

ПРОГРАММНО-АППАРАТНАЯ СИСТЕМА ЗАХВАТА ДВИЖЕНИЙ РУК

Видничук В. Н., Альшевский А. В., Бысов С. А.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: Vidnichuk@bsuir.by, nexus.alekseishshevski@gmail.com, sergbysov@gmail.com

Предложена модель системы захвата движений на основе датчиков положения в пространстве. Описан механизм получения данных и их дальнейшей обработки для использования в программном обеспечении, анализирующем параметры датчиков и обрабатывающем значения для использования программным обеспечением виртуализации движения руки человека.

ВВЕДЕНИЕ

На данный момент разработка новых интерфейсов взаимодействия пользователя с компьютерными системами является одним из ведущих направлений в области компьютерных технологий (КТ). Существующие решения в области устройств ввода, такие как клавиатура или компьютерная мышь, постепенно перестают удовлетворять потребностям быстро растущих рынков, использующих КТ. В связи с бурным развитием технологий виртуальной реальности как никогда проявляется необходимость в введении новых методов взаимодействия с виртуальными моделями. Одной из существующих задач в области цифрового захвата движений является распознавание движений кистей рук, т.к. руки человека являются самыми подвижными конечностями, что вызывает существенные сложности в построении точной модели руки пользователя в реальном времени для использования в системах виртуальной реальности.

ПРЕДЛАГАЕМАЯ РЕАЛИЗАЦИЯ

Одной из возможных реализации такой системы является использование датчиков, способных определять положение в пространстве каждой отдельно взятой точки кисти руки, что позволит точно проецировать не только хватательные движения, но и мелкую моторику руки. Подобные устройства уже существуют, но на данный момент их точность весьма ограничена ввиду того, что количество точек считывания существенно ограничено.

Цель данной работы — реализовать устройство ввода, способное проецировать модель кисти человека на основе датчиков положения в пространстве, установленных так, чтобы отслеживать изменения в положении фаланг пальцев в пространстве. Точность проецирования будет обеспечена использованием датчиков на каждой фаланге, а также на запястье, предплечье и плече. Определив точное расстояние между датчиками, установленными на руке, можно будет построить проекцию руки человека в пространстве, основываясь на анатомических параметрах каж-

дого из пальцев и положении каждой из точек в пространстве. На основе крена, рысканья и тангажа на каждой из верхних плоскостей фаланг и известного расстояния между точками считывания программное обеспечение будет иметь возможность рассчитать трехмерную модель положения кисти в пространстве, после чего эти данные могут быть использованы сторонним программным обеспечением для визуализации данной модели.

Крен - отклонение плоскости симметрии отслеживаемого объекта по вертикали, что можно трактовать как сгиб фаланги пальца. Тангаж - отклонение плоскости симметрии отслеживаемого объекта по горизонтали, что можно трактовать как поворот самого пальца, который может являться следствием поворота руки. Рысканье - угловое движение объекта относительно вертикально оси. Это можно трактовать как отклонение движения руки в горизонтальной плоскости.

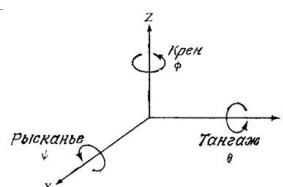


Рис. 1 – Крен, тангаж, рысканье

Для реализации данного проекта будет использован следующий набор аппаратных решений:

- Датчики MPU-9250, достаточно компактные и точные, будут использоваться для считывания положения верхних плоскостей фаланг пальцев в пространстве.
- Устройства первоначальной обработки данных на основе FPGA (ПЛИС). Собственная архитектура, созданная для конкретных задач, существенно повысит скорость работы всей системы. А также благодаря возможности перепрограммирования схемы минорное обновление устройства можно будет производить без демонтажа и замены уже существующих узлов.

- Одноплатный микрокомпьютер (напр. Raspberry Pi), чьей задачей будет анализ и подготовка полученных данных, а также отправка полученных значений ПО, выполняющему построение 3D модели в виртуальном пространстве.

MPU-9250 - девятиосевой датчик положения в пространстве. Обладая трёхосевыми магнитометром, гироскопом и акселерометром, данный датчик позволит решить множество проблем связанных с последующей фильтрацией данных, т.к. имеющийся в корпусе магнетометр позволит решить проблему дрейфа рысканья.

Передача данных между датчиками и фильтрующей FPGA-схемой будет осуществляться посредством интерфейса I2C, который уже реализован в датчиках MPU-9250. Полученные сырые данные будут обработаны с помощью реализованного на аппаратном уровне Фильтра Махони. Формула фильтра Махони выглядит следующим образом:

$$\dot{\mathbf{C}} = \mathbf{C}(\boldsymbol{\Omega} + \boldsymbol{\Omega}')$$

$$\boldsymbol{\omega}' = k_g \hat{\mathbf{g}}_{xyz} \times \mathbf{C}^T \mathbf{g}_{NED} + k_m \hat{\mathbf{m}}_{xyz} \times \mathbf{C}^T \mathbf{m}_{NED}$$

Где слагаемое в первой части является векторным произведением ускорения свободного падения на истинное ускорение. При этом второй множитель требуется преобразовать к той же системе координат, что и первый. Второе слагаемое - аналогичное произведение векторов магнитных полей. Благодаря знанию ориентации двух неколлинеарных векторов в разных системах мы можем точно установить взаимную ориентацию этих систем.

Задача фильтра будет заключаться в коррекции дрейфа тангажа, крена и рысканья для каждого из датчиков. Также благодаря фильтру можно будет нивелировать низкочастотный дрейз датчиков. Далее данные при помощи интерфейса UART будут переданы на один из вспомогательных микрокомпьютеров, чьей задачей является накопление, окончательная фильтрация и отправка данных на центральный микрокомпьютер, через который осуществляется связь со средством визуализации.

Согласно предварительным расчётам, для передачи данных между узлами системы не потребуется скорости соединения более 30 КБайт/сек, что полностью покрывается выбранными интерфейсами. Выбор I2C обусловлен наличием такового в корпусе датчиков MPU-9250, благодаря чему можно избежать монтажа дополнительных радиоэлектронных схем в ограниченных размерах. Интерфейс UART обладает достаточной простотой реализации с точки зрения FPGA-программирования, а также присутствует в одноплатном микрокомпьютере Raspberry Pi любой модификации, что дополнительно облегчит разработку.

Далее для визуализации полученных данных о движении датчиков будет использован инструментарий Unity 3D. Данное программное обеспечение обладает весомыми преимуществами для нашего проекта: наличием визуальной среды разработки и возможностью выполнения на различных платформах. Также используется модульная система компонентов, с помощью которой происходит конструирование виртуальных функциональных объектов. В отличие от механизмов наследования, объекты в Unity создаются посредством функциональных блоков, а не помещения в узлы дерева наследования. Такой подход ускоряет создание прототипов.

Алгоритм работы системы можно представить следующим образом:

- инициализация соединений и компонентов, а также их калибровка;
- поочередное считывание данных из гироскоп-акселерометров;
- предварительная фильтрация сырых данных;
- передача данных на микроконтроллер;
- отправка данных в концентратор;
- преобразование данных в сферические координаты и дополнительная фильтрация при необходимости;
- передача данных в среду трёхмерного отображения;
- интерпретация полученных сферических координат в положение виртуальных объектов в виртуальном пространстве.

ЗАКЛЮЧЕНИЕ

В итоге будет получена система, которая может снимать данные о положении рук человека в реальном пространстве и визуализировать их в виртуальной среде в реальном времени. Данную систему можно использовать в широком спектре задач: например, в кинопроизводстве (в качестве более точной замены Motion Capture), или в промышленности и медицине (для осуществления управления манипуляторами или для реабилитации).

СПИСОК ЛИТЕРАТУРЫ

1. Peterson, E. Developing tamper resistant designs with Xilinx Virtex-6 and 7 series FPGAs, San Jose, CA, USA, Oct. 2013.
2. Львовский, С. М. Набор и вёрстка в системе LaTeX / С. М. Львовский // Издательство: МЦНМО, 2006. – 448 с.
3. MPU9250 Datasheet [Электронный ресурс]. Режим доступа: <https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>
4. Unity User Manual [Электронный ресурс]. Режим доступа: <https://docs.unity3d.com/Manual/index.html>
5. Raspberry Pi Documentation [Электронный ресурс]. Режим доступа: <https://www.raspberrypi.org/documentation/>