

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ИМИТАТОРА ВОЗДУШНОЙ ОБСТАНОВКИ С ПОМОЩЬЮ ГРАФИЧЕСКИХ ИНТЕРФЕЙСОВ OPENGL И UNITY3D

Солонар А.С., Гуцев Р.А., Цуприк С.В., Бабарека А.С.

УО «Военная академия Республики Беларусь», Минск, Беларусь, sergey.tsuprik@mail.ru

На сегодняшний день оптико-локационные системы находят все большее применение в современной технике. Возможность принимать и обрабатывать в реальном масштабе времени большой объем информации выдвигает оптико-локационные системы на первые место среди сложных систем автоматического обнаружения и распознавания изображений [8,9].

Одной из проблем, возникающих при разработке оптико-локационных систем является сложность проверки эффективности и отладки алгоритмов их работы. Эти проблемы могут решаться в процессе лабораторных исследований или в ходе полевых испытаний. Поскольку проводить полевые испытания не всегда возможно, чаще всего отладку алгоритмов осуществляют в лаборатории. В лабораторных условиях проверка работоспособности алгоритмов и их оптимизация может производиться как по неподвижным, так и по движущимся целям в результате обработки видеосъемки. Для неподвижных объектов это допустимо. Проблема возникает при сопровождении движущихся объектов, поскольку центр оптической системы перемещается, а закон перемещения не известен.

Формирования адекватного входного воздействия для оптико-локационной системы является актуальной задачей. В ходе ее решения не тривиальным вопросом является выбор графического интерфейса для построения трехмерной сцены и формирования цифрового изображения. Одним наиболее распространенным, для работы с двумерной и трехмерной компьютерной графикой, является графический кроссплатформенный программный интерфейс Open Graphics Library (OpenGL). Использование OpenGL, позволяет разработчику максимально использовать возможности трехмерной графики в своих проектах, как элемента визуализации.

Развитие современного программного обеспечения привело к появлению новых графических интерфейсов, из которых выделяется кроссплатформенный интерфейс Unity3D. В отличие от OpenGL, интерфейс Unity3D представляет собой визуальную среду проектирования, что позволяет относительно просто формировать трехмерные сцены и их анимацию. Однако интерфейс Unity3D нельзя непосредственно встроить в проект стороннего разработчика. Непосредственное использование Unity3D требует написания отдельной программы визуализации и средств работы с ней.

Целью доклада является: 1) предложить обобщенную структуру имитатора воздушной обстановки, позволяющего в реальном масштабе времени сформировать адекватное входное воздействие для оптико-локационной системы в виде цифрового изображения; 2) рассмотреть особенности формирования цифрового изображения при помощи графического интерфейса OpenGL; 3) рассмотреть особенности формирования цифрового изображения при помощи графического интерфейса Unity3D.

Обобщенная структура имитатора воздушной обстановки. Для определения облика будущего имитатора воздушной обстановки, необходимо определить его структуру. На рисунке 1 предложена обобщенная структура имитатора воздушной обстановки. Она включает в себя: неподвижную оптико-локационную систему, модель входного воздействия, проектор и экран. Идея заключается в имитации вращения центра оптической системы за счет перемещения изображения на экране относительно неподвижной оптико-локационной системы.

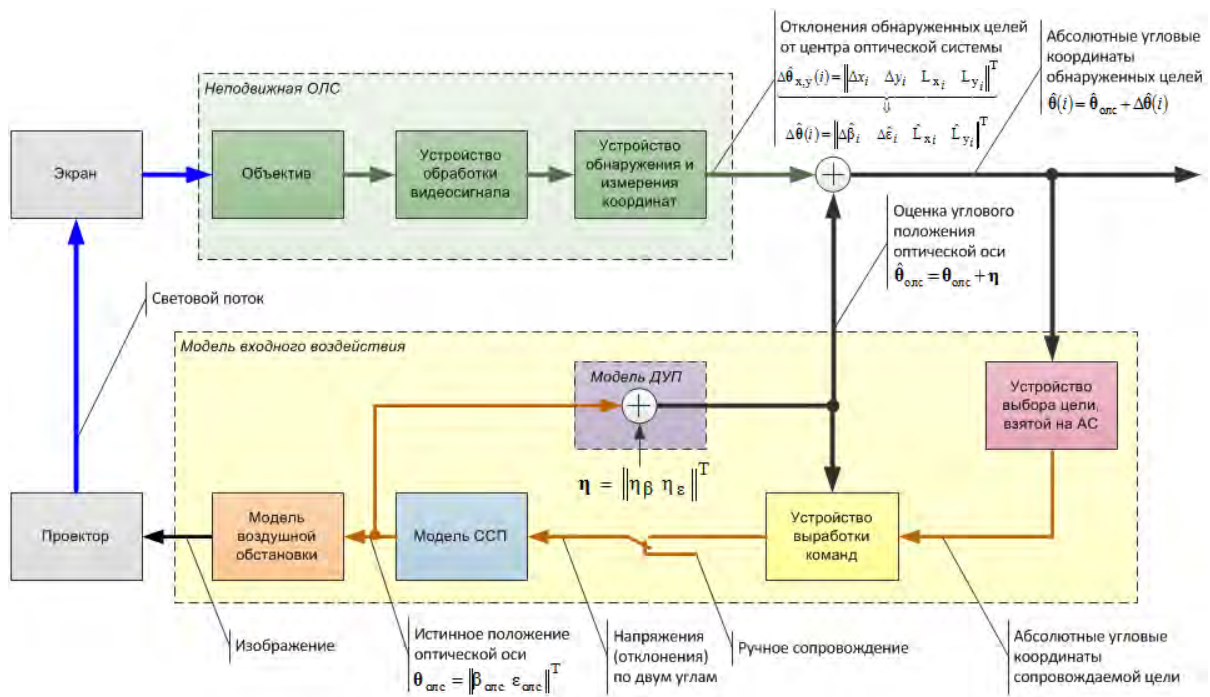


Рисунок 1 – Обобщенная структура аппаратно-программного имитатора

Структурно в модель входного воздействия входят устройство выбора цели, взятой на сопровождение, устройство выработки команд, модель силового следящего привода, модель датчика угла поворота и модель воздушной обстановки [7]. Входным воздействием является световой поток, сформированный при помощи проектора и экрана. Световой поток представляет собой синтезированное цифровое изображение, сформированное моделью воздушной обстановки. Для имитации вращения оптической системы используется несколько математических моделей силового следящего привода. Для оценки истинного положения оптической оси используются модели датчиков угла поворота. Они формируют оценки текущего положения привода $\hat{\theta}_{опс}$ путем добавления к истинному положению $\theta_{опс}$ белого гауссовского шума η с нулевым математическим ожиданием и заданной дисперсией $D_{дуп_\beta}, D_{дуп_\varepsilon}$ независимо по каждой угловой координате [2,3]. В результате на выходе неподвижной оптико-локационной системы формируются M оценок абсолютных угловых координат целей $\hat{\theta}(i), i = \overline{1, M}$ путем добавления к $\Delta \hat{\theta}(i)$ оценки углового положения центра оптической системы $\hat{\theta}_{опс}$. Одна из оценок может быть выбрана для взятия на автоматическое сопровождение при помощи соответствующего устройства. Результатом работы имитатора воздушной обстановки является синтезированное цифровое изображение для различных условий наблюдения, сформированное при помощи светового потока проектора и экрана.

Модель воздушной обстановки представляет собой синтезированное цифровое изображение. Для его формирования может использоваться программный интерфейс OpenGL (Open Graphics Library), который является графическим стандартом в области компьютерной графики.

Особенности формирования цифрового изображения при помощи графического интерфейса OpenGL. На рисунке 2 представлена структура модели воздушной обстановки, разработанная при помощи OpenGL. Ключевым элементом является блок формирования сцены, в котором формируется сценарий воздушной обстановки. Выбор требуемой модели и установка параметров движения осуществляется блоком загрузки и задания параметров модели. Изменений условий обстановки наблюдения осуществляется с помощью блоков управления контрастностью и окружающая среда. Использование технологии OpenGL, предоставляет возможность разработчику самому определять структуру модели и непосредственно интегрировать ее в свой проект.

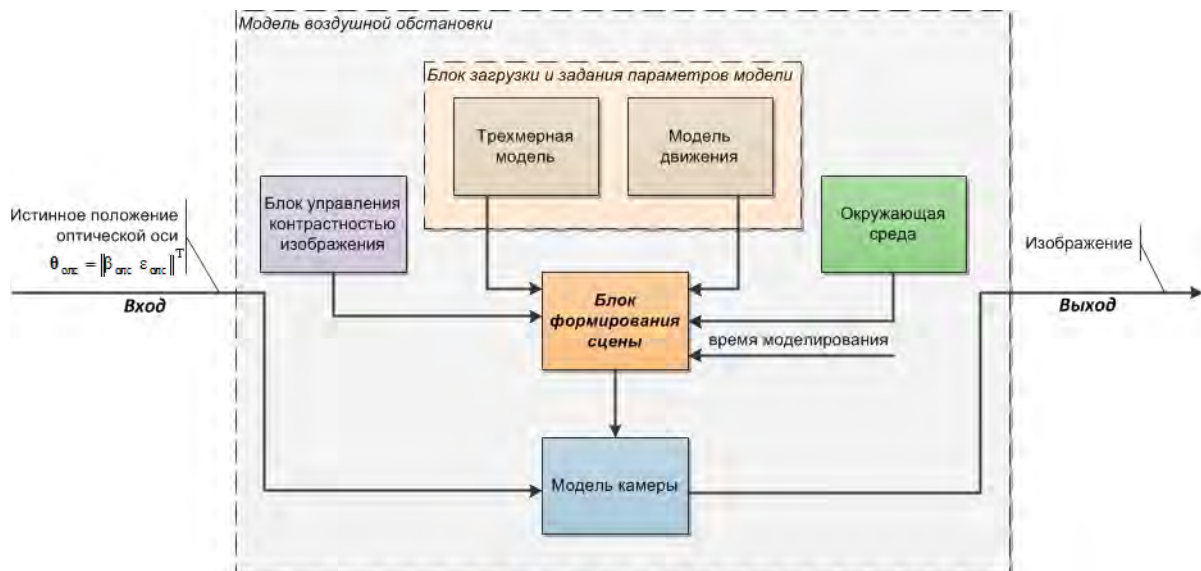


Рисунок 2 – Структура модели воздушной обстановки, разработанная на основе графического интерфейса OpenGL

OpenGL – это программный интерфейс, который включает в себя набор различных команд, позволяющих определять различные объекты и осуществлять их отображение на экране [1]. Под определением объекта в OpenGL понимается задание местоположения в трехмерном пространстве, задание свойств объекта (цвет, текстуры, материал, и.т.д.), положение наблюдателя. Необходимо отметить, что OpenGL осуществляет только отображение и не поддерживает какие-либо периферийные устройства, такие как клавиатуры и мышь, а также менеджеров окон. Программист должен самостоятельно обеспечивать взаимодействие периферийных устройств с OpenGL. Несмотря на хорошо продуманную внутреннюю структуру и довольно простой процедурный интерфейс, с помощью OpenGL можно создавать сложные и мощные программные комплексы, затрачивая при этом минимальное время по сравнению с другими графическими библиотеками. Мультиплатформенность позволяет без труда переносить программное обеспечение с одной операционной системы в другую. OpenGL предоставляет в распоряжение всю мощь аппаратных возможностей, которые имеются на данном компьютере и при написании программ не нужно беспокоиться о конкретных деталях используемого оборудования. Однако написание сложных приложений с помощью OpenGL, требует от разработчика хороших знаний языков программирования и достаточно больших временных затрат на разработку и отладку трехмерных сцен, что значительно усложняет использование OpenGL.

Особенности формирования цифрового изображения при помощи графического интерфейса Unity3D. Другим, не менее эффективным инструментом формирования цифрового изображения является графический интерфейс Unity3D [4,5,6]. Он имеет достаточно простой интерфейс, состоящий из различных окон, благодаря чему можно

производить отладку сцены прямо в редакторе (визуальной среде разработки). Визуализация трехмерной сцены и ее анимации в Unity3D формируется в виде законченного исполняемого приложения. Сложность заключается в способе взаимодействия программы имитатора воздушной обстановки (рис.1) с модулем визуализации написанном в Unity3D, как отдельным приложением.

На рисунке 3 представлена обобщённая структура модели воздушной обстановки взаимодействующая модулем визуализации написанном в Unity3D при помощи сетевого протокола TCP/IP.

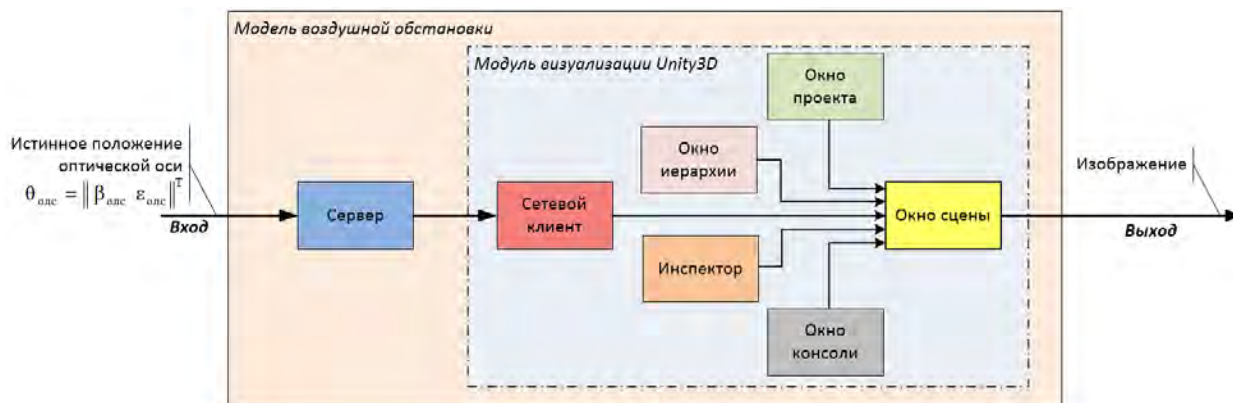


Рисунок 3 – Обобщённая структура модели воздушной обстановки взаимодействующая модулем визуализации написанном в Unity3D

Интерфейс Unity3D представляет собой набор функциональных элементов в виде отдельных окон, позволяющих настраивать трехмерную сцену требуемым образом. В отличие от графического интерфейса OpenGL, Unity3D легко интегрируемо с такими программными продуктами как 3DS Max и Adobe Photoshop. Трехмерные модели, текстуры и материалы легко импортируются в Unity3D, что значительно облегчает и ускоряет процесс разработки программного продукта. Также в Unity3D реализован собственный механизм создания и настройки анимации, что по сравнению с OpenGL является большим достоинством. Отладку и изменение различных параметров трехмерной сцены можно осуществлять в режиме реального времени. Однако существует проблема взаимодействия с другими приложениями. Одним из способов взаимодействия с другими разрабатываемыми приложениями является использование сетевых компонентов Network Views. Это вынуждает разработчика подстраиваться под интерфейс Unity3D, что достаточно сложно и долго по времени. Именно этап настройки взаимодействия Unity3D с другими приложениями является самым трудоемким процессом. Unity3D является самостоятельным программным продуктом, что не позволяет внешним приложениям напрямую получить доступ к интерфейсу Unity. Этим обусловлено нарушение целостности разрабатываемого на его основе имитатора воздушной обстановки, что является большим недостатком. Из вышесказанного можно сделать вывод о том, что графический интерфейс Unity3D, за счет простого, удобного и многофункционального интерфейса позволяет сформировать высокореалистичное и адекватное входное воздействие для имитатора воздушной обстановки за более короткий срок, и не требует при этом от разработчика серьёзных знаний и умений в программировании. Но при этом накладываются ограничения в организации взаимодействия с внешними приложениями.

Проанализировав два мощных и многофункциональных инструмента для работы с двумерной и трехмерной графикой можно сделать вывод, что и OpenGL, и Unity3D позволяют в реальном масштабе времени сформировать адекватное входное воздействие в

виде цифрового изображения для оптико-локационной системы при различных условиях наблюдения. Используя графический интерфейс OpenGL, разработчику предоставляется большая свобода действия. Это значит, что разработчик может полностью использовать весь функционал трехмерной графики, и на его основе создавать полноценные приложения, в той или иной степени использующие трехмерную и двухмерную графику. С одной стороны, такой подход требует от программиста глубокого знания законов трёхмерной графики и математических моделей, с другой стороны — даёт свободу внедрения различных инноваций. Unity3D в свою очередь предоставляет возможность воспользоваться набором готовых реализаций, что, с одной стороны позволяет сформировать высоко детализированное и качественное изображение с меньшими усилиями и за меньшее время, но ограничивает разработчика рамками возможностей, реализованных в Unity. Так же трудностью в использовании Unity3D является сложность взаимодействия с другими разрабатываемыми приложениями, что служит причиной нарушения целостности рассматриваемого на его основе имитатора воздушной обстановки.

Список литература

1. Ричард С.-мл., Липчак, Бенджамин. OpenGL. Суперкнига, 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1040 с.: ил. – Парал. тит. англ.
2. Демьянович Ю.Н. Основы автоматического управления. Проектирование следящих систем Ч.1: Аналитическое конструирование силовых следящих систем: учеб. – метод. Пособие – Минск: ВА РБ, 2009.– 92 с.
3. Вентцель Е.С. Теория вероятностей: Учеб.для вузов. – 7-е изд. стер. – М.: Высш. шк., 2001. – 575 с.: ил.
4. Хокинг Дж. Unity в действии. Мультиплатформенная разработка на C# / Пер. с англ. И. Рузмайкиной. – СПб.: Питер, 2016. – 336с.: ил. – (Серия «для профессионалов»).
5. Торн А. Основы анимации в Unity / пер. с англ. Р. Рагимова. – М.: ДМК Пресс, 2016. – 176 с.: ил.
6. Unity Manual [Электронный ресурс] /Режим доступа: <https://docs.unity3d.com/Manual/>. – Дата доступа: 05.11.2016.
7. А. С. Солонар, С. В. Цуприк. Модель входного воздействия для оптико-локационной системы зенитного ракетного комплекса с подвижной оптической системой. Информационные технологии и системы 2016 (ИТС 2016): И74 материалы международной научной конференции, БГУИР, Минск, Беларусь, 26 октября 2016 г. с 236 – 237.
8. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах: Учебное пособие. – Новосибирск: Изд-во НГТУ, 2000. – 168 с.
9. Мирошников М.М. Теоретические основы оптико-электронных приборов. Учебное пособие для вузов: Л., «Машиностроение» (Ленингр. отд-ние), 1977. – 600 с. с ил.