

Эргономичный интерфейс управления виртуальной камерой с использованием Kinect¹

А.В. Мальцев

Аннотация. В работе предлагаются методы реализации бесконтактного интерфейса управления виртуальными камерами в визуализируемых на компьютерах трехмерных сценах. Рассматриваемые решения основаны на использовании устройства Microsoft Kinect для идентификации управляющих поз оператора, при которых осуществляется движение, поворот и трансфокация управляемой виртуальной камеры. Подробно описывается подход к цифровому кодированию и определению таких поз.

Ключевые слова: интерфейс, Kinect, виртуальная камера, трехмерная сцена, поза, управление.

Введение

Одной из важных задач при проектировании и реализации систем виртуальной реальности и имитационно-тренажерных комплексов является выбор интерфейсов управления визуализируемыми на компьютерах объектами трехмерной виртуальной среды. Такие интерфейсы подразделяются на контактные (требующие непосредственного прикосновения руками к элементам управления, таким как клавиатура, "мышь" и т.п.) и бесконтактные.

Контактные интерфейсы управления зачастую не являются интуитивно понятными и эргономичными (например, перемещение и поворот виртуальных объектов с помощью кнопок клавиатуры). Они требуют специального обучения оператора принципам работы с интерфейсом, но даже после обучения остаются неудобства в процессе его использования. Благодаря бурному развитию техники и техно-

логий виртуальной реальности всё большее распространение получают бесконтактные способы управления. Особое внимание уделяется использованию в основе таких интерфейсов устройства Kinect, созданного фирмой Microsoft сначала для игровой станции Xbox, а затем для персональных компьютеров с операционной системой Windows. Так, например, в [1] демонстрируются методы управления виртуальной моделью кисти руки, в [2, 3] предлагается подход к управлению в копирующем режиме трехмерной виртуальной моделью антропоморфного робота. Публикации [4, 5] описывают управление виртуальными объектами с помощью жестов, определяемых с использованием Kinect.

В данной работе предлагаются методы реализации бесконтактного типа интерфейса управления трехмерными виртуальными объектами с помощью поз тела человека. Эта задача рассматривается на примере управления

¹ Работа выполнена при поддержке РФФИ, грант № 15-07-04544.

виртуальной камерой. Разработанные решения основаны на использовании устройства Kinect для идентификации управляющих поз рук оператора, при которых осуществляется движение, поворот и трансфокация выбранной виртуальной камеры. Рассмотрим предлагаемые подходы подробнее.

1. Идентификация позы руки оператора

Управление виртуальной камерой будем осуществлять с помощью рук оператора. Позу руки определим как совокупность значений углов в сочленениях этой руки и состояния S ее ладони (1 – сжата в кулак, 0 – разжата). На рисунке 1 показано схематичное изображение левой руки человека, а также углы φ , ψ , θ , γ в плечевом и локтевом суставах, используемые в данной работе для задания ее позы. В представленном положении рука согнута в локте (плечо и предплечье перпендикулярны друг другу), прижата к туловищу и параллельна ему. Пусть все указанные углы при этом равны 0. Положительным поворотом (угол > 0) будем считать поворот звена руки против часовой стрелки, если смотреть с конца направляющего вектора Z_i оси, вокруг которой осуществляется этот поворот (Рис. 1).

Представим плечевой и локтевой суставы в виде системы иерархически связанных шарни-

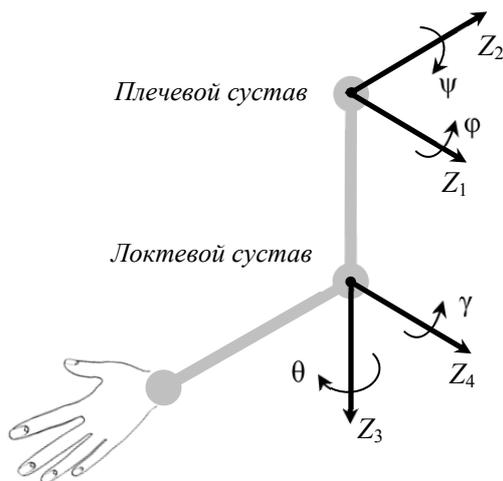


Рис. 1. Углы в сочленениях руки, используемые для задания ее позы

ров, каждый из которых имеет одну степень свободы (Рис. 2). Тогда текущие углы φ , ψ , θ и γ для руки оператора, находящегося в рабочей области устройства Kinect, можно определить на основе координат опорных точек его скелета, получаемых с помощью Kinect API (подробное описание данного API можно найти в [6]). Методы и алгоритмы решения этой задачи подробно изложены в [2]. Состояние ладони S также можно получить с помощью Kinect API при использовании устройства второго поколения. Параметры управляющих поз руки задаются вручную в соответствии с требуемыми положениями руки.

Хранение позы руки в виде набора углов и состояния ладони является неэффективным с точки зрения занимаемого объема памяти и неудобным для проведения сравнения нескольких поз. Поэтому в данной работе предлагается кодировать каждую позу. Суть такого кодирования состоит в записи состояния S ладони и углов φ , ψ , θ , γ в 32-битное беззнаковое целое число – *идентификатор позы*. При этом непрерывный диапазон $[-180^\circ, 180^\circ]$, охватывающий значения рассматриваемых углов, делится на открытые справа полуинтервалы одинаковой длины d_a (задается в зависимости от решаемой задачи, в нашем случае составляет 20°), последовательно нумеруемые от 0 до $n = (360/d_a) - 1$. Все углы, входящие в один полуинтервал,

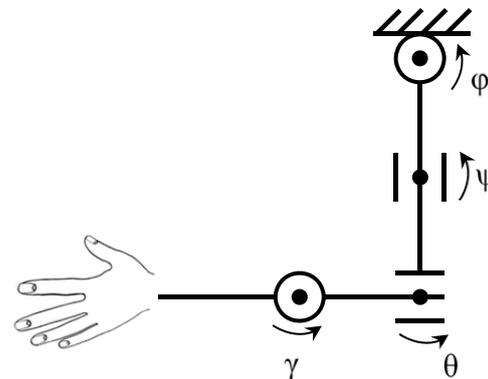


Рис. 2. Представление суставов в виде иерархии шарниров

Данные				S	I(φ)							I(ψ)				I(θ)							I(γ)											
Разряды	31	29	28	27	21							20	14				13	7							6	0								
Значения	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	0

Рис. 3. Структура 32-разрядного идентификатора позы руки

будем считать идентичными значению его середины (с погрешностью равной $\pm d_a/2$). 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. Значение 180° не достигается ни одним из углов φ , ψ , θ , γ в связи с особенностями строения человеческой руки, поэтому его можно не рассматривать. Записанные в фиксированной последовательности состояние S ладони, кодируемое одним битом (1 – сжата в кулак, 0 – разжата), и идентификаторы $I(\varphi)$, $I(\psi)$, $I(\theta)$, $I(\gamma)$ углов руки, представленные каждый 7 битами, составляют целочисленный идентификатор I_p позы руки (Рис. 3). Вычисление значения такого идентификатора будем производить по формуле

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

где \ll обозначает операцию побитового сдвига влево.

При решении задачи управления виртуальным объектом с помощью поз оператора можно определить такие управляющие позы, для которых один или несколько углов могут принимать любое значение. В таком случае во все разряды I_p , хранящие значения идентификаторов этих углов, предлагается записывать 1. Это можно сделать, выполнив операцию «побитовое ИЛИ» между I_p и целочисленной беззнаковой маской M_p , имеющей структуру, эквивалентную I_p , в которой каждый значащий разряд будет содержать 0, а незначащий – 1. Маску M_p необходимо хранить вместе с идентификатором позы для проведения в дальнейшем корректного сравнения поз.

Само сравнение текущей позы $I_{p,cur}$ руки оператора, определяемой с помощью Kinect, с одной из заранее заданных управляющих поз I_p , имеющей маску M_p , производится 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.

2. Управление виртуальной камерой

2.1. Управляющие позы

В данной работе предполагается, что виртуальная камера может совершать три операции: движение, поворот и трансфокация (изменение фокусного расстояния для масштабирования видимых посредством камеры объектов). Движения осуществляются вдоль осей X, Y, Z локальной системы координат VCS самой камеры, а повороты – вокруг этих осей.

Для решения задачи эргономичного управления виртуальной камерой с применением Kinect предлагается использовать позы левой и правой рук. При этом правая рука задействована для выбора операции, производимой камерой, а также оси локальной системы VCS, которая участвует в этой операции. В Табл. 1 сведены выполняемые камерой операции, используемые для них оси, соответствующие им управляющие позы правой руки, а также 32-битные идентификаторы I_p поз (вычисленные по формулам (1), (2) с учетом длины интервала $d_a = 20^\circ$) и маски M_p этих поз.

Левая рука определяет режим работы интерфейса (включение или выключение реакции интерфейса на позы оператора, пауза или управление камерой), а также в режиме «управление» задает параметры выполняемой в текущий момент операции (скорость и направление движения, направление поворота, параграф 2.2). В Табл. 2 представлены режимы интерфейса, соот-

Табл. 1. Управляющие позы правой руки

Операция	Ось	Поза						I_p	M_p
		рука, ладонь	S	φ	ψ	θ	γ		
Движение	X	горизонтально вправо, разжата	0	-	-90°	-	90°	0x0FE13F8D	0x0FE03F80
	Y	вертикально вверх, разжата	0	-175°	0°	-	90°	0x00027F8D	0x00003F80
	Z	горизонтально вперед, разжата	0	-90°	0°	-	90°	0x00827F8D	0x00003F80
Поворот	X	горизонтально вправо, сжата	1	-	-90°	-	90°	0x1FE13F8D	0x0FE03F80
	Y	вертикально вверх, сжата	1	-175°	0°	-	90°	0x10027F8D	0x00003F80
	Z	горизонтально вперед, сжата	1	-90°	0°	-	90°	0x10827F8D	0x00003F80
Трансфокация	-	свободна, сжата	1	-	-	-	-	0x1FFFFFFF	0x0FFFFFFF

Табл. 2. Управляющие позы левой руки

Режим интерфейса	Поза						I_p	M_p
	рука, ладонь	S	φ	ψ	θ	γ		
Включение / выключение	горизонтально влево, сжата	1	-	-90°	-	90°	0x1FE13F89	0x0FE03F80
Пауза	свободна, сжата	1	-	-	-	-	0x1FFFFFFF	0x0FFFFFFF
Управление	согнута в локте и прижата к туловищу, разжата	0	0°	0°	-	-	0x01227FFF	0x00003FFF

ветствующие им позы левой руки, идентификаторы I_p этих поз (вычислены по формулам (1) и (2)) и применяемые маски M_p .

Колонки I_p и M_p Табл. 1 и Табл. 2, а также действия камеры или интерфейса, соответствующие указанным в них управляющим позам, задают базы B_L и B_R управляющих поз для левой и правой рук оператора.

2.2. Вычисление параметров управляемой камеры

Принцип работы предлагаемого бесконтактного интерфейса управления виртуальной камерой проиллюстрирован на Рис. 4. Интерфейс, используя функции Kinect API, опрашивает устройство Kinect с частотой 30 раз в секунду и получает в ответ координаты опорных точек скелета оператора. На основе этих данных вычисляются текущие углы φ , ψ , θ , γ для каждой из рук оператора [2] и состояния S его ла-

дней (сжаты или разжаты). Определение соответствующих данным углам и состояниям ладоней идентификаторов $I_{p,L,cur}$ и $I_{p,R,cur}$ текущих поз левой и правой рук оператора производится по формулам (1) и (2). Далее осуществляется поиск $I_{p,L,cur}$ в базе B_L управляющих поз левой руки. При поиске столбец идентификаторов в B_L просматривается сверху вниз и для каждой строки проверяется равенство (3). При его соблюдении дальнейший поиск останавливается и из базы считывается режим интерфейса, связанный с найденной управляющей позой.

Необходимо отметить, что при наличии в базе поз какой-либо управляющей позы с незначительными углами возможна ситуация, когда также заданы позы, в которых эти углы имеют значение, а остальные углы эквивалентны (т.е. из множества поз выделены конкретные). В таком случае, при использовании описанного алгоритма поиска позы в базе, соответствующей

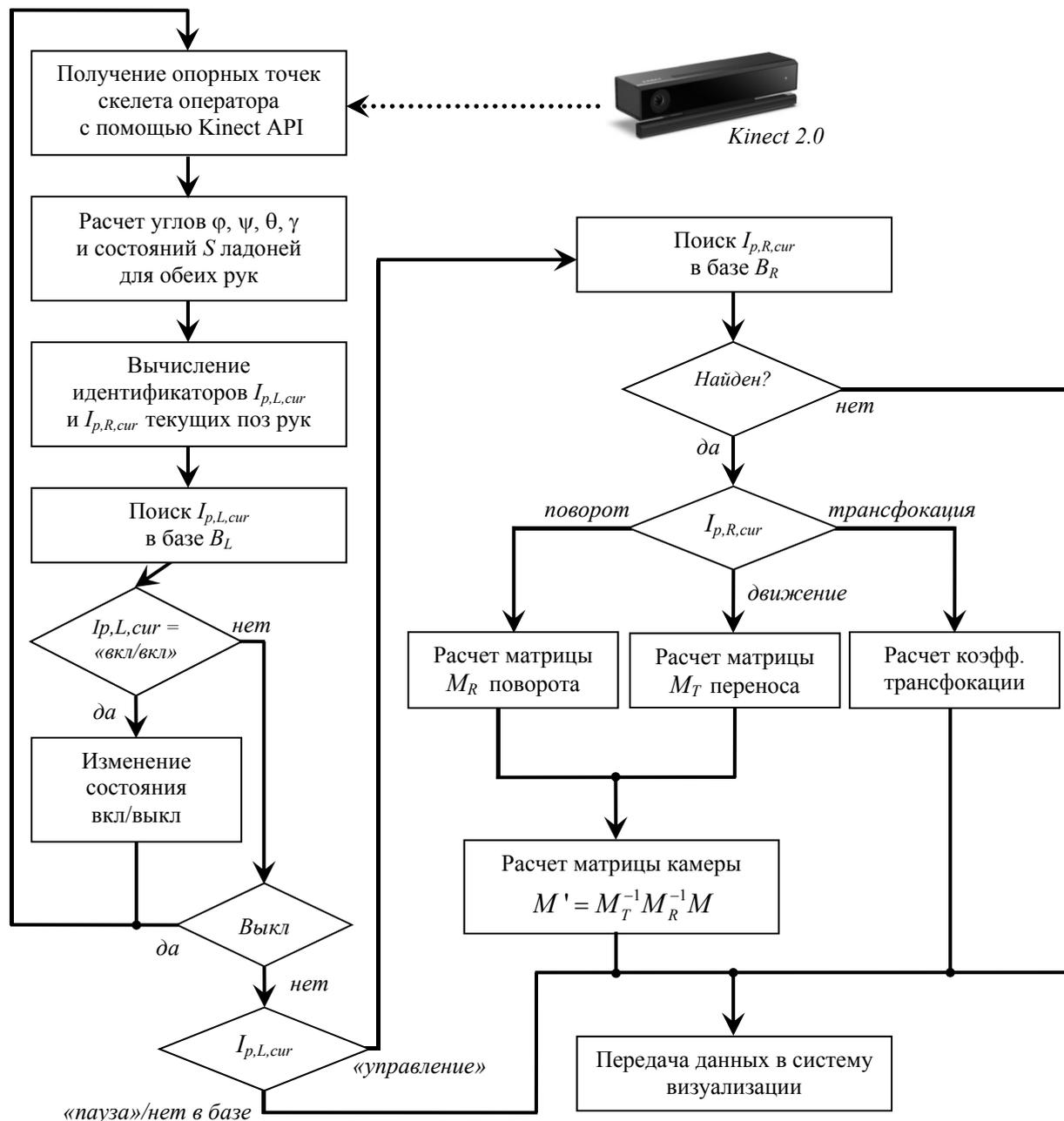


Рис. 4. Схема работы интерфейса управления виртуальной камерой

текущей позе, приоритет будет иметь та поза из базы, которая записана в строке, находящейся выше остальных. Поэтому позы с большим количеством значимых углов следует размещать в базе выше, чем позы с меньшим.

Если текущая поза левой руки – «включение/выключение», то соответствующее состояние интерфейса изменяется на противоположное (включается, если был выключен и

наоборот). После этого интерфейс начинает цикл работы сначала (то же самое происходит при любой другой позе левой руки, если текущее состояние интерфейса – «выключен»).

При идентификации позы «управление» и состоянии интерфейса «включен» выполняется расчет текущих параметров виртуальной камеры. Для этого производится поиск идентификатора $I_{p,R,cur}$ в базе B_R управляющих поз правой

руки (аналогично вышеописанному поиску в B_L). Если идентификатор найден, необходимо выполнить приписанное к нему в базе действие: движение, поворот или трансфокацию виртуальной камеры. Интерфейс постоянно хранит в оперативной памяти коэффициент k_t трансфокации камеры, а также матрицу M преобразования из мировой системы координат WCS в систему VCS, определяющую текущее положение и ориентацию камеры (в начальный момент матрица является идентичной, а $k_t = 1.0$). Увеличение k_t производится при сгибании ($\gamma < 0$), а уменьшение – при разгибании левой руки в локте ($\gamma > 0$) относительно исходного положения, представленного на Рис. 1. При движении камеры рассчитывается матрица M_T переноса [7] относительно ее текущего положения на некоторую величину Δs вдоль оси системы VCS, задаваемой идентификатором $I_{p,R,cur}$. Значение Δs определяется, исходя из угла γ для движения вдоль осей Y, Z (если $\gamma < 0$, $\Delta s > 0$ для Y и $\Delta s < 0$ для Z, при $\gamma > 0$ – наоборот) или угла θ при движении вдоль X (если $\theta > 0$, $\Delta s > 0$ и наоборот). При $\gamma = 0$ или $\theta = 0$ движения виртуальной камеры вдоль соответствующих осей не осуществляются, т.е. $\Delta s = 0$.

Аналогично, в случае поворота формируется матрица M_R поворота [7] на угол $\Delta\alpha$ вокруг нужной оси системы VCS. Значение $\Delta\alpha$ определяется, исходя из угла γ для поворота вокруг оси X (если $\gamma < 0$, $\Delta\alpha > 0$ и наоборот) или угла θ при повороте вокруг Y, Z (если $\theta < 0$, $\Delta\alpha > 0$ для Y и Z, при $\theta > 0$ – наоборот). При $\gamma = 0$ или $\theta = 0$ повороты виртуальной камеры вокруг соответствующих осей не осуществляются, т.е. $\Delta\alpha = 0$.

Для осуществления движений и поворотов матрица M пересчитывается по формуле $M' = M_T^{-1} M_R^{-1} M$. Полученная матрица и коэффициент трансфокации передаются по сети с использованием протокола UDP в систему визуализации.

В случае, если идентифицирована поза «пауза» или текущая поза левой/правой руки не найдена в базе, то в систему визуализации передаются значения M и k_t , сохраненные ранее, когда интерфейс находился в активном состоянии.

Заключение

На основе предложенных в работе методов и алгоритмов было создано программное приложение (Рис. 5), обеспечивающее бесконтактное управление виртуальной камерой в трехмерной сцене с помощью устройства Microsoft Kinect v2.0. Данное приложение вычисляет текущее положение, ориентацию и значение трансфокации (коэффициент масштабирования) виртуальной камеры на основе набора поз, принимаемых оператором, который находится в рабочей области Kinect. Полученные данные передаются по локальной сети в систему визуализации виртуальных сцен, где производится генерация и отображение видимого камерой изображения трехмерной сцены в соответствии с рассчитанными параметрами. Обмен данными между интерфейсом управления и системой визуализации осуществляется по протоколу UDP, с использованием несвязанных сокетов (unconnected sockets). Сетевой адрес и порт для отправки пакетов задаются пользователем в полях «Сервер IP» и «Порт» соответственно (Рис. 5). С целью обеспечения визуального контроля оператором работы интерфейса программа также отображает параметры текущего состояния самого интерфейса и управляемой с помощью него виртуальной камеры (скорость движения, значение трансфокации).

Для апробации разработанного приложения была использована система визуализации «GLView» [8], созданная в НИИСИ РАН.

Литература

1. Oikonomidis I., Kyriazis N., Argyros A. Efficient model based 3D tracking of hand articulations using Kinect // In Proceedings of the 22nd British Machine Vision Conference. – 2011. – P. 101.1-101.11.
2. Мальцев А.В., Михайлюк М.В. Реализация эргономичного интерфейса управления виртуальной моделью антропоморфного робота с использованием Kinect // Программная инженерия. – 2015. – № 10. – С. 12-18.
3. Лапта А.И., Мальцев А.В., Михайлюк М.В. Эргономичный интерфейс управления антропоморфным роботом // Труды 11-й Международной научно-практической конференции «Пилотируемые полеты в космос». – 2015. – С. 245-247.
4. Kim J.O., Kim M., Yoo K.H. Real-Time Hand Gesture-Based Interaction with Objects in 3D Virtual Environments // International Journal of Multimedia and Ubiquitous Engineering. – 2013. – Vol. 8, № 6. – P. 339-348.

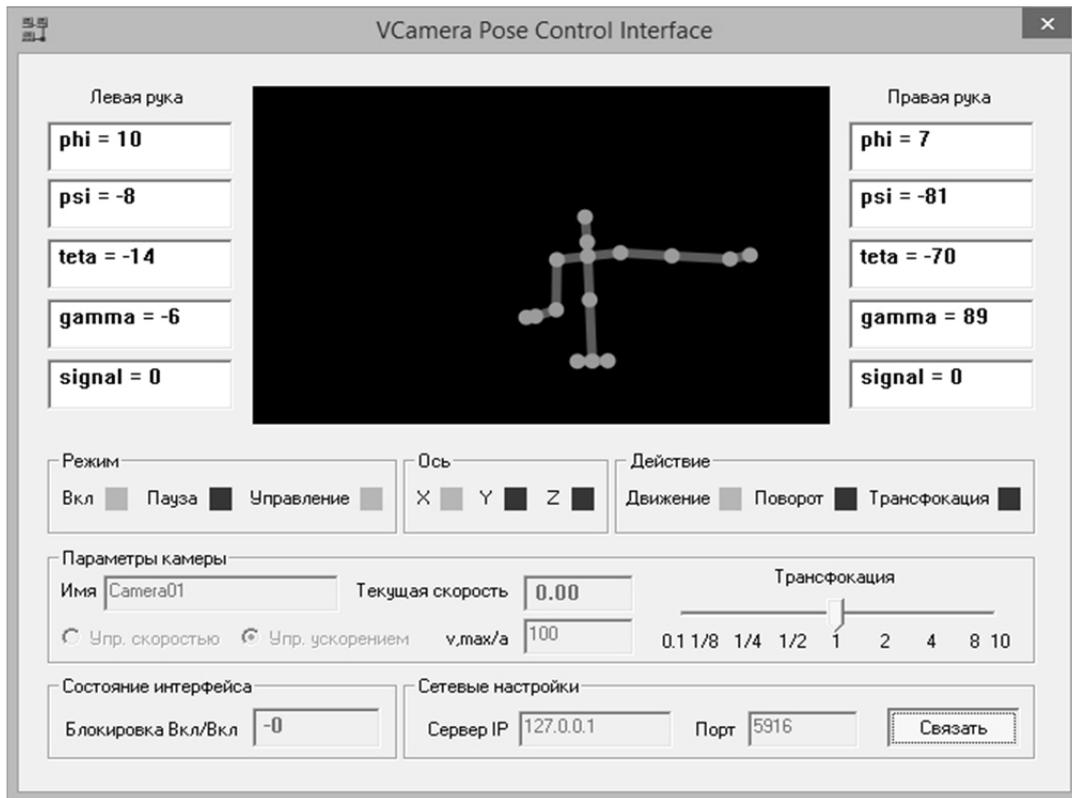


Рис. 5. Окно интерфейса управления виртуальной камерой

5. Soh J., Choi Y.J., Park Y., Yang HS. User-friendly 3D object manipulation gesture using kinect // Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry. – 2013. – P. 231-234.
6. Kinect API Overview // Url: <https://msdn.microsoft.com/ru-ru/library/dn782033.aspx> (дата обращения: 26.02.2016).
7. Михайлюк М.В. Основы компьютерной графики. – М.: МГТУ МИРЭА, 2011. – 156 с.
8. Михайлюк М.В., Торгашев М.А. Система визуализации «GLView» для имитационно-тренажерных комплексов и систем виртуального окружения // Труды Международной научной конференции Графикон-2015. – 2015. – С. 96-101.

Мальцев Андрей Валерьевич. Старший научный сотрудник Центра визуализации и спутниковых информационных технологий ФГУ ФНЦ НИИСИ РАН. Окончил Московский государственный институт радиотехники, электроники и автоматики в 2008 году. Кандидат физико-математических наук. Автор 43 печатных работ. Область научных интересов: компьютерная графика, системы виртуальной реальности, информационные технологии. E-mail: avmaltcev@mail.ru