

# Подход к параметризации движений в компьютерной анимации персонажа

Максим Сергеевич Ильиных  
Ижевский Государственный Технический Университет  
г. Ижевск, Россия

## Аннотация

В работе предложен подход к представлению параметризуемого движения в компьютерной анимации персонажа как совокупности элементарных параметризуемых составляющих. Рассмотрены основные принципы подхода, приведена используемая модель персонажа и его движения. Дано краткое описание созданной на основе подхода системы анимации персонажа: механизмов ее работы и функциональных возможностей.

**Ключевые слова:** компьютерная анимация персонажа, физическое моделирование, движение.

## ВВЕДЕНИЕ

Сегодня, компьютерная анимация персонажа широко используется для создания правдоподобного, органичного движения виртуальных актеров, способного передать их намерения, мысли и эмоции. Это обусловлено тем, что воспроизведение движения персонажа является самым сложным видом анимации, требующим большого мастерства и высокой точности. Использование компьютера позволяет автоматизировать этот процесс, давая аниматору возможность уделять больше внимания ходу сюжета, нежели нюансам движения частей тела персонажа.

Существующие методы компьютерной анимации персонажа либо требуют от пользователя глубоких знаний в дисциплинах, не относящихся непосредственно к анимации, либо существенно ограничивают его пространство возможных действий, не давая достигнуть желаемого результата [1]. С другой стороны, классический метод анимации по ключевым кадрам свободен от этих недостатков, но он плохо поддается автоматизации.

## ПРЕДСТАВЛЕНИЕ ДВИЖЕНИЯ ПЕРСОНАЖА

Суть предлагаемого подхода состоит в совмещении классического метода анимации с физическим моделированием. Это позволяет сократить число необходимых для определения анимации ключевых кадров за счет использования в межкадровой интерполяции, наряду со сплайнами, механических и биомеханических свойств тела персонажа. Ключевые кадры определяются специальными микропрограммами, названными элементарными движениями. Сложное движение персонажа представляется комбинацией последовательно и параллельно выполняемых элементарных.

Будем рассматривать персонажа как совокупность скелета и мышечной системы:

$$\text{Character} = \langle \text{Skeleton}, \text{MuscleSystem} \rangle$$

где *Skeleton* – скелет персонажа;

*MuscleSystem* – мышечная система персонажа.

Скелет персонажа представлен системой твердых тел (элементов скелета) и связей между ними (суставов) [2]:

$$\text{Skeleton} = \langle \{ \text{RigidBody} \}, \{ \text{Joint} \} \rangle,$$

где *RigidBody* – твердое тело;

*Joint* – связь.

Связь задает ограничения относительного движения двух соседних элементов скелета и определяется совокупностью векторов обобщенных координат, скоростей, ускорений и сил:

$$\text{Joint} = \langle p(t), p'(t), p''(t), u(t) \rangle,$$

где *t* – момент времени;

*p(t)* – вектор обобщенных координат;

*p'(t)* – вектор обобщенных скоростей;

*p''(t)* – вектор обобщенных ускорений;

*u(t)* – вектор обобщенных сил.

Состояние персонажа в некоторый момент времени можно определить как совокупность множества значений обобщенных координат каждого сустава скелета, мировых координат его основы и множества опорных точек:

$$\text{CharacterState}(t) = \langle X_{base}, \{ p(t) \}, \{ \text{SupportPoint} \} \rangle,$$

где *X<sub>base</sub>* – положение и ориентация в пространстве основы скелета персонажа – тела, относительно которого, через обобщенные координаты, определяется положение других элементов;

*SupportPoint* – опорная точка, представленная одной из вершин "логического скелета" персонажа, заданного в виде множества многогранников и определяющего возможность взаимодействия персонажа с окружающей средой.

Для каждой обобщенной координаты ключевое значение задается в виде пары следующего вида:

$$StatePair = \langle t, p(t) \rangle.$$

Тогда, элементарное движение персонажа может быть представлено функцией, которая, исходя из момента времени, состояния персонажа и параметров, определяет ключевые пары каждой обобщенной координаты для данного элементарного движения:

$$BasicMovement(t, CharacterState(t), \{Param\}) = \{StatePair\},$$

где  $Param$  – параметр движения, представленный именем, типом, областью определения и значением.

Применение сплайновой интерполяции к полученным ключевым парам дает множество всех значений, принимаемых обобщенной координатой скелета персонажа в ходе элементарного движения. Полученные таким образом траектории-сплайны в дальнейшем будут называться идеальными, то есть такими, какими они были бы при использовании классического метода анимации по ключевым кадрам.

Действительные траектории движения элементов скелета персонажа определяются на основе идеальных путем применения физического моделирования. Для этого используется динамика твердого тела [3] и биомеханика [4]. Согласно первой, осуществляется моделирование движения тела персонажа под действием внешних сил. В соответствии со второй, каждой обобщенной координате сопоставляется пара мышц:

$$MuscleSystem = \{MusclePair\},$$

$$MusclePair = \langle JointRf, dofNum, PosMuscle, NegMuscle \rangle,$$

где  $MusclePair$  – мышечная пара;

$JointRf$  – ссылка на соответствующий мышце сустав скелета;

$dofNum$  – номер изменяемой обобщенной координаты;

$PosMuscle = Muscle$  – положительно направленная мышца;

$NegMuscle = Muscle$  – отрицательно направленная мышца;

$Muscle$  – мышца.

В каждый момент времени  $t$ , одна из пары мышц является активной, то есть сообщает такую силу  $u(t)$ , которая дает ускорение  $p''(t)$ , необходимое для выполнения заданного элементарного движения, а другая – пассивной и действует как упругое тело. Если необходимо увеличить текущее значение обобщенной координаты, то активной становится положительно направленная мышца, в противном случае – отрицательная.

Мышца определяется совокупностью напряжения, скорости напряжения, максимальной сообщаемой силы и силой упругости:

$$Muscle = \langle StrainSpd(t), Strain(t), VFC(p'), StrainF(p), RelaxF(p) \rangle,$$

где  $StrainSpd(t)$  – скорость напряжения мышцы;

$Strain(t) \in [0, 1]$  – текущее напряжение мышцы;

$VFC(p')$  – коэффициент, определяющий зависимость изменения силы мышцы от скорости ее сокращения;

$StrainF(p)$  – максимальная сила, сообщаемая мышцей в активном состоянии;

$RelaxF(p)$  – сила, сообщаемая мышцей в пассивном состоянии.

Действительной усиле пары мышц в момент времени  $t$  определяется по формуле:

$$RealF(t) = AM.StrainF(p(t)) \cdot AM.Strain(t) \cdot AM.VFC(p'(t)) + PM.RelaxF(t),$$

где  $AM$  – активная мышца пары;

$PM$  – пассивная мышца пары.

Дважды проинтегрировав сплайн идеальной траектории движения, получаем идеальное ускорение. Действительное обобщенное ускорение определяется исходя из возможностей мышечной системы персонажа путем решения обратной задачи динамики (определения идеального усилия мышцы на основе идеального ускорения), нахождения действительного усилия мышц и последующего моделирования прямой динамики. После получения действительного обобщенного ускорения производится повторная интерполяция текущего участка сплайна траектории при задании в качестве крайних условий значений второй ее производной (найденных действительных ускорений).

Для контроля правильности выполнения элементарного движения используются начальные, промежуточные и конечные условия, налагаемые пользователем на обобщенные и максимальные координаты элементов тела персонажа, а также опорные точки, которые определяют положения персонажа относительно опорной плоскости, по которой он движется. Каждое условие представляет собой функцию следующего вида:

$$Condition = f(t, CharacterState(t), \{Param\}) \in \{True, False\}.$$

Начальные условия есть совокупность требований, выполнение которых необходимо для начала элементарного движения, конечные – для его завершения, а промежуточные – продолжения выполнения. Существуют условия трех основных типов:

- 1) проверка отсутствия/наличия опорных точек;
- 2) проверка принадлежности значения обобщенной координаты текущего состояния персонажа окрестности некоторой фиксированной величины или текущего идеального значения координаты: окрестность может быть задана в виде абсолютного или относительного отклонения;
- 3) проверка времени, прошедшего с момента начала элементарного движения на вхождение его в допустимый абсолютный или относительный интервал.

Элементарное движение задает ключевые значения не для всех обобщенных координат тела, а лишь для части из них. Комбинируя такие движения в группы, выполняемых последовательно и параллельно, пользователь получает сложное (составное) движение персонажа:

$$CompoundMovement = \langle \{Param\}, \{(BasicMovement)\} \rangle.$$

При этом элементарные движения, выполняемые параллельно, должны быть непротиворечивыми, то есть не должны определять значения одних и тех же обобщенных координат, а у выполняемых последовательно должны быть совместными конечные и начальные условия соответственно. Невыполнение начальных или промежуточных условий приводит к прекращению моделирования движения персонажа, выполнение конечных – к переходу на следующее элементарное движение последовательности.

Основной сложностью такого подхода к получению анимации персонажа является создание элементарных движений. Поскольку число ключевых состояний, определяющих каждое элементарное движение, невелико, последнее может быть получено и параметризовано пользователем. Для этого, ему необходимо задать положение управляющих звеньев кинематической цепи, представляющей скелет персонажа, и указать, если необходимо, параметры, а также характер влияния принимаемых ими значений на положения управляющих звеньев. Обобщенные координаты остальных звеньев цепи могут быть рассчитаны автоматически путем решения обратной задачи о положениях с учетом особенностей заданной топологии персонажа.

Ключевые состояния управляющих звеньев скелета персонажа могут быть выделены из реально существующего движения (наблюдаемого или записанного) в соответствии со следующими принципами:

- 1) логическая неделимость – элементарное движение должно быть целостной единицей, в составе которой нельзя выделить более простых, логически обособленных элементов;
- 2) универсальность – элементарное движение должно создаваться с учетом возможности его последующего применения другими пользователями;
- 3) монотонность – в качестве элементарных должны выделяться такие части существующего движения, в течении которых характер последнего не претерпевает значительных изменений;
- 4) минимальность числа ключевых состояний – согласуется с монотонностью и говорит о необходимости стремления, при выделении элементарного движения персонажа, к сокращению числа состояний, необходимых для его задания.

Полученные таким образом элементарные движения объединяются в составные, которые, в свою очередь, могут войти в состав более сложных. Элементы сложного движения персонажа образуют выполняемые параллельно последовательности, которые могут рассматриваться независимо друг от друга или выполняться согласованно. В последнем случае, элементы двух различных последовательностей одного сложного движения могут быть синхронизированы по типу начало-начало (начинаться одновременно или последовательно с указанной величиной задержки) или типу начало-конец (следовать одно за другим через заданный промежуток времени). Для элементов одной последовательности сложного движения может быть использован только второй тип синхронизации.

Параметры составного движения определяются на основе параметров входящих в него элементарных и составных. Для этой цели, однотипные параметры нескольких движений, входящих в составное, могут быть объединены под одним именем или сгруппированы. Для каждого параметра элементарного и составного движения может быть определено значения по умолчанию, которое будет использоваться в том случае, если пользователь, при задании движения персонажа, не укажет значение явно. Такой подход позволяет аниматору сконцентрировать внимание только на важной для решения стоящей перед ним задачи параметрах движения, игнорируя остальные.

Автоматический синтез составного движения персонажа становится возможным при наличии набора предварительно созданных элементарных и составных движений в том случае, когда не все, указанные пользователем движения последовательно согласованы. Возникает необходимость получения *промежуточных движений* персонажа, которые бы обеспечили переход от конечного состояния одного движения последовательности определенного пользователем составного к начальному состоянию следующего за ним. Такие промежуточные движения могут быть найдены среди набора предварительно созданных путем поиска наикратчайшей последовательности, позволяющей осуществить переход между упомянутыми состояниями персонажа.

## ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Описанный выше подход к представлению движения был использован для создания системы анимации персонажа. Ее ядром является процессор моделирования динамики твердого тела, на основе которого организовано моделирование движения персонажа. Работа системы обеспечивается тремя программными потоками: управления, моделирования и визуализации. В состав системы входит также банк данных и редакторы, которые дают пользователю возможность создавать объекты окружающей среды, персонажей, задавать внешнее воздействие, создавать составные движения на основе входящих в банк данных системы элементарных. Последние представляют собой микропрограммы, написанные на языке C++ и хранящиеся в виде экспортируемых функций динамических библиотек (dll). Банк данных разделен на две части: модельную и виртуальное пространство. В первой хранятся объекты, которые могут быть использованы для создания анимации, в во второй непосредственно производится моделирование.

В работе системы анимации можно выделить два основных режима: редактирования и моделирования. В первом, основную роль играет поток управления, который обеспечивает взаимодействие с пользователем, а во втором – поток моделирования, осуществляющий расчет динамики твердого тела и движения персонажа. Поток визуализации реализован на основе графической библиотеки OpenGL и связан с двумя другими потоками по средством специального информационного канала, по которому поступают отображаемые на экране геометрические примитивы (отрезки прямых, полигоны).

Перед началом моделирования движения, пользователь должен создать из объектов, хранящихся в банке данных, трехмерную сцену – окружающую персонажа среду. Для этого он переносит объекты из модельной части в виртуальное пространство и там комбинирует их. После этого, пользователь помещает на сцене персонажа и задает его начальное положение. Следующим шагом является определение сценария перемещения – комбинирование элементарных и составных движений, указание их параметров. Когда перечисленные действия выполнены, пользователь запускает процесс моделирования. Рассчитанные в его ходе траектории, вместе с объектами анимации, могут быть экспортированы в другие системы трехмерной графики. Если полученный результат не устраивает пользователя, он вносит необходимые изменения в параметры движений персонажа и запускает моделирование повторно.

## **ЗАКЛЮЧЕНИЕ**

Таким образом, предложенный подход к параметризации сложного движения персонажа позволяет автоматизировать процесс получения анимации, при наличии библиотек предварительно созданных элементарных движений. Использование такого подхода, дает пользователю возможность определять перемещение персонажа в ходе анимации как совокупность элементарных составляющих, задавая необходимые и упуская несущественные параметры движения, после чего наблюдать за ходом моделирования и вносить требуемые изменения. Основной трудностью рассмотренного подхода является создание библиотек элементарных движений, однако использование ряда указанных принципов по их выделению и параметризации позволяет облегчить этот процесс.

## **ЛИТЕРАТУРА**

1. David Sturman, The State of Computer Animation. – ACM SIGGRAPH Computer Graphics, Vol.32, No.1, February, 1998.
2. Larry Israel Gritz, Evolutionary controller synthesis for 3-D character animation. – dissertation, George Washington University, 1999.
3. D. Baraff, An Introduction To Physically Based Modeling. – SIGGRAPH Course Notes, ACM SIGGRAPH, 1995.
4. Р. Глазер, Очерк основ биомеханики. – М.: Мир, 1988 г., 128 с.

## **Об авторе**

Максим Сергеевич Ильиных – аспирант кафедры АСОИУ Ижевского Государственного Технического Университета.

E-mail: amby@cd.istu.udm.ru

## **Abstract**

This work presents an approach to representation of parametrical movement in computer aided character animation as a set of elementary parametrical units. Main principles of the approach is considered along with description of used character and movement models. A character animation system based on the suggested approach is also briefly considered – its main functional mechanisms and capabilities are described.