

МЕТОДЫ ЖЕСТОВОГО УПРАВЛЕНИЯ ТРЕХМЕРНЫМИ ОБЪЕКТАМИ И ПАРАМЕТРАМИ ВИРТУАЛЬНОЙ СРЕДЫ С ПОМОЩЬЮ KINECT¹

А. В. Мальцев, М. В. Михайлюк, П. Ю. Тимохин

GESTURE CONTROL METHODS FOR THREE-DIMENSIONAL OBJECTS AND VIRTUAL ENVIRONMENT SETTINGS USING KINECT

A. V. Maltsev, M. V. Mikhaylyuk, P. Yu. Timokhin

Аннотация. *Актуальность и цели.* В настоящее время большое внимание уделяется разработкам новых типов интерфейсов управления робототехническими средствами. Привычные интерфейсы (джойстики, пульта управления, экзоскелеты и т.д.) имеют ряд недостатков, которые хотелось бы устранить в новых эргономичных интерфейсах. Цель данного исследования – разработка методов жестового управления трехмерными объектами и параметрами окружающей обстановки в системах виртуального окружения. В качестве примера объекта управления выбран мобильный робот, а в качестве параметров окружающей среды – интенсивность дождя и снега. *Материалы и методы.* Реализация задач была достигнута за счет использования устройства *Kinect*, которое позволяет в масштабе реального времени получать координаты суставов оператора, находящегося в рабочей зоне. На основе этих данных определяются позы и жесты оператора, которые затем используются для управления движением мобильного робота, его манипулятора и изменения параметров внешней среды в виртуальной сцене. *Результаты.* В работе подробно описаны методы кодирования и определения управляющих поз и жестов оператора, а также предлагается конкретный набор поз и жестов рук оператора для управления движением виртуальной моделью гусеничного мобильного робота и его манипулятора, а также интенсивностью дождя и снега в виртуальной сцене. *Выводы.* Предлагаемые методы реализации жестового интерфейса могут быть использованы для разработки новых типов эргономичного интерфейса управления объектами в условиях, когда привычные средства управления применять затруднительно.

Ключевые слова: виртуальная среда, распознавание жестов, поза, интерфейс управления, мобильный робот.

Abstract. *Background.* Today much attention is paid to the development of new types of robotic control interfaces. Regular interfaces (joysticks, control panels, exoskeletons, etc.) have a number of drawbacks, which we would like to overcome in new ergonomic interfaces. The goal of this research – the development of methods of gesture control of 3D objects and environment setting in virtual environment systems. As the example, a mobile robot was selected as control object, and rain and snow intensities – as environment settings. *Materials and methods.* The solution was implemented by means of *Kinect* device which allows obtaining joints' coordinates in real time for the operator being in work area. On the basis of these data operator's poses and gestures are defined and then used for movement control of mobile robot, its manipulator and environment settings in the virtual scene. *Results.* The paper described in detail encoding methods and definitions for control

¹ Исследования выполнены при поддержке РФФИ, грант № 15-07-04544.

poses and gestures of the operator, as well as offers a specific set of poses and gestures for operator's hands to control the movement of a virtual model of tracked mobile robot and its manipulator, as well as rain and snow intensities in the virtual scene. *Conclusions.* Proposed methods of gesture interface implementation can be used to develop new types of ergonomic object control interfaces, when regular control tools are hardly to use.

Key words: virtual environment, gesture recognition, pose, control interface, mobile robot.

Введение

В настоящее время проблема эргономичного (удобного и естественно-го) интерфейса между оператором и управляемым им роботом является достаточно актуальной. Использование стандартных устройств (клавиатуры, мыши, джойстика) не является подходящим для таких задач. Пульты управления могут быть использованы для роботов с небольшим числом степеней свободы, тогда оператор двумя руками может одновременно воздействовать на два элемента управления. Однако при росте степеней свободы такое управление приводит к необходимости попеременно управлять отдельными шарнирами, что крайне неэргономично. Удобным интерфейсом можно считать так называемый копирующий режим, при котором управление осуществляется оператором с помощью экзоскелета, однако это применимо только для антропоморфных роботов. Управление с помощью голоса [1] реализует командный режим и освобождает руки оператора, что очень удобно во многих случаях, однако имеет и свои ограничения, заключающиеся в слабой контролируемости оператором процесса выполнения запущенной команды.

В данной работе для управления мобильным колесным или гусеничным роботом предлагается использовать жестовый интерфейс. В реальности люди манипулируют объектами с помощью пальцев и рук, поэтому целесообразно поставить задачу разработать интуитивный и естественный интерфейс путем имитации того, что делает человек в реальном пространстве.

Работы в этом направлении ведутся многими исследователями. Так, в работе [2] рассматривается управление манипулятором с помощью движения кисти руки и пальцев. Координаты ладони и кончиков пальцев вычисляются на основе изображений, полученных с помощью камеры. Позиция руки используется для перемещения манипулятора, позиции кончиков пальцев – для управления пальцами манипулятора. Апробация технологии проводилась на виртуальной модели. В работе [3] с использованием *Kinect* также отслеживается положение кисти и пальцев руки. Полученные данные ассоциируются с различными командами работы с приложениями (открыть, закрыть и т.д.). В некоторых исследованиях основой жеста предлагается рассматривать траекторию ладони оператора. Так, в [4] изображение руки выделяется из карты глубины по некоторому порогу (так как считается, что рука всегда вытянута перед телом). Затем вычисляется 2D-траектория проекций положения руки, которая интерпретируется как скрытая Марковская модель.

В настоящей работе для распознавания жеста базовым также является использование устройства *Microsoft Kinect*, которое позволяет в масштабе реального времени на основе карты глубины вычислить трехмерные координаты суставов (плеч, локтей, запястий и т.д.) оператора. По ним определяются углы между соответствующими костями оператора [5]. Вектор этих углов об-

разует позу, а вектор поз – жест оператора. Далее подробно рассмотрим реализацию предлагаемого жестового интерфейса управления.

Кодирование управляющих жестов рук оператора

Рассматривая жестовое управление трехмерными объектами и параметрами виртуальной среды, мы будем предполагать, что это управление производится оператором только с помощью рук. Однако предлагаемые методы и алгоритмы применимы также к случаям управления с использованием нижних конечностей или всего тела в целом.

Любой жест руки оператора можно представить в виде некоторой конечной последовательности ее поз – положений в заданные моменты времени. Поэтому, прежде чем описывать жесты, рассмотрим подход к цифровому кодированию позы руки. Определим позу как совокупность значений углов в суставах руки и состояния S ее ладони. В данной работе используются углы φ и ψ в плечевом суставе, а также θ , γ – в локтевом (рис. 1). Пусть в представленном на рисунке положении (плечо и предплечье перпендикулярны друг другу, плечо прижато к туловищу и параллельно ему, предплечье направлено вперед) все указанные углы равны 0. Положительным поворотом (угол > 0) будем считать поворот звена (плеча или предплечья) руки против часовой стрелки, если смотреть с конца направляющего вектора Z_i оси, вокруг которой осуществляется этот поворот (см. рис. 1). Состояние S примем равным 1, если ладонь сжата в кулак, и 0 – в противном случае.

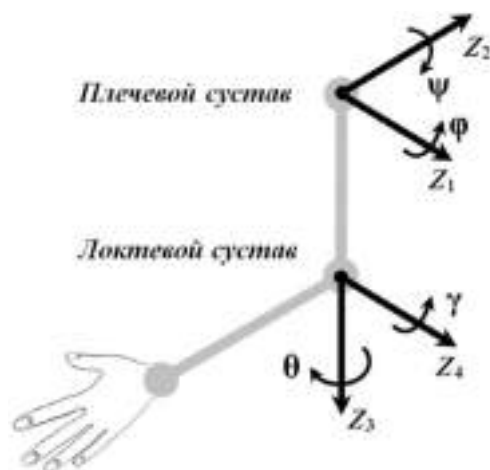


Рис. 1. Задание позы руки с помощью углов в ее сочленениях

Текущие значения углов φ , ψ , θ , γ руки оператора, а также состояние ее ладони S можно определить с помощью устройства *Microsoft Kinect*. Представим, как показано на рис. 2, плечевой и локтевой суставы в виде системы иерархически связанных осевых шарниров. Используя *Kinect API* [6], получим координаты опорных точек скелета оператора, находящегося в рабочей области устройства. Для нахождения искомым углов на основе этих координат можно воспользоваться методами и алгоритмами, подробно изложенными в работе [2].

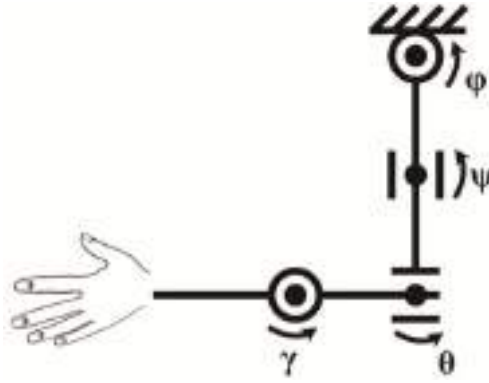


Рис. 2. Представление суставов в виде иерархии шарниров

Хранение позы руки в виде набора углов и состояния ладони является затратным с точки зрения занимаемого объема памяти и неудобным при сравнении нескольких поз. Для более эффективного решения этих задач мы предлагаем кодировать каждую позу в виде одного 32-битного беззнакового целого числа I_p – идентификатора позы, содержащего данные об углах φ , ψ , θ , γ и состоянии S . При этом непрерывный диапазон $[-180^\circ, 180^\circ]$, охватывающий значения данных углов, делится на открытые справа полуинтервалы одинаковой длины d_a (выбирается в зависимости от решаемой задачи, в среднем составляет порядка 20°), последовательно нумеруемые от 0 до $n = (360/d_a) - 1$. Все углы, входящие в один полуинтервал, будем считать идентичными значению его середины. Тогда для любого угла $x \in [-180^\circ, 180^\circ]$ можно вычислить идентификатор $I(x)$ – номер полуинтервала, в котором расположен данный угол:

$$I(x) = \left[(x + 180^\circ) / d_a \right], \quad (1)$$

где квадратные скобки обозначают целую часть числа. В связи с анатомией человеческой руки значение 180° не достигается ни одним из углов φ , ψ , θ , γ , поэтому его можно не рассматривать. Записанные в фиксированной последовательности состояние S ладони (один бит: 1 – сжата в кулак, 0 – разжата) и идентификаторы $I(\varphi)$, $I(\psi)$, $I(\theta)$, $I(\gamma)$ углов руки, представленные каждый 7 битами, составляют идентификатор I_p позы руки. Вычисление значения такого идентификатора будем производить по формуле

$$I_p(S, \varphi, \psi, \theta, \gamma) = (S \ll 28) + (I(\varphi) \ll 21) + \\ + (I(\psi) \ll 14) + (I(\theta) \ll 7) + I(\gamma), \quad (2)$$

где \ll обозначает операцию побитового сдвига влево.

Заметим, что при решении некоторых задач можно определить такие позы, для которых значения одного или нескольких углов будут неважны. Тогда во все разряды I_p , хранящие идентификаторы этих углов, будем записывать 1. Это можно сделать, выполнив операцию «побитовое ИЛИ» между I_p и целочисленной беззнаковой маской M_p , имеющей структуру, эквивалентную I_p , причем каждый ее значащий разряд будет содержать 0, а незначащий – 1.

Для проведения корректного сравнения поз маску M_p необходимо хранить вместе с соответствующим ей идентификатором позы.

Само сравнение текущей позы $I_{p,cur}$ руки оператора, определяемой с помощью *Kinect*, с одной из заранее заданных поз I_p , имеющей маску M_p , будем производить путем проверки равенства

$$I_p = I_{p,cur} | M_p, \quad (3)$$

где знак «|» обозначает операцию «побитовое ИЛИ». При соблюдении данного равенства позы $I_{p,cur}$ и I_p являются идентичными.

Жест руки можно задать в виде N последовательно пройденных рукой поз, где $N > 1$. Для простоты в данной работе ограничимся тремя позами ($N = 3$). Идентификаторы I_p поз, входящих во все выбранные управляющие жесты, вместе с соответствующими им масками M_p занесем в таблицу T поз, исключая повторение (рис. 3). Тогда любой из этих пар (I_p, M_p) будет соответствовать уникальному номеру $T_p \in [0, mN - 1]$ строки в таблице T , где m – общее количество заданных управляющих жестов.

Каждый управляющий жест представим в виде беззнакового целочисленного идентификатора I_g , содержащего в заданном порядке номера $T_{p,i}$ строк из таблицы T , которые соответствуют входящим в этот жест позам, где $i \in [0, 2]$ – очередность позы в жесте. В рассматриваемом случае I_g представляет собой 32-битное целое число, в котором под значение каждого $T_{p,i}$ отводится 10 бит (рис. 3). Рассчитать его можно по формуле

$$I_g = \sum_{i=0}^2 (T_{p,i} \ll 10i). \quad (4)$$

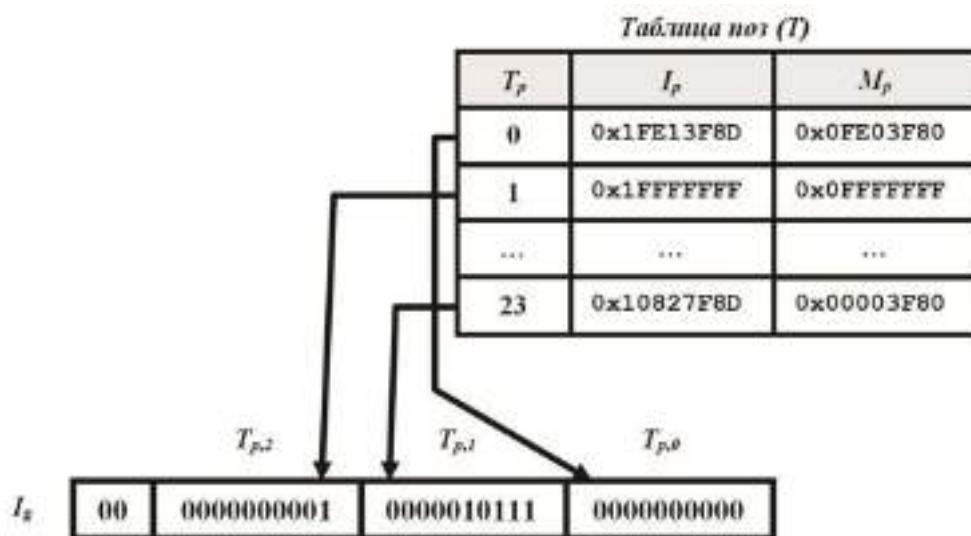


Рис. 3. Кодирование жеста с помощью целочисленного идентификатора

При решении задачи управления какими-либо объектами или параметрами виртуальной среды необходимо заранее определить набор управляющих жестов и соответствующих им операций, а также вычислить для каждого

жеста его идентификатор I_g описанным выше способом. Совокупность всех пар (I_g , операция) образует базу B_g управляющих жестов для данной задачи.

Определение жестов пользователя

Рассмотрим алгоритм идентификации жеста пользователя, находящегося в рабочей области устройства *Kinect*. Пусть максимальное время, отведенное на совершение одного жеста, равно t_{max} . Опрашивая устройство с помощью *Kinect API* с частотой $f = 30$ Гц, будем получать координаты опорных точек скелета оператора. Используя подход, детально описанный в работе [2], вычислим на основе полученных данных значения углов φ , ψ , θ , γ интересующей нас руки (или обеих рук). Далее по формуле (2) определим идентификатор $I_{p,t}$ позы руки в текущий момент времени t . Если $I_{p,t}$ не совпадает с идентификатором $I_{p,t-\Delta t}$ позы той же руки в момент $t - \Delta t$ (где $\Delta t = 1/f$) предыдущего опроса устройства, значит, рука движется и необходимо начать отслеживание производимого ею жеста. Время начала жеста $t_{g,0}$ при этом равно t , первая поза жеста определяется как $I_{p,0} = I_{p,t-\Delta t}$, вторая – $I_{p,1} = I_{p,t}$. Далее, продолжая опрашивать устройство *Kinect*, каждый раз проверяем равенство текущей позы $I_{p,t}$ и предыдущей позы $I_{p,t-\Delta t}$. При условии $t - t_{g,0} \geq t_{max}$ считаем, что жест не состоялся. Тогда начинаем процесс сначала.

Если $t - t_{g,0} < t_{max}$ и $I_{p,t} \neq I_{p,t-\Delta t}$, то мы достигли финальной третьей позы жеста и $I_{p,2} = I_{p,t}$. В данном случае, имея три идентификатора $I_{p,0}$, $I_{p,1}$, $I_{p,2}$, проводим поиск соответствующих им поз в таблице T . Для этого, пробегая таблицу сверху вниз, выполняем сравнение идентификаторов с использованием формулы (3). При нахождении в T всех поз выявленного жеста формируем на основе соответствующих им номеров $T_{p,i}$ 32-битный идентификатор I_g жеста, используя формулу (4). Если какой-либо из поз нет в таблице T , то устанавливаем все биты ее $T_{p,i}$ в 1. Если сформированный идентификатор I_g присутствует в подготовленной заранее базе B_g управляющих жестов, выполним приписанное к данному жесту действие. Таким образом, имеем следующий алгоритм определения управляющего жеста руки пользователя:

- 1) опрос устройства *Kinect* и получение опорных точек скелета пользователя;
- 2) вычисление углов φ , ψ , θ , γ в суставах руки;
- 3) расчет идентификатора $I_{p,t}$ позы руки в текущий момент времени t по формуле (2);
- 4) если $I_{p,t} = I_{p,t-\Delta t}$, возврат к шагу 1; иначе сохраняем время начала жеста $t_{g,0} = t$ и две первые позы этого жеста $I_{p,0} = I_{p,t-\Delta t}$ и $I_{p,1} = I_{p,t}$;
- 5) выполнение пунктов 1–3 до тех пор, пока $I_{p,t} = I_{p,1}$ или $t - t_{g,0} < t_{max}$;
- 6) если $I_{p,t} \neq I_{p,1}$, то сохраняем третью позу жеста $I_{p,2} = I_{p,t}$; иначе переходим к шагу 1;
- 7) поиск $T_{p,i}$ для каждого сохраненного $I_{p,i}$ ($i \in [0, 2]$) по таблице T с применением формулы (3); при отсутствии $I_{p,i}$ в T записываем 1 во все разряды $T_{p,i}$;
- 8) формирование идентификатора I_g жеста по формуле (4);
- 9) если I_g входит в базу B_g управляющих жестов, то происходит выполнение приписанной к жесту операции.

Управление мобильным роботом и параметрами виртуальной среды

В качестве примера в данной работе рассматривается жестовое управление виртуальной моделью мобильного робота MF-4 с манипулятором на борту (рис. 4), а также изменение параметров моделируемых в виртуальной среде атмосферных осадков (снег и дождь).

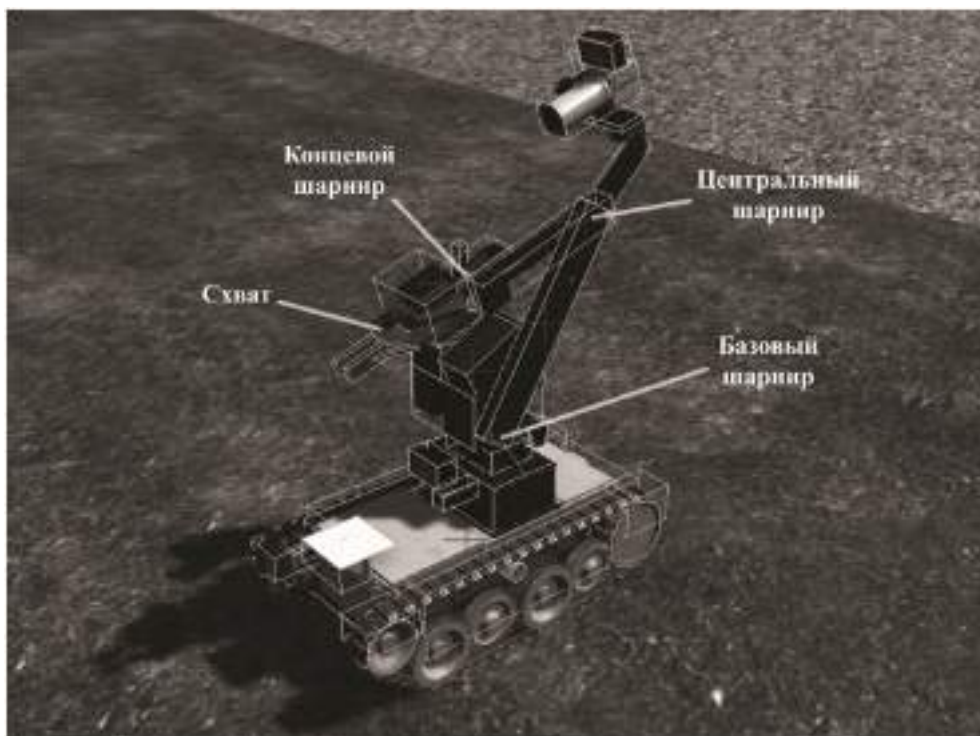


Рис. 4. Виртуальная модель робота MF-4 с манипулятором

Для решения этих задач с применением *Kinect* предлагается использовать сочетания жестов и поз обеих рук оператора. При этом левая рука отвечает:

- за включение или выключение реакции интерфейса на жесты и позы оператора;
- переключение между режимами управления роботом, манипулятором и параметрами погодных условий;
- выбор шарнира манипулятора, в котором будет осуществляться поворот, или вида атмосферных осадков, которым предполагается управлять.

Правая рука задает направление движений и поворотов модели робота и манипулятора, а также регулирует интенсивность осадков.

Чтобы реализовать указанные выше режимы управления, введем четыре универсальных жеста руки, начинающихся от позы $I_p(0, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$:

- $I_{g,up}$ – сгибание руки в локте до $\gamma = -50^\circ$;
- $I_{g,down}$ – разгибание руки в локте до $\gamma = 50^\circ$;
- $I_{g,left}$ – поворот руки в локте налево до $\theta = 50^\circ$;
- $I_{g,right}$ – поворот руки в локте направо до $\theta = -50^\circ$.

Все позы, входящие в эти жесты, отражены в табл. 1. Идентификаторы поз вычисляются по формулам (1), (2) с учетом $d_a = 20^\circ$.

Таблица 1

Позы управляющих жестов

| T_p | Поза I_p | Маска M_p |
|-------|--|-------------|
| 0 | $I_p(0, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | 0x00000000 |
| 1 | $I_p(0, 0^\circ, 0^\circ, 0^\circ, -25^\circ)$ | 0x00000000 |
| 2 | $I_p(0, 0^\circ, 0^\circ, 0^\circ, -50^\circ)$ | 0x00000000 |
| 3 | $I_p(0, 0^\circ, 0^\circ, 0^\circ, 25^\circ)$ | 0x00000000 |
| 4 | $I_p(0, 0^\circ, 0^\circ, 0^\circ, 50^\circ)$ | 0x00000000 |
| 5 | $I_p(0, 0^\circ, 0^\circ, -25^\circ, 0^\circ)$ | 0x00000000 |
| 6 | $I_p(0, 0^\circ, 0^\circ, -50^\circ, 0^\circ)$ | 0x00000000 |
| 7 | $I_p(0, 0^\circ, 0^\circ, 25^\circ, 0^\circ)$ | 0x00000000 |
| 8 | $I_p(0, 0^\circ, 0^\circ, 50^\circ, 0^\circ)$ | 0x00000000 |

Значения идентификаторов $I_{g, up}$, $I_{g, down}$, $I_{g, left}$, $I_{g, right}$ получим, подставляя номера T_p строк таблицы, соответствующие одному жесту, в формулу (4).

В табл. 2 сведены набор управляемых операций, выполняемых роботом MF-4 и его манипулятором, параметры моделируемых в виртуальной среде осадков и соответствующие им управляющие позы и жесты рук оператора.

Таблица 2

Операции и соответствующие им управляющие жесты и позы

| Операция | Поза/жест левой руки | Поза/жест правой руки |
|-------------------------------------|--|--|
| 1 | 2 | 3 |
| <i>Режимы интерфейса</i> | | |
| Включение/выключение интерфейса | $I_p(1, 0^\circ, -90^\circ, 0^\circ, 0^\circ)$ | – |
| Управление роботом MF-4 | $I_{g, up}$ | – |
| Управление манипулятором | $I_{g, down}$ | – |
| Управление параметрами осадков | $I_{g, left}$ | – |
| <i>Управление роботом MF-4</i> | | |
| Движение вперед | – | $I_{g, up}$ |
| Движение назад | – | $I_{g, down}$ |
| Поворот налево | – | $I_{g, left}$ |
| Поворот направо | – | $I_{g, right}$ |
| Остановка | – | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ |
| <i>Управление манипулятором</i> | | |
| Поворот в базовом шарнире налево | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 30^\circ)$ | $I_{g, left}$ |
| Поворот в базовом шарнире направо | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 30^\circ)$ | $I_{g, right}$ |
| Наклон в базовом шарнире вперед | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 30^\circ)$ | $I_{g, up}$ |
| Наклон в базовом шарнире назад | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 30^\circ)$ | $I_{g, down}$ |
| Поворот в центральном шарнире вверх | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | $I_{g, up}$ |
| Поворот в центральном шарнире вниз | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | $I_{g, down}$ |
| Поворот в концевом шарнире вверх | $I_p(1, 0^\circ, 0^\circ, 0^\circ, -30^\circ)$ | $I_{g, up}$ |
| Поворот в концевом шарнире вниз | $I_p(1, 0^\circ, 0^\circ, 0^\circ, -30^\circ)$ | $I_{g, down}$ |
| Вращение в концевом шарнире влево | $I_p(1, 0^\circ, 0^\circ, 0^\circ, -30^\circ)$ | $I_{g, left}$ |
| Вращение в концевом шарнире вправо | $I_p(1, 0^\circ, 0^\circ, 0^\circ, -30^\circ)$ | $I_{g, right}$ |
| Сжатие схвата | $I_p(1, 0^\circ, 0^\circ, 0^\circ, -60^\circ)$ | $I_{g, up}$ |

| 1 | 2 | 3 |
|---------------------------------------|--|--|
| Разжатие схвата | $I_p(1, 0^\circ, 0^\circ, 0^\circ, -60^\circ)$ | $I_{g, down}$ |
| Остановка | – | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ |
| <i>Управление параметрами осадков</i> | | |
| Усиление дождя | $I_p(0, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | $I_{g, up}$ |
| Ослабление дождя | $I_p(0, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | $I_{g, down}$ |
| Прекращение дождя | $I_p(0, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ |
| Усиление снега | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | $I_{g, up}$ |
| Ослабление снега | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | $I_{g, down}$ |
| Прекращение снега | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ | $I_p(1, 0^\circ, 0^\circ, 0^\circ, 0^\circ)$ |

Заключение

Предложенные в работе новые методы жестового интерфейса с помощью устройства *Kinect* позволят расширить доступные средства эргономичного управления объектами (в частности робототехническими средствами) как в системах виртуального окружения, так и в реальном пространстве.

Список литературы

1. Михайлюк, М. В. Эргономичный голосовой интерфейс управления антропоморфным роботом / М. В. Михайлюк, М. А. Торгашев, Д. В. Омельченко // Программные продукты и системы. – 2015. – № 4. – С. 75–78.
2. Нагапетян, В. Э. Бесконтактное управление роботизированной рукой посредством жестов человека / В. Э. Нагапетян, И. Л. Толмачев // Вестник Российского университета дружбы народов. Серия: Математика. Информатика. Физика. – 2014. – № 2. – С. 157–163.
3. Котюжанский, Л. А. Интерфейс бесконтактного управления / Л. А. Котюжанский // Фундаментальные исследования. – 2013. – № 4. – С. 44–48.
4. Qian, K. Developing a Gesture Based Remote Human-Robot Interaction System Using Kinect / K. Qian, J. Niu, H. Yang // International Journal of Smart Home. – 2013. – Vol. 7, № 4. – P. 203–208.
5. Мальцев, А. В. Реализация эргономичного интерфейса управления виртуальной моделью антропоморфного робота с использованием Kinect / А. В. Мальцев, М. В. Михайлюк // Программная инженерия. – 2015. – № 10. – С. 12–18.
6. Kinect API Overview. – URL: <https://msdn.microsoft.com/ru-ru/library/dn782033.aspx> (дата обращения: 25.04.2016).

Мальцев Андрей Валерьевич

кандидат физико-математических наук,
старший научный сотрудник,
Научно-исследовательский институт
системных исследований РАН
E-mail: avmaltcev@mail.ru

Maltsev Andrey Valerievich

candidate of physical and mathematical
sciences, senior researcher,
Scientific Research Institute
for System Analysis RAS

Михайлюк Михаил Васильевич

доктор физико-математических наук,
заведующий отделом,
Научно-исследовательский институт
системных исследований РАН
E-mail: mix@niisi.ras.ru

Mikhaylyuk Mikhail Vasilievich

doctor of physical and mathematical
sciences, head of department,
Scientific Research Institute
for System Analysis RAS

Тимохин Петр Юрьевич
научный сотрудник,
Научно-исследовательский институт
системных исследований РАН
E-mail: webpismo@yahoo.de

Timokhin Petr Yurievich
researcher,
Scientific Research Institute
for System Analysis RAS

УДК 004.5, 004.946

Мальцев, А. В.

Методы жестового управления трехмерными объектами и параметрами виртуальной среды с помощью Kinect / А. В. Мальцев, М. В. Михайлюк, П. Ю. Тимохин // Модели, системы, сети в экономике, технике, природе и обществе. – 2016. – № 3 (19). – С. 147–156.