

МЕТОД СИНТЕЗА ДВИЖЕНИЙ В ЗАДАЧАХ ТРЕХМЕРНОЙ КОМПЬЮТЕРНОЙ АНИМАЦИИ

Семакин Михаил Михайлович
Ижевский Государственный Технический Университет
г. Ижевск, Россия

Аннотация

В данной работе предлагается метод синтеза сложных движений, основанный на модели знаний об объектах и их движениях, а также описываются способы согласования и заимствования движений.

Ключевые слова: компьютерная анимация, движение, синтез.

ВВЕДЕНИЕ

Современные системы трехмерной компьютерной анимации достигли того уровня, когда группе профессионалов под силу создать фактически любой визуальный эффект, придать характер поведения персонажу или же создать целый виртуальный мир со своими законами. В этом можно убедиться, посмотрев, какой огромный успех получили компьютерные спецэффекты в кино, например, в таких фильмах как: «Звездные войны» (Star Wars), «Скрытая угроза» (The Phantom Menace), «Последняя фантазия» (Final Fantasy) и многих других. Тем не менее, для достижения такого высокого уровня требуются огромные усилия и тысячи человеко-часов художников, аниматоров и инженеров.

Теперь, когда доступна технология создания компьютерных персонажей и их оживления, следующим закономерным этапом развития программных систем является поиск методов упрощения этого технологического процесса. Основная сложность анимации компьютерного персонажа заключается в том, что, во-первых, необходимо указать множество параметров для определения пространственного положения персонажа, а во-вторых, необходимо глубокое понимание динамики естественных движений людей и животных.

Существуют технологии, позволяющие повысить уровень абстракции при задании положений трехмерных объектов, включающие моделирование физических законов движений и позволяющие упростить работу практически с любыми типами компьютерных персонажей. Так, например, алгоритм «инверсной кинематики» (inversed kinematics) [1,2] используется для упрощения задания положения объектов, связанных в иерархическую цепочку (так называемый «скелет модели»), когда достаточно задать изменение положения одного объекта (например, ладони руки или ступни ноги), а положения остальных объектов скелета будут найдены автоматически с учетом заданных отношений и параметров подвижности.

Отдельно следует выделить технологию «захвата движения» (motion capture), которая на сегодняшний день представляет собой самый мощный и эффективный инструмент, и наиболее широко используется в кино-видео индустрии. Здесь различными способами при помощи датчиков прикрепленных к живым актерам записываются совершаемые ими движения, а затем эти движения переносятся на виртуальные персонажи.

Другим направлением развития технологии создания анимации является автоматический синтез движений, где задачей ставится генерация сложного движения на основе высокоуровневой модели описания требуемого действия. Разработки в этой области в основном направлены на обеспечение возможности высокоуровневого управления моделями человека или животных для задания наиболее свойственных им движений (например, ходьбы). Существуют системы, в которых создаются «виртуальные контроллеры» (virtual robotic controllers) [3] и на их основе осуществляется управление и параметризация движений модели персонажа, имитируя поведение робота в виртуальной среде с учетом физических законов. Другой метод предназначен для оптимизации созданных движений на основе заданных физических и биомеханических параметров объектов, чтобы обеспечить выполнение набора высокоуровневых ограничений целостности данного движения [4,5].

В данной работе предлагается метод синтеза движений, основанный на модели знаний об объектах и их движениях, а также описываются способы согласования и заимствования движений.

Предполагается, что задача синтеза движений (действия) может быть решена путем целенаправленного перебора имеющихся в базе знаний движений заданного персонажа или его аналогов и согласованием полученной последовательности движений. Критерием отбора является максимальное приближение к конечному результату. В этом случае задача синтеза движений может быть поделена на три части: 1) поиск аналогов (задача «заимствования движений»); 2) логический вывод (поиск решения) – определение комбинации движений; 3) согласование и оптимизация комбинации движений.

Исходными данными являются произвольное начальное и конечное состояние объекта. Для определения промежуточных состояний решается задача поиска в базе знаний движений для этого объекта или его аналогов. Комбинация их движения после выполнения операции согласования приводит объект из заданного начального в заданное конечное состояние.

МОДЕЛИ ЗНАНИЯ О ПРЕДМЕТАХ И ПРОЦЕССАХ ДВИЖЕНИЯ

Объектами предметной области являются *предметы* и *процессы движения* предмета. В общем виде модель знаний о предметной области можно представить деревом классификации:

$$KM = T(C, D, E),$$

где: C – множество узлов дерева – *концептов*; D – множество листьев дерева – *денотатов*; E – множество ребер графа, выстраивающих его вершины в классификационное дерево.

Компоненты C и D представляют собой вершины графа типа дерево.

В общем случае концепты содержат информацию о понятиях, терминах, классах объектов. Денотаты – это информация о конкретных объектах, являющихся образцами (экземплярами) некоторого концепта.

В базе знаний объекты, имеющие оригинальную структуру (отличающуюся от структуры любых других ранее описанных объектов) будут соответствовать вершинам из множества концептов C . Все другие объекты, созданные по подобию структуры определенных концептов, но отличающиеся значениями параметров, представлены листьями дерева – элементами из множества денотатов D .

Ребра в дереве классификации определяют связи между объектами и их предками, то есть определяют схему наследования состава и свойств объектов.

Параметрическое описание объекта отражает его функциональные, физические или иного рода отличительные особенности. Параметрическое описание концепта объекта представляет собой объединение множества *атрибутов* и *состава объекта*.

Состав представляет структурное описание объекта, то есть то, из чего состоит объект. В структуре концепта состав является объединением графов *образов объектов* [6]. Образ объекта – это та часть (условно) полной информации об объекте, которая необходима и достаточна для решения поставленной задачи.

Граф образа объекта:

$$G^p = (Q, EQ),$$

где: Q – множество элементов состава (вершин графа) q^{p-1} , представляющих ссылки на объекты уровня $(p - 1)$; EQ – множество ребер, связывающих элементы состава (вершины графа) с помощью некоторых бинарных отношений.

Каждый элемент состава содержит ссылку на объект-прообраз, а также *шаблон заимствования* значений атрибутов у объекта-прообраза. Образ объекта копирует все значения собственных атрибутов объектов-прообразов, ссылки на которые вошли в его состав. Шаблон заимствования позволяет указывать атрибуты и значения, которые необходимо изменить в прототипе (и в подобъектах прототипа) для представления состава данного объекта. Этот шаблон можно представить в виде объединения:

$$PATTERN = AT \cup Q,$$

где: AT – отражает множество собственных атрибутов объекта-прообраза (прототипа), значения которых отличаются от значений этих же атрибутов этого объекта-прообраза; Q – множество ссылок и шаблонов (аналог состава), которые определяют какие образы объектов состава необходимо заимствовать (наследовать) для данного объекта, а также значения каких атрибутов необходимо изменить (переопределить) при заимствовании.

В отличие от собственных атрибутов, состав может заимствоваться не полностью (или вообще не заимствоваться), так как в ряде случаев заимствовать весь состав оказывается нецелесообразно. В этом случае, именно шаблон определяет, какие объекты состава будут заимствованы. Такая схема, позволяет организовать более гибкий механизм заимствования (наследования) состава и атрибутов из объектов-прототипов.

Ребра дерева классификации объектов предметной области описывают иерархию наследования атрибутов и состава. Этот механизм можно расширить, если каждому ребру этого дерева поставить в соответствие шаблон заимствования значений атрибутов $PATTERN$. Тогда денотаты можно определить как объекты, не содержащие собственных атрибутов и состава, но наследующие их от концептов из множества C в соответствии с шаблоном заимствования значений атрибутов.

Модель предмета включает описание кинематической схемы – структурное описание, а также набор некоторых *классификационных свойств* – параметрическое описание:

$$M = (AC, Q, EQ),$$

где: AC – множество атрибутов (классификационных свойств); Q – звенья кинематической схемы; EQ – множество связей между звеньями – кинематических пар.

Звенья кинематической схемы предмета ссылаются на геометрические модели объектов, которые также могут быть представлены в БЗ или при помощи ссылок на внешние файлы.

Кинематическая схема представляется при помощи графа образа объекта, где вершинами $q \in Q$ (элементами состава) являются звенья, а множеством ребер EQ описываются кинематические пары звеньев.

Классификационные свойства – это атрибуты концепта, используемые для построения альтернативных деревьев классификации.

Модель процесса движения определяет множество промежуточных состояний для осуществления движения из заданного начального в заданное конечное состояние предмета:

$$PM = (KS_n, P, KS_k), \quad (1)$$

где: KS_n – кинематическая схема, определяющая начальное состояние движения; KS_k – кинематическая схема, определяющая конечное состояние движения; P – некоторая операция, определяющая метод получения промежуточных состояний (семейство графиков) для осуществления движения из начального состояния KS_n в конечное состояние KS_k .

KS_n и KS_k должны представлять одну и ту же кинематическую схему, а их отличие заключается только в разнице значений параметров, определяющих положения звеньев в пространстве (состояние кинематической схемы).

Кинематическая схема, используемая в описании процесса движения, может отображать только подграф схемы некоторого предмета или же включать звенья и кинематические пары из нескольких моделей предметов, а также образовывать новые кинематические пары. Поэтому, для представления начального и конечного состояния движения используется *модель состояния* или *модель сцены*.

Модель состояния предназначена для описания объектов, участвующих в том или ином действии или работе. Модель состояния описывается аналогично модели предмета.

Можно выделить следующие основные виды операций P , используемые системой для описания и синтеза процессов движения:

- 1) генерация графиков перехода из KS_n в KS_k без использования имеющихся процессов движения;
- 2) интерпретация семейства графиков $R(KS, t)$, описывающих изменения относительного положения звеньев кинематической схемы;
- 3) *заимствование* и интерпретация указанного процесса движения;
- 4) *последовательное согласование* двух простых движений, заданных семействами графиков $R_1(KS, t)$ и $R_2(KS, t)$;
- 5) *параллельное согласование* двух простых движений, заданных семействами графиков $R_1(KS, t)$ и $R_2(KS, t)$;
- 6) выполнение встроенных функций параметризации семейства графиков получаемых из указанного процесса движения;
- 7) интерпретация *схемы составного (сложного) процесса движения*.

Начальное KS_n и конечное KS_k состояния, а также операция P представляют собой базовое параметрическое описание процесса движения. Модель процесса движения расширяется за счет включения входных параметров и структурного описания (в случае сложного процесса), необходимых для выполнения операции, а также за счет последующей параметризации самого процесса (добавление классификационных атрибутов и вычислительных моделей).

Рассмотрим последовательность действий для определения в базе знаний движения «ходьба».

1. Создание простого процесса движения «ходьба».
 - 1.1. Задание начального состояния KS_n таким образом, чтобы направление ходьбы было параллельно оси X .
 - 1.2. Задание промежуточных положений (ключевых кадров) движений «шаг левой» и «шаг правой». Движения задаются вдоль оси X .
 - 1.3. Запись полученного движения как простого процесса в БЗ.
2. Параметризация «ходьбы».
 - 2.1. Создание в СУБЗ нового процесса движения.
 - 2.2. В качестве операции P новому процессу указывается функция «масштабирование поступательного движения» (из библиотеки встроенных операций).
 - 2.3. Задание параметров операции:
 - «масштабируемый процесс» (указывается ссылка на процесс созданный в п.1);
 - «масштабируемые оси» (т.к. в п.1 в качестве направления движения была выбрана ось X , то указываем только эту ось для масштабирования);
 - коэффициент масштаба (по умолчанию задается равным 1, но далее для него определяется вычислительная модель).
 - 2.4. Определение новых параметров процесса «ходьба»:
 - «длина шага» – число, определяющее относительную длину одного шага;
 - «количество шагов» – число, определяющее количество генерируемых процессом шагов левой и правой ноги, поочередно.
 - 2.5. Задание вычислительных моделей для связи с нововведенными параметрами:

$$T_1 = 0,5 \cdot nstep,$$
$$kscl = step_len / step_len_1,$$

где: T_1 – параметр, определяющий приведенный период времени выполнения первого процесса; $nstep$ – параметр «длина шага» (см. 2.4); $kscl$ – параметр «коэффициент масштаба» (см. 2.3); $step_len$ – параметр «длина шага» (см. 2.4); $step_len_1$ – константа, определяющая относительную длину шага в первоначальном движении.

Значение 0,5 приведенного полупериода начального движения (равного 1) умножается на количество шагов, т.к. одному полупериоду заданного в п.1 движения «ходьба» соответствует один шаг левой, а затем правой ногой.

Коэффициент масштабирования поступательного движения в задается пропорциональным отношением между заданной длиной шага $step_len$ и длиной шага первоначальном движении $step_len_1$ (для простоты, можно задать $step_len_1 = 1$, тогда параметр $step_len$ будет определять коэффициент сжатия или растяжения одного шага при ходьбе).

3. Создание модификаций процесса движения «ходьба» для упрощения автоматического синтеза сложных движений и удобства использования в последующих проектах анимации.

- 3.1. Создание процесса, унаследованного от процесса, созданного в п.2, где сначала выполняется шаг правой ногой, и, соответственно изменяются начальное и конечное состояния кинематической схемы.
- 3.2. Создание процессов (аналогично п.п. 3.1), где начальными состояниями являются: $\frac{1}{4}$ и $\frac{3}{4}$ первоначального движения, а конечными – $\frac{3}{4}$ и $\frac{1}{4}$, соответственно.

ГЕНЕРАЦИЯ ПЕРЕХОДНЫХ ГРАФИКОВ

Каждому процессу движения соответствует заданное начальное состояние KS_n и графики, предназначенные именно для этого начального состояния. Таким образом, возникает задача, когда необходимо «подстроиться» под начальное состояние некоторого выбранного процесса движения, чтобы его выполнение шло корректно. Для этого системой автоматически выполняется операция по созданию графиков перехода, описывающих необходимое движение для «подстройки» под начало выполнения указанного процесса, а также выполняется оптимизация переходного движения.

Описываемый способ основывается на алгоритме инверсной кинематики и применяется, когда исходные состояния отличаются незначительно и нет необходимости в поиске и подстановке ранее созданных движений.

Состояние кинематической схемы описывается положением ее звеньев. Для более качественного моделирования движения при задании инверсной кинематики в некоторых звеньях задается значение коэффициента подвижности меньше 1. В данном случае, такие звенья можно охарактеризовать как *ведущие*, т.к. именно с помощью них мы будем определять поступательное движение всей кинематической схемы (на рис.2,а ведущие звенья закрашены в темный цвет).

На первом этапе алгоритма по начальному и конечному состоянию кинематической схемы рассчитываются вектора направления поступательных движений ведущих звеньев. Затем, выполняется расчет инверсной кинематики и фиксируется полученное конечное состояние.

На втором этапе определяются различия между параметрами относительного положения звеньев в полученном состоянии и в требуемом результате. После задания графиков для изменения относительных параметров звеньев, результат сравнивается с ожидаемым и, если все сходится, выполняется оптимизация движения, иначе считается, что в данном случае операция генерации графиков перехода невыполнима.

В качестве примера рассмотрим генерацию переходного движения из положения «сидя» в начальное состояние движения «ходьба». Начальное и конечное состояния переходного движения показаны на рис. 1.

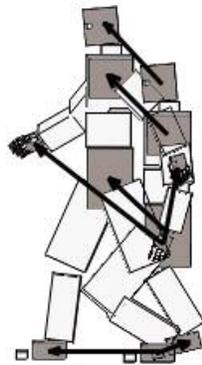


а) начальное состояние перехода

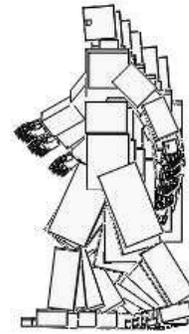
б) конечное состояние перехода

Рис. 1. Пример начального и конечного состояния переходного движения

Определив вектора поступательного движения ведущих звеньев (рис.2,а), при помощи алгоритма инверсной кинематики (см. п.) строим переходное движение – множество состояний (рис.2,б) – семейство графиков звеньев.



а) определение векторов поступательного движения ведущих звеньев



б) генерация промежуточных состояний при помощи инверсной кинематики

Рис. 2. Пример генерации переходного движения

Для оптимизации полученного движения анализируются и подбираются комбинации из следующих действий:

- 1) одновременное выполнение графиков поступательного (инверсная кинематика) и относительных движений звеньев;
- 2) определение времени движения звеньев на основе подсчета средней скорости движения этих же звеньев в последующем процессе движения;
- 3) коррекция времени движения звеньев пропорционально заданным для них коэффициентам подвижности;
- 4) выполнение операции последовательного согласования полученного движения с последующим процессом движения.

СОГЛАСОВАНИЕ ДВИЖЕНИЙ

Операция согласования движений предназначена для создания нового (сложного) движения, состоящего из двух и более определенных ранее движений.

Задачу согласования двух движений можно решить двумя способами:

- 1) задание в базе знаний специальных движений предназначенных только для корректного перехода от одного движения к другому;
- 2) процедурный способ, т.е. использование алгоритма синтеза промежуточных состояний (сюда также входит определение возможности выполнения операции согласования).

В рамках данной задачи движение будем рассматривать как семейство графиков, связанных с параметрами звеньев кинематической схемы и отображающих изменение их значений в заданном периоде времени.

Для примера рассмотрим две задачи:

1. Выполнить движение «ходьба» (2 шага), затем «прыжок» и продолжить движение «ходьба» (еще 3 шага).
2. Выполнить движение «ходьба» (3 шага) и одновременно движение «бросок мяча».

В первом случае необходимо решить задачу последовательного согласования, а во втором – параллельного выполнения движений.

Временная диаграмма для первого примера представлена на рис.3, здесь:

$t_{1,н}$, $t_{1,к}$ – начальное и конечное время выполнения первого движения («ходьба» на 2 шага);

$t_{2,н}$, $t_{2,к}$ – начальное и конечное время выполнения второго движения («прыжок»);

$t_{3,н}$, $t_{3,к}$ – начальное и конечное время выполнения третьего движения («ходьба» на 3 шага).

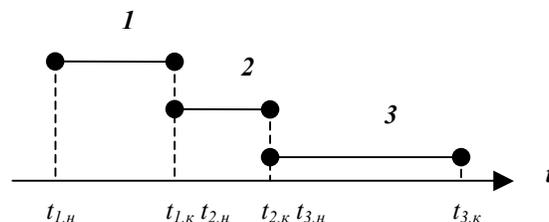


Рис. 3. Пример временной диаграммы последовательности процессов

Используемые движения «ходьба» и «прыжок» характеризуются начальным и конечным положениями (состояниями), которые в общем случае не совпадают.

Задача сводится к определению промежуточных (переходных) состояний между окончанием одного и началом выполнения другого движения. При этом можно выделить три способа определения переходных временных областей:

- 1) «плавный переход» от одного движения к другому за счет использования областей окончания и начала двух движений (рис.4);
- 2) пересечение области окончания первого с началом второго движения (рис.5);

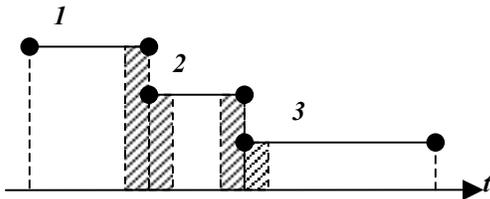


Рис. 4

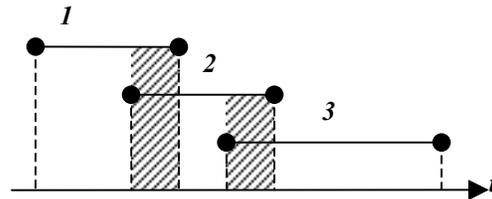


Рис. 5

- 3) вставка переходных сегментов, приводящих из заданного конечного состояния одного движения к заданному начальному состоянию для другого движения (рис.6).

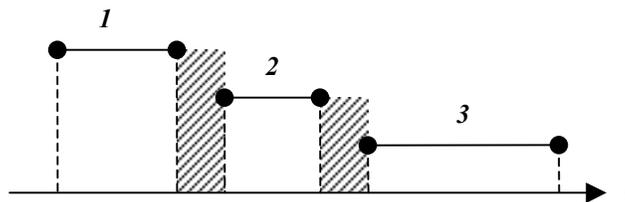


Рис. 6

В первом случае («плавный переход»), на границах области согласования ставятся узловые точки сплайна и определяются касательные, затем с учетом условия непрерывности и гладкости сплайна строится сегмент между этими точками, который и определяет искомые промежуточные состояния.

Пересечение области окончания первого с началом второго движения (см. рис.5) используется для циклических движений и, если в некоторый момент времени графики обоих движений имеют наиболее близкие значения. Если разница значений невелика, то ее можно интерполировать при помощи вставки небольшого сегмента (аналогично способу «плавного перехода»).

В третьем случае (см. рис. 6), вставка переходных сегментов интерполируется, используя значения касательных на границах и с учетом условия гладкости кривой.

При параллельном выполнении двух движений определяется, какие объекты (звенья кинематической схемы) будут изменяться в соответствии с графиками первого движения, а какие по графикам второго движения.

Для определения, какие графики первого движения необходимо заменить графиками второго движения, можно (как показали эксперименты) сравнивать их длины путей.

В некоторых случаях используются усредненные значения двух графиков:

$$\mathbf{r}_{cp}(t) = \mathbf{r}_1(t) + [\mathbf{r}_2(t) - \mathbf{r}_1(t)] \cdot k,$$

где: t – текущий момент времени; $\mathbf{r}_{cp}(t)$ – искомое промежуточное значение в момент времени t ; $\mathbf{r}_1(t)$, $\mathbf{r}_2(t)$ – значения 1-го и 2-го графиков в момент времени t ; $k \in [0, 1]$ – коэффициент «усреднения».

ЗАИМСТВОВАНИЕ ДВИЖЕНИЯ

Займствование движения выполняется, когда известное движение необходимо распространить на другую, близкую по топологии, но, быть может, отличающуюся по параметрам, кинематическую схему.

Исходные данные для задачи займствования движений:

- 1) кинематическая схема предмета (актера), для которого осуществляется поиск движений;
- 2) кинематическая схема выбранного аналога;
- 3) модель процесса движения аналога.

Сначала необходимо проанализировать возможность выполнения операции займствования, а затем, если все условия выполнены, произвести параметризацию процесса движения на основе критериев, определяющих различия между кинематическими схемами актера и выбранного аналога.

Для анализа входных данных можно ограничиться простым условием: множество кинематических пар актора должно входить во множество пар кинематической схемы аналога, или наоборот.

Как показали эксперименты по подстановке одних и тех же процессов движений на разные кинематические схемы, наиболее ощутимые различия связаны с поступательными движениями, заданными при помощи инверсной кинематики.

Если проанализировать изменение формы поступательного движения при изменении размерных характеристик объекта, то можно увидеть прямую зависимость между размерами объекта и параметрами совершаемого им движения, то есть основным критерием сходства движений при заимствовании, в данном случае, будет условие сохранения пропорций между параметрами движения, определяющими размеры выходных графиков и размерными характеристиками объекта. При чем, можно определить выполнение этого условия с заданной точностью.

СОСТАВНОЙ ПРОЦЕСС ДВИЖЕНИЯ

Модель знаний о составном (сложном) процессе движения содержит следующую информацию:

- начальное и конечное состояния кинематической схемы;
- состав подпроцессов движения;
- шкала ключевых моментов времени подпроцессов;
- последовательность, определяющая связь подпроцессов со шкалой моментов времени для определения времени начала и окончания каждого подпроцесса;
- атрибуты- вычислительные модели для параметризации процесса движения;
- классификационные свойства.

Структура составного процесса движения:

$$CP = \langle KS_n, p_{cp}, KS_k, \{SubP_{nproc}\}, CompProcTimeScale, CompProcKeys, \{ParamCM_{nparam}\}, \{ClassAttr_{na}\} \rangle,$$

где: KS_n, KS_k – начальное и конечное состояние кинематической схемы;

p_{cp} – идентификатор составного процесса (тип операции P в (1));

$\{SubP_{nproc}\}$ – последовательность подпроцессов $SubP_i$, где $i=1..nproc$, $nproc$ – количество подпроцессов;

$CompProcTimeScale$ – шкала ключевых моментов времени в которые могут начинаться и оканчиваться подпроцессы;

$CompProcKeys$ – последовательность, определяющая связь подпроцессов $SubP_i$ со шкалой моментов времени $CompProcTimeScale$;

$\{ParamCM_{nparam}\}$ – множество атрибутов- вычислительных моделей, определяющих способы параметризации процесса;

$nparam$ – количество вычисляемых параметров процесса;

$\{ClassAttr_{na}\}$ – множество классификационных свойств составного процесса; na – количество классификационных свойств.

Схему интерпретации составного процесса движения графически можно представить как граф, узлами которого являются ключевые состояния кинематических схем предметов, расставленные в соответствии с последовательностью моментов времени, а ребрами отображаются подпроцессы движения.

Операция согласования подпроцессов может осуществляться автоматически в процедуре интерпретации схемы процесса. Сначала выполняется последовательное согласование подпроцессов, ребра которых располагаются друг за другом. Затем производится параллельное согласование подпроцессов, ребра которых располагаются друг под другом.

ЗАКЛЮЧЕНИЕ

Предложенный метод может быть использован не только в сфере производства компьютерных анимационных фильмов, но и в таких применениях как:

- моделирование движений в динамических интерактивных средах (игры, виртуальная реальность, и т.п.);
- сокращение объема хранимой или передаваемой по сети информации;
- моделирование движений робототехнических систем;
- решение отдельных задач, связанных с поиском оптимальной траектории движения, и т.д.

ЛИТЕРАТУРА

1. Norman I. Badler and Kamran H. Manoochehri and Graham Walters. Articulated figure positioning by multiple constraints, IEEE Computer Graphics & Applications, 7 (6), pp. 28-38 (June 1987).
2. Paul M. Isaacs and Michael F. Cohen. Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics, Computer Graphics (Proceedings of SIGGRAPH 87), 21 (4), pp. 215-224 (July 1987, Anaheim, California). Edited by Maureen C. Stone.
3. Jessica K. Hodgins and Wayne L. Wooten and David C. Brogan and James F. O'Brien. Animating Human Athletics, Proceedings of SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series, pp. 71-78 (August 1995, Los Angeles, California). Addison-Wesley. Edited by Robert Cook.

4. Andrew Witkin and Kurt Fleischer and Alan Barr. Energy Constraints on Parameterized Models, Computer Graphics (Proceedings of SIGGRAPH 87), 21 (4), pp. 225-232 (July 1987, Anaheim, California). Edited by Maureen C. Stone.
5. Andrew Witkin and Michael Kass. Spacetime Constraints, Computer Graphics (Proceedings of SIGGRAPH 88), 22 (4), pp. 159-168 (August 1988, Atlanta, Georgia). Edited by John Dill.
6. Кучуганов В. Н. Автоматический анализ машиностроительных чертежей. – Иркутск: Изд-во Иркут. ун-та, 1985. – 112 с.: ил.

Об авторе

Семакин Михаил Михайлович – аспирант кафедры САПР Ижевского Государственного Технического Университета.

E-mail: mike@cd.istu.udm.ru, semik@udm.net

Abstract

This work presents a method of synthesis of complex motions, founded on models of knowledges about objects and their movements, as well as are described the ways of coordination and borrowing the motions.