implementation, which considers generated dataset from neural network as next generations. The results of an experiment separately with neural networks and also with genetic algorithm are presented – the genetic algorithm shows accuracy up to seven or eight digits after comma.

* 1. Solvability-unconcerned inverse kinematics by the levenberg–marquardt method

In this paper[18] a robust numerical solution to the inverse kinematics is proposed based on the Levenberg–Marquardt (LM) method, where the squared norm of residual of the original equation with a small bias is used for the damping factor. This paper gives a novel feature changing this damping factor. Robustness and convergence performance are achieved by a rather simple method to use the squared norm of residual for the damping factor. A similar method was proposed earlier in several articles, which has a problem that the computation becomes unstable if the goal position and orientation of the link of interest is set near the singular points. The proposed method resolved this problem by slightly biasing the damping factor.

* 1. A novel application of artificial neural network for the solution of inverse kinematics controls of robotic manipulators

This paper[19] presents another neural network approach to inverse kinematics task solution, but it’s different from [8] and [15]. In this paper, a novel application of artificial neural network is used for controlling a robotic manipulator. The proposed methods are based on the establishments of the non-linear mapping between Cartesian and joint coordinates using multi layer perceptron and functional link artificial neural network. All relationships between joints are described by geometrical non-linear equations. A function for weight computation in neural network is adapted and modified, as the derivatives of an error function and some other parameters are added to the neural network model.

* 1. Efficient inverse kinematics computation based on reachability analysis

This work[20] shows how precomputed reachability information can be used to efficiently solve complex inverse kinematics (IK) problems such as bimanual grasping or re-grasping for humanoid robots. Authors present an integrated approach which generates collision-free IK solutions in cluttered environments while handling multiple potential grasping configurations for an object. Therefore, the spatial reachability of the robot's workspace is efficiently encoded by discretized data structures and sampling-based techniques are used to handle arbitrary kinematic chains. The algorithms are employed for single-handed and bimanual grasping tasks with fixed robot base position and methods are developed that allow to efficiently incorporate the search for suitable robot locations.

1. NEURAL NETWORKS

A neural network is represented as a set of neurons, where every neuron has an activation function, which is used in network training process. An activation function is necessary to compute the weight matrix, which defines a form of output data (in this particular case in animation modeling it’s a function of many joint angles parameters). Weights are need to be corrected, based on the input data, and this process presents a training of neural network.

Many real-world tasks in most cases is hard to be modeled with strict mathematical equations, and existing models often may not be accurate or suitable. As human movement cannot be presented by a strict mathematical function, there is a need to find or approximate it, basing on an input data, and it’s the task, which neural network can perform well on.

* 1. Function approximation using artificial neural networks

This article[7] describes a function approximation with neural network, basing on output and input data, e.g. argument values and function values. Various types of neural networks, such as RBFN (radial basis function network) and WNN (wavelet neural network), which perform well in function approximation, are overviewed and compared with the expected function via least square root method. As a result, WNN, which has Gaussian wavelet as a basis function, performed best in modeling of various function. This article includes necessary neural networks types review to determine, which function will perform best in the approximation task, which is required by the approach.

* 1. A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm

This article[12] applies a neural network to a regression task. Actually, prediction of the human movement function is also a task of regression, so it uses similar methods. A neural network, proposed in the paper, is the GRNN (generalized regression neural network), which is similar to RBFN (radial basis function network), which has already been mentioned as well-performed in [7], and is based on a standard statistical technique – kernel regression. The GRNN has excellent performances on approximation ability and learning speed, and it is fast learning and convergence to the optimal regression surface as the number of sample data becomes very large. The architecture of the GRNN is described. A GRNN has only one constant parameter, which is estimated by fruit fly algorithm, which is also described. Even though both GRNN with fruit fly are used for forecast, they are also can be applied to movement function prediction, and it’s expected to be quite accurate, as the systems as forecasting, require high accuracy.

4.3 Dropout: A Simple Way to Prevent Neural Networks from Overfitting

An approach to avoid overfitting of neural network, extremely important in deep learning.