

Вычисление углов Тейта-Брайана ориентации трекера HTC VIVE

И.П. Саблин¹, М.В. Михайлюк², Д.В. Омельченко,
Д.А. Кононов³, Д.М. Логинов

^{1,2}ФГУ ФНЦ НИИСИ РАН, Москва, Россия, ³ИПУ РАН, Москва, Россия,
¹sablinivan97@gmail.com, ²mix@niisi.ras.ru

Аннотация. В работе рассматривается подключение, настройка и технология работы с трекером HTC Vive Tracker 2.0. Также предложен метод перевода матрицы перехода, получаемой от трекера, в углы Тейта-Брайана, с учетом обхода проблемы «блокировки кардана».

Ключевые слова: система виртуального окружения, HTC Vive Tracker, SteamVR, OpenVR API, углы Тейта-Брайана

1. Введение

В настоящее время одним из важных направлений исследований является разработка технологий, методов и алгоритмов создания и использования виртуальной реальности (VR) при помощи компьютерных технологий.

VR технологии успешно применяются в различных сферах жизнедеятельности. Их используют для создания систем виртуального окружения, тренажеров различных видов (космических, авиационных, медицинских и др.), виртуальных лабораторий, игровой индустрии, виртуальных выставок и т.д. Системы виртуального окружения включают подсистему создания виртуальных трехмерных сцен, подсистему расчета динамики виртуальных объектов и процессов, подсистему визуализации и подсистему управления динамическим объектами и процессами. Эти системы можно расширить до комплексов виртуальной реальности, добавив специальные устройства: шлемы (для отображения стерео визуализации), контроллеры (для управления объектами), трекеры (для получения их координат и ориентаций и передачи их в виртуальную среду) и др. Если прикрепить трекер к реальному объекту, то позиция и ориентация этого объекта также может быть вычислена и передана в систему виртуального окружения. Одним из таких устройств является HTC VIVE Tracker.

Данный трекер используется для широкого круга задач. В работе [1] он рассматривается как бюджетная замена костюма отслеживания движений тела. В ходе исследований установлено, что задержка в передаче данных из трекера составляет около 7 миллисекунд, что удовлетворяет требованиям систем виртуального окружения. В статье [2] проверяется точность определения позиции данного трекера, установленного на

мобильном роботе. В ходе испытаний установлено, что HTC VIVE Tracker соответствует необходимой точности определения траектории робота. Авторы работы [3] использовали HTC VIVE Tracker в автоматизированном тесте на способность человека держать статический и динамический балансы (Berg Balance Scale или BBS) и показали его эффективность как инструмента скрининга, который обладает преимуществами низкой стоимости, высокой мобильности и простой настройки. В исследовании [4] на основе данных от трекеров были предложены два метода описания кинематики суставов нижних конечностей человека. Недорогая система позволяла собирать полные кинематические данные нижней конечности, получая непрерывные траектории трекера без трудоемкой калибровки системы. Полученные данные могут служить справочными данными для нормальной походки. Примером использования трекера для создания тренажеров может служить статья [5], в которой исследователи прикрепили трекеры HTC VIVE на симулятор велосипеда, и на основе передаваемых данных разработали тренажер для обучения безопасному передвижению в трафике.

Программный интерфейс трекера (OpenVR API) позволяет в любой момент времени получить его положение и ориентацию в виде матрицы перехода размером 3 x 4. Однако для многих задач более удобным является представление ориентации в виде углов Эйлера или Тейта-Брайана. Это позволяет, например, сократить число переменных, задающих ориентацию, с девяти (в матрице), до трех (в углах). HTC VIVE Trackers можно использовать и без дорогостоящего VR оборудования (шлема и контроллеров). Целью данной работы является настройка HTC VIVE Tracker без использования дополнительной VR гарнитуры и вычисление углов Тейта-

Брайана из матрицы перехода.

2. HTC VIVE Tracker

HTC VIVE Tracker – это трекер виртуальной реальности, разработанный компанией HTC (см. рис. 1.). В состав комплекта входит сам трекер и специальное передающее устройство Dongle. Время автономной работы трекера до 6 часов.



Рис. 1. HTC VIVE Tracker

Помимо этого, для использования трекера необходимы базовые станции HTC VIVE (см.



Рис. 2. HTC VIVE Base Station 1.0

рис. 2.).

Для отслеживания положения и ориентации в пространстве данный трекер использует метод Lighthouse tracking, в котором базовые станции являются так называемыми «маяками». Каждая станция состоит из двух инфракрасных лазеров и массива светодиодов. Линейный лазер в любой момент времени излучает направленный плоский световой треугольник. Первый лазер меняет направление этого треугольника вокруг

горизонтальной оси базовой станции, а второй – вокруг вертикальной оси. На самом же трекере располагаются фотодиоды. Вспышка светодиодов на базовой станции дает трекеру сигнал о начале временного интервала сканирования горизонтального лазера. Следующая вспышка дает сигнал для вертикального лазера. Если лазер попадает на фотодиод трекера, то трекер по времени этого попадания и ранее зафиксированного начала интервала вычисляет угол поворота плоскости лазера. Пересечение плоскостей от двух лазеров определяет прямую, а пересечение прямых от двух базовых станций определяет точку расположения фотодиода. Эти вычисления, проведенные для всех фотодиодов на трекере, позволяют точно и однозначно вычислить его позицию и ориентацию [6, 7].

3. Подключение и настройка HTC VIVE Tracker

Подключение трекера к компьютеру производится через среду SteamVR, которая обеспечивает работу приложений виртуальной реальности. Для установки данной среды необходимо найти в Интернет и скачать приложение Steam, которое является онлайн-сервисом цифрового распространения компьютерных игр и программ, разработанным компанией Valve в 2003 году. После установки и запуска приложения Steam необходимо в нем зарегистрировать пользователя, перейти во вкладку «МАГАЗИН», найти в поиске приложение SteamVR и установить его на компьютер.

Несмотря на то, что в данной работе используется только трекер, функционал Steam VR требует указание наличия шлема виртуальной реальности (HMD). Для этого необходимо перейти в директорию: `steam\steamapps\common\SteamVR\drivers\null\resources\settings`, открыть при помощи блокнота файл `default.vrsettings` и установить для параметра «enable» значение `true` (см. рис. 3). Аналогично, в директории `steam\steamapps\common\SteamVR\resources\settings` надо изменить файл `default.vrsettings`, а именно (см. рис. 4):

- установить значение `false` для параметра `"requireHmd"`;
- установить значение `null` для параметра `"forcedDriver"`;
- установить значение `true` для параметра `"activateMultipleDrivers"`.

Следующим шагом будет установка базовых станций. На официальном сайте компании VIVE есть ряд советов и рекомендаций для правильного расположения этих станций и подготовки помещения [8]. В частности, станции можно установить на треноги на некотором расстоянии

друг от друга, направить их на участок части пространства, отслеживаемого трекером, и включить в электрическую сеть. Тренога с базовой станцией показана на рис. 5. Синхронизация базовых станций может осуществляться с помощью кабеля или оптически. Так как мы рассматриваем оптическую синхронизацию, то необходимо для одной базовой станции выбрать

```
"steamvr": {
  "requireHmd": false,
  "forcedDriver": "null",
  "forcedHmd": "",
  "displayDebug": false,
  "debugProcessPipe": "",
  "enableDistortion": true,
  "displayDebugX": 0,
  "displayDebugY": 0,
  "allowDisplayLockedMode": false,
  "sendSystemButtonToAllApps": false,
  "loglevel": 3,
  "ipd": 0.063,
  "ipdOffset": 0.0,
  "background": "",
  "backgroundUseDomeProjection": false,
  "backgroundCameraHeight": 1.6,
  "backgroundDomeRadius": 0.0,
  "environment": "",
  "hdcpl4legacyCompatibility": false,
  "gridColor": "",
  "playAreaColor": "",
  "showStage": false,
  "drawTrackingReferences": true,
  "showGridCircles": true,
  "activateMultipleDrivers": true,
  "usingSpeakers": false,
```

Рис. 3. Фрагмент файла default.vrsettings.

```
"driver_null": {
  "enable": true,
  "loadPriority": -999,
  "serialNumber": "Null Serial Number",
  "modelNumber": "Null Model Number",
  "windowX": 0,
  "windowY": 0,
  "windowWidth": 2160,
  "windowHeight": 1200,
  "renderWidth": 1512,
  "renderHeight": 1680,
  "secondsFromVsyncToPhotons": 0.01111111,
  "displayFrequency": 90.0
}
```

Рис. 4. Фрагмент файла default.vrsettings.

режим работы «В», а для другой - «С» с помощью кнопки, расположенной на задней стороне.

Сам трекер автоматически подключается к адаптеру с электронным ключом (Dongle, см. рис. 6), который, в свою очередь, присоединяется к компьютеру с помощью USB кабеля. Для каждого трекера необходим свой dongle, поэтому использование нескольких трекеров может быть ограничено количеством USB портов. Для того, чтобы убедиться, что компьютер видит все подключенные устройства (включая искусственно воссозданный шлем), у SteamVR есть

индикатор состояния. Если все подключено верно, то на индикаторе будут гореть четыре иконки: для трекера, шлема и двух базовых станций.



Рис. 5. Базовая станция на треноге



Рис. 6. Адаптер

Для настройки трекера нужно переопределить его роль, так как автоматически он определяется системой как контролер правой руки. Для этого необходимо перейти в настройки SteamVR, далее на вкладку «контролеры» - «управление трекерами VIVE» (см. рис. 7) и указать, что трекер не имеет определенной роли (Роль трекера = ОТКЛЮЧЕНО).

Также для более удобной работы в будущем, приложение Steam можно поставить в автозагрузку и при дальнейшем использовании трекера, достаточно будет просто включить его, а приложение SteamVR запустится автоматически.

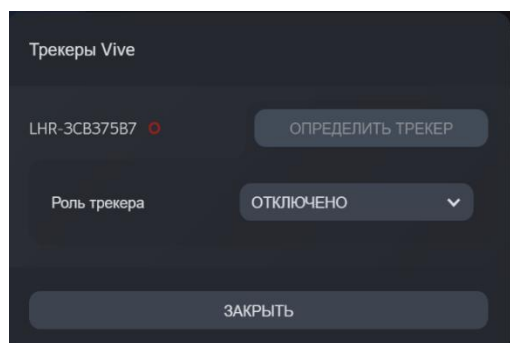


Рис. 7. Настройка роли трекера.

4. OpenVR

OpenVR – это открытый набор средств (SDK - software development kit и API - application programming interface) для последующей разработки приложений виртуальной реальности, предоставленный компанией Valve. При помощи OpenVR можно взаимодействовать с устройствами VR различных поставщиков [9]. Предоставленная библиотека является набором классов и виртуальных функций, написанных на языке C++. Вся дополнительная информация, а также сама библиотека OpenVR, находятся на официальном аккаунте корпорации Valve в GitHub [10], откуда ее можно скачать и разархивировать.

Для подключения данной библиотеки к проекту в среде Microsoft Visual Studio 2019 необходимо прописать путь к файлу `openvr.lib` в настройках Additional Include Directories, Additional Library Directories и Additional Dependencies. Кроме того, в одном из файлов проекта с помощью `#include` указывается директория, в которой расположен `openvr.lib`.

Для работы с системой виртуального окружения сначала необходимо инициализировать сам API с помощью функции `VR_Init()`. В OpenVR можно работать одновременно с `k_unMaxTrackedDeviceCount = 64` устройствами.

Для определения номера слота, в который система подключила наш трекер, необходимо в цикле от 0 до `k_unMaxTrackedDeviceCount-1` с помощью функции `IsTrackedDeviceConnected` проверить, что к *i*-му слоту подключено какое-то устройство, а с помощью функции `GetTrackedDeviceClass` – что это устройство является трекером и запомнить номер этого слота. По умолчанию в OpenVR первый слот занимает шлем виртуальной реальности, поэтому для

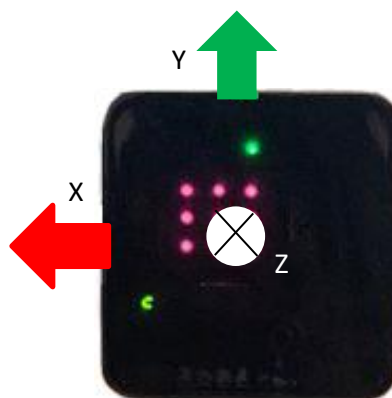


Рис. 8. Мировая система координат

определения слота трекера необходимо перебрать все остальные слоты. необходимо установить значение второго подключенного слота за трекером.

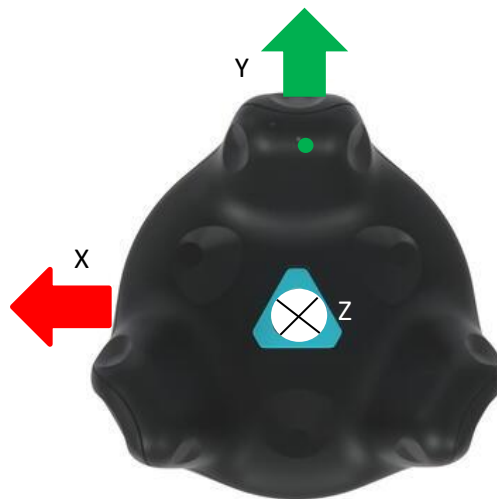


Рис. 9. Система координат трекера

В качестве мировой системы координат выбирается правосторонняя система координат базовой станции, работающей в режиме «С». Начало этой системы координат расположено в центре базовой станции, а оси направлены как показано на рис. 8. Начало правосторонней системы координат трекера расположено в его центре, а оси направлены как показано на рис. 9. Здесь важно, что ось *Y* направлена на индикатор зарядного устройства трекера. Для получения

матрицы преобразования из системы координат трекера в мировую систему координат можно использовать функцию `GetControllerStateWithPose`, первый параметр которой указывает тип выбранной системы координат, а четвертый параметр возвращает структуру типа `TrackedDevicePose_t`, содержащую матрицу перехода из системы координат трекера в эту систему координат.

5. Вычисление позиции и углов Тейта-Брайана трекера

Матрица перехода размерности 3×4 , имеет в OpenVR тип `HmdMatrix34_t`. Ее четвертый столбец задает координаты позиции трекера в метрах, а первые три столбца – матрицу поворота, из которой нам необходимо вычислить углы Тейта-Брайана.

Рассмотрим в трехмерном пространстве две правосторонние (левосторонние) ортонормированные системы координат K_1 и K_2 , центры которых совпадают, а оси направлены произвольно. Известно, что систему K_1 можно перевести с помощью трех поворотов вокруг ее осей в систему K_2 . Если повороты осуществляются вокруг произвольных двух осей (например, XYX), то углы поворотов называются углами Эйлера. В зависимости от выбора пар осей можно получить шесть различных наборов углов Эйлера. Если же повороты выполняются вокруг трех различных осей (например, ZXY), то говорят об углах Тейта-Брайана. Их также существует шесть видов, в зависимости от выбора порядка осей поворота [11]. Кроме того, в силу двойственности, поворот вокруг некоторой оси координат на произвольный угол ϕ эквивалентен повороту всего пространства вокруг этой оси на тот же угол, только в противоположную сторону, т.е. на угол $(-\phi)$. При этом система координат остается неподвижной. Здесь мы как раз и будем рассматривать поворот пространства вокруг осей ZXY именно в этом порядке, т.е. сначала поворот вокруг оси Z , затем вокруг оси X и, наконец, вокруг оси Y . Порядок поворота можно указать с помощью индекса, записав это в виде $Z_\alpha X_\beta Y_\gamma$.

Выведем матрицу поворота пространства вокруг оси Z . Для этого возьмем произвольный вектор V в плоскости XY . После поворота на угол α он перейдет в вектор V' (см. рис. 10).

$$\begin{aligned} \text{Рассчитаем координаты вектора } V': \\ x' = r \cos(\theta - \alpha) = r \cos \theta \cos \alpha + r \sin \theta \sin \alpha = \\ = x \cos \alpha + y \sin \alpha \\ y' = r \sin(\theta - \alpha) = r \sin \theta \cos \alpha - r \cos \theta \sin \alpha = \\ = y \cos \alpha - x \sin \alpha \\ z' = z \end{aligned}$$

или

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

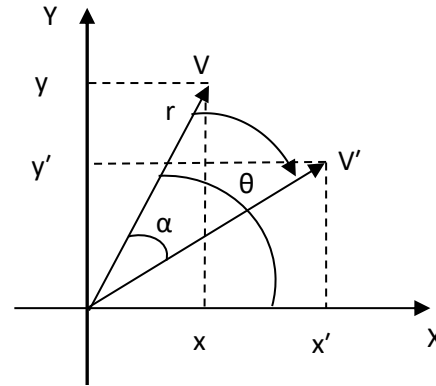


Рис. 10. Поворот вектора V в плоскости XY

Таким образом, матрица поворота пространства вокруг оси Z имеет вид

$$R_Z(\alpha) = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Аналогично для поворотов вокруг осей X и Y матрицы поворота будут следующими:

$$R_X(\beta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{pmatrix}$$

$$R_Y(\gamma) = \begin{pmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{pmatrix}$$

Перемножив эти матрицы, получим матрицу R поворота последовательно вокруг осей $Z_\alpha X_\beta Y_\gamma$:

$$R = R_Z R_X R_Y = \begin{pmatrix} c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -c_\beta s_\alpha & c_\alpha s_\gamma + c_\gamma s_\alpha s_\beta \\ c_\gamma s_\alpha + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta & s_\alpha s_\gamma - c_\alpha c_\gamma s_\beta \\ -c_\beta s_\gamma & s_\beta & c_\beta c_\gamma \end{pmatrix},$$

где c обозначает косинус, а s – синус соответствующего угла.

При вычислении углов поворота из этой матрицы возникает проблема в случае, когда угол β поворота вокруг оси X равен $\pm 90^\circ$ градусам. Эта ситуация носит название «блокировки кардана» [12]. В случае, когда $\beta = 90^\circ$ получаем $\cos \beta = 0$, $\sin \beta = 1$ и матрица поворота будет выглядеть следующим образом:

$$R_{\beta=90^\circ} = \begin{pmatrix} \cos(\alpha + \gamma) & 0 & \sin(\alpha + \gamma) \\ \sin(\alpha + \gamma) & 0 & -\cos(\alpha + \gamma) \\ 0 & 1 & 0 \end{pmatrix}$$

Видно, что матрица зависит от суммы углов, поэтому невозможно однозначно вычислить каждый из них. В этом случае один из углов (например, α) можно взять равным нулю, а второй угол вычислить. Мы рассмотрим отдельно три случая: когда угол β близок к 90° , когда он близок к (-90°) и когда угол принимает другие значения. Близость угла к $\pm 90^\circ$ можно проверить с помощью достаточно малого значения ϵ . Элементы матрицы R будем указывать с помощью индексов: первый индекс задает номер строки (от 1 до 3), второй – номер столбца (также от 1 до 3). Мы будем использовать также функцию $\text{atan2}(x, y)$, см. [13] Таким образом, алгоритм будет иметь следующий вид:

Начало.

Если $(1 - R_{32} < \epsilon)$, то // угол β близок к 90° ,

$\alpha = 0$;

$\beta = \pi/2$;

$\gamma = \text{atan2}(R_{21}, R_{11})$;

иначе, если $(1 + R_{32} < \epsilon)$, то

// угол β близок к (-90°)

$\alpha = 0$;

$\beta = -\pi/2$;

$\gamma = \text{atan2}(-R_{21}, R_{11})$;

иначе

$\alpha = \text{atan2}(-R_{12}, R_{22})$;

$\beta = \text{asinf}(R_{32})$;

$\gamma = \text{atan2}(-R_{31}, R_{33})$;

Конец.

6. Заключение

В данной работе рассматривается подключение, настройка и технология работы с устройством HTC VIVE Tracker, а также предложен алгоритм вычисления углов Тейта-Брайана ориентации трекера из матрицы перехода, получаемой с помощью библиотеки Open VR. При этом разрешается проблема «блокировки кардана».

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН «Проведение фундаментальных научных исследований (47 ГП)» по теме № FNEF-2022-0012 «Системы виртуального окружения: технологии, методы и алгоритмы математического моделирования и визуализации».

Calculation of Tait–Bryan Angles of HTC VIVE Tracker Orientation

I.P. Sablin, M.V. Mikhaylyuk, D.V. Omelchenko, D.A. Kononov, D.M. Loginov

Abstract. The paper considers the connection, configuration and technology of working with the HTC Vive Tracker 2.0. A method is also proposed for translating the transition matrix received from the tracker into Tait–Bryan angles, taking into account the "gimbal lock" problem.

Keywords: Virtual environment system, HTC Vive Tracker, Steam VR, OpenVR API, Tait–Bryan angles

Литература

1. Polona Caserman, Augusto Garcia-Agundez, Robert Konrad, Stefan Göbel, Ralf Steinmetz. Real-time body tracking in virtual reality using a Vive tracker. Virtual Reality vol. 23, p. 155–168 (2019). <https://link.springer.com/article/10.1007/s10055-018-0374-z> (дата обращения: 25.11.2021).
2. Yong Wang, Peng Tian, Yu Zhou, Mao Mao Zhu, Qing Chen and Chang Zhang. Using VIVE Tracker to Detect the Trajectory of Mobile Robots. EasyChair Preprint № 6241, August 5, 2021. https://www.easychair.org/publications/preprint_open/bt8n (дата обращения: 25.11.2021).
3. Susanne M. van der Veen, James S. Thomas. A Pilot Study Quantifying Center of Mass Trajectory

during Dynamic Balance Tasks Using an HTC Vive Tracker Fixed to the Pelvis. *Sensors*, vol. 21, (2021). <https://www.mdpi.com/1424-8220/21/23/8034> (дата обращения: 25.11.2021).

4. Magdalena Żuk, Magdalena Wojtków, Michał Popek, Jakub Mazur, Katarzyna Bulińska. Three-dimensional gait analysis using a virtual reality tracking system. *Measurement*, vol. 188 (2022). <https://www.sciencedirect.com/science/article/pii/S0263224121014974> (дата обращения: 25.11.2021).

5. Tamara von Sawitzky, Thomas Grauschopf, Andreas Riener. The Next Stage of Road Traffic Education: A Mixed Reality Bicycle Simulator to Improve Cyclist Safety. https://dl.gi.de/bitstream/handle/20.500.12116/33434/GI_VRAR_20_paper_31.pdf?sequence=1&isAllowed=y (дата обращения: 30.11.2021).

6. Lighthouse tracking. <https://shazoo.ru/2015/05/18/30236/kak-rabotaet-sistema-otslezhivaniya-polozeniya-v-prostranstve-ot-valve-lighthouse> (дата обращения: 30.11.2021).

7. Lighthouse tracking. <https://habr.com/ru/post/369553/> (дата обращения: 30.11.2021).

8. Инструкция по установке базовых станций. https://www.vive.com/eu/support/vive-pro/category_howto/tips-for-setting-up-the-base-stations.html (дата обращения: 30.11.2021).

9. Документация OpenVR. <https://partner.steamgames.com/doc/features/steamvr/openvr> (дата обращения: 30.11.2021).

10. GitHub OpenVR. <https://github.com/ValveSoftware/openvr> (дата обращения: 30.11.2021)

11. Углы Эйлера. https://en.wikipedia.org/wiki/Euler_angles (дата обращения: 30.11.2021)

12. Блокировка кардана. https://ru.wikipedia.org/wiki/Складывание_рамок (дата обращения: 30.11.2021)

13. Функция atan2. <https://en.wikipedia.org/wiki/Atan2> (дата обращения: 30.11.2021)