

DESIGN AND CONSTRUCTION OF A VIRTUAL COMPUTER MODEL

STOYAN V. STOYANOV, STANIMIR K. ZHELEZOV

ABSTRACT: *The purpose of this work is to build a virtual model of a computer system for student education, based on analysis of different platforms, creating virtual reality and its related 3D graphics. An classification of three-dimensional models was made. Software platforms for three-dimensional modeling and creation of virtual scenes have been analyzed and selected. A virtual model has been built.*

KEYWORDS: *Virtual reality, 3D – modeling, virtual scenes.*

ПРОЕКТИРАНЕ И ИЗГРАЖДАНЕ НА ВИРТУАЛЕН МОДЕЛ НА КОМПЮТЪРНА СИСТЕМА*

СТОЯН В. СТОЯНОВ, СТАНИМИР К. ЖЕЛЕЗОВ

АБСТРАКТ *Целта на настоящата разработка е на базата на анализ на различни платформи, за създаване на виртуална реалност и свързаните с нея 3D графики, да се създаде виртуален модел на компютърна система за обучение на студенти. Направена е класификация на тримерните модели. Анализирани са и са избрани софтуерни платформи за тримерно моделиране и създаване на виртуални сцени. Разработен е виртуален модел.*

1 Въведение

Целта на настоящата разработка е на базата на анализ на различни платформи, за създаване на виртуална реалност и свързаните с нея 3D графики, да се създаде виртуален модел за обучение на студенти по дисциплините „Асемблиране на компютри“ и „Компютърни архитектури“. За тази цел е необходимо да се решат следните задачи:

- Да се направи анализ на софтуерните системи за 3D компютърна графика и виртуално моделиране;
- На базата на анализа на най-популярните инструменти за 3D моделиране и създаване на виртуална реалност да се избере подходящ инструментариум;
- Да се разработи приложение за нуждите на обучението.

2 Моделиране на 3D обекти

Всяко триизмерно изображение, създадено от компютър, изисква три основни компонента:

1. Описание на триизмерна сцена;
2. Един или повече източници на светлина
3. Описание на камерата или изглед към сцената.

Описанието на сцената обикновено се състои от един или повече модели или триизмерни структури. Обикновено един човек мисли за модел като самостоятелна част, например молив или дърво, а сцената като сглобяване на тези части в пълна триизмерна среда. Това отношение повлиява най-често срещаната процедура за изграждането на

* Настоящата статия е частично финансирана по проект № РД-08-159/09.02.2018 “Надеждност и защита на информация в социалните мрежи, графичните и 3D обекти в добавена и виртуална реалност” на ШУ

триизмерна сцена: изграждат се много модели и след това се сглобяват. Всеки модел има две описания[1]:

- математическо представяне на структурата на формата.
- изглед на формата при осветяване.

Структурното описание е основно геометрия. То ни казва къде е обектът в пространството и къде не е. Повечето компютърни програми използват вградената изчислителна система в днешните компютри. Този хардуер е с много висока, но ограничена прецизност. Някои методи за моделиране са много близки до този подход; те създават точки в пространството или го нарязват много фино и му слагат етикет празно или пълно. Други методи се опитват да позволят да се опише модела по-абстрактно, като например, че една чаша представлява цилиндър с дъно.

Другата част от всеки модел е описанието на повърхността. Това се свежда до описване физиката на взаимодействието на повърхността на модела със светлината. За щастие се оказва, че тази физика се ръководи от няколко описателни термини, които имат интуитивно значение за човека, като цвят, блясък и прозрачност[2].

Тъй като моделите често симулират форми от реалния свят, трябва да се определи колко подробен да бъде моделът [3,4].

За да бъдат анализирани и сравнени съществуващите програми за 3D моделиране е необходимо да се разгледат видовете модели, които се реализират от софтуерните системи.

Класификация на моделите

Повърхностни и гранични модели

Най-простият модел на повърхност е просто колекция от 3D точки. Те могат да бъдат организирани в прост списък или могат да бъдат по-добре структурирани като контури, части или секции. Повърхностите, представени от точки, изискват доста гъсто разпределение на точките за точно моделиране.

Импулсни повърхности / алгебрични уравнения

Тези повърхности са определени като решения за алгебрични формули. Един познат клас от такива повърхности са квадратичните обекти. По принцип повърхността е решение на уравнение като $F(x, y, z) = 0$.

Полигони

Полигоналните (полиедрични) модели са едно от най-често срещаните представяния в компютърната графика. Моделите се дефинират като мрежи от полигони, образуващи триизмерни полиедри. Всеки полигон (примитив) се състои от някои свързани върха, ръба и лицевата структура. Възможни са различни структури от данни[8]. Групи полигони се съхраняват в списъци, таблици или свързани структури. Полигоните са оразмерени, оформени и позиционирани така, че да покриват напълно необходимата повърхност на определена резолюция. Многоъгълните модели са относително лесни за дефиниране, манипулиране и показване. Те са най-разпространеният модел, обработен от хардуер и софтуер за графичен дизайн на работната станция. По принцип полигоните са най-добри при моделирането на обекти, които имат плоски повърхности.

Извити повърхности

Тъй като полигоните са добри при представянето им на повърхности, са изразходвани значителни усилия за определяне на математическите формули за действително извити повърхности. Всяка извита повърхност се нарича пластир; пластирите могат да бъдат съединени по техните гранични ръбове в по-сложни повърхности. Обикновено пластирите се дефинират от полиноми от нисък ред

(обикновено кубични), които дават на пластира лесно изчислителни математически свойства, като добре дефинирани повърхностни нормали и допирателни и изчислими условия на непрекъснатост между периферните съседни пластири. Съществуват многобройни формули на извити повърхности, включително: Безие, Хермайт, полиноми, рационални полиноми, кардинални сплайни, композитни спланове и др.

Обем и CSG модели

Първият обемен модел, който се разглежда, е моделът на воксела. Тук пространството е напълно запълнено с мозайка от кубчета или паралелепипеди, наречени „voxels“ (обемни елементи). Обикновено има плътност или друга цифрова стойност, свързани с всеки воксел. Съхраняването на мозайка с висока разделителна способност е скъпо в пространството, но не струва нищо в структурата на данните (просто голям триизмерен масив от стойности). Обикновено някои схеми за оптимизиране на съхранението са необходими за подробна работа (1K x 1K x 1K пространства)[5].

Осмични дървета

Осмичните дървета са една от структурите за данни, използвани за обемни модели, които се натрупва в дадено триизмерно пространство. Оригиналният обем, например куб, е разделен на 8 кубчета, ако не е празен. Рекурсивно, всеки под-куб е разделен, когато не е празен, докато не достигне някакъв елемент с минимален размер. Тъй като празните кубчета не са подразделени, ефективността на пространството за съхранение се увеличава. Основното използване на осмичните дървета изглежда е схема за индексване на ефективността на достъпа в голям триизмерен масив.

Конструктивна геометрия

Една от най-ефективните и мощни техники за моделиране е конструктивната геометрия. За разлика от схемите на воксел и осмичните дървета, няма изискване редовно да се натрупва цялото пространство. Освен това примитивните обекти не се ограничават до кубове; съществуват множество прости примитиви като куб, сфера, цилиндър, конус, половин пространство и т.н. Всеки примитив се трансформира или деформира и се позиционира в пространството. Комбинациите от примитиви или от предишни комбинирани обекти се създават от булевите операции. Следователно обектът съществува като дървовидна структура, която се оценява по време на рендиране или измерване[6].

Специализирани (единични примитивни) системи

Общата структура на метода на конструктивната геометрия с нейните множество примитивни обекти и скъп и бавен метод ray-tracing често се намалява, за да се спечели ефективност при конструкцията на модела, да се избегнат булеви комбинации, различни от обединението, и да се увеличи скоростта на показване. Идеята е да се ограничат примитивите до един тип, а след това да се проектират манипулации и характеризират алгоритми, за да се възползва от еднаквостта на представянето.

Потенциални функции

Интересно обобщение на модела на сферата е обмислянето на обема като потенциална функция с централна и полева функция, която намалява монотонно (чрез експоненциална или полиномна функция) от центъра навън. Няма "радиус" или размер на потенциалната функция; по-скоро, размерът или повърхността се определя чрез задаване на прагова стойност за полето. Това, което прави това по-интересно, е, че потенциалните функции действат като енергийни източници: прилежащите потенциални функции имат припокриващи се полета и получената стойност в дадена точка в пространството е всъщност сумата от активните полета в тази точка. По този начин съседните полета се смесват гладко, за разлика от "гънките", които се получават с фиксирани радиусни сфери.

Системи от частици

Обобщавайки сферите в различна посока, системите за частици намаляват сферата до нулев радиус. Обемът се характеризира с набор от (обикновено движещи се) частици, които индиректно определят пространството. Всяка частица има свой цвят, път, история и времетраене. Тяхното движение обикновено се контролира от вероятностни алгоритми. Системите от частици са използвани за моделиране на огън, газове, експлозии, фойерверки и тревисти полета[7].

Има няколко типа програми, които да помогнат създаването на триизмерни модели. Интерактивен софтуер – един клас на програма за моделиране интерактивен, когато обикновено дизайнерът седи пред терминал с клавиатура и мишка и манипулира форми на екрана.

- Интерактивните моделиращи софтуери обикновено съдържат разнообразни помощни средства за проектиране, които помагат да се постигне известна точност в модела: често срещан пример е да се "snap"(вмъкват) точки на модел в точки на друг модел или към невидима решетка. Моделите се изграждат чрез интерактивно избиране, модифициране и сглобяване на примеси на примитиви.
- Скриптов софтуер - друг клас на програма за моделиране разчита на входния скрипт за дефиниране на модела. Скрипта определя формите в модела една по една, като идентифицира всяка от тях с примитива и параметрите и, които определят този модел.
- Друг софтуер - Съществуват и други начини за създаване на модели, например чрез дигитализиране на съществуващи триизмерни структури или чрез анализ на снимки.

Тези програми обикновено се наричат моделиращи и съчетават в себе си различни типове моделиране от посочените в класификацията. Една идея, която е обща за повечето софтуери, е, че дизайнерът може да създаде копие на някаква основна форма (като блок или кръпка) и след това да я модифицира, за да създаде част от модела. Основните форми се наричат примитиви за тази програма. Прототипната форма е идеалната и случаите са практични реализации на този идеал [6].

3 Избор на софтуер и реализация на 3D модела

При избора на софтуер за изграждане на необходимия за виртуализацията на компютърната конфигурация 3D модел бяха анализирани най-разпространените софтуерни системи за 3D моделиране - Cinema 4D, Blender, 3ds Max, Maya и CAD/CAM [8].

Всяка от разгледаните системи притежава редица предимства като мощен анимационен пакет, поддръжка на широк формат за импортиране и експортиране, богат списък от инструменти за моделиране и др. Естествено те притежават и множество недостатъци - слаба система от частици, недостатъчна интуитивност, забавяне при високо детайлни модели, лошо организиран интерфейс и др. Някои от тези системи имат голям набор от инструменти и готови вградени обекти, но при тях се наблюдават недостатъци като висока цена, огромни изисквания към хардуера, стръмна крива на изучаване и др. От друга страна безплатните софтуери са бедни от към вградени инструменти и обекти, държат се доста нестабилно, забавят работата си при по-детайлни модели и т.н.

От направените анализи на софтуерните системи за 3D моделиране като най-подходяща се откроява Cinema 4D [9]. Някои от основните и предимства пред останалите са:

- Изчистен интерфейс: Персонализируемите плаващи и конфигурируеми панели за наблюдение позволяват пространството да бъде посветено на произведението, а не на интерфейса.

- Cinema 4D е известна със своята достъпност. Лесна за използване от артисти и дизайнери, както и от потребители без технически познания в областта.

- MoGraph е невероятен набор от инструменти в Cinema 4D, който ускорява процеса на създаване на анимации и графики за движение.

- Новоразработен кеш за излъчване (Irradiance Cache).

- Актуализиран инструмент Bevel.

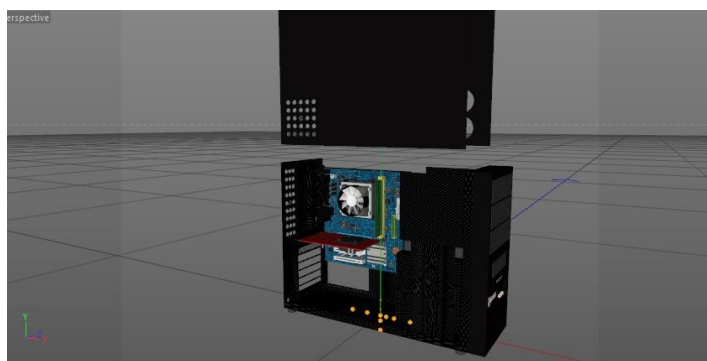
- Intel Embree във физически рендерер.

- Изключително стабилна.

- Cinema 4D е напълно заредена с богата библиотека от предварително зададени обекти.

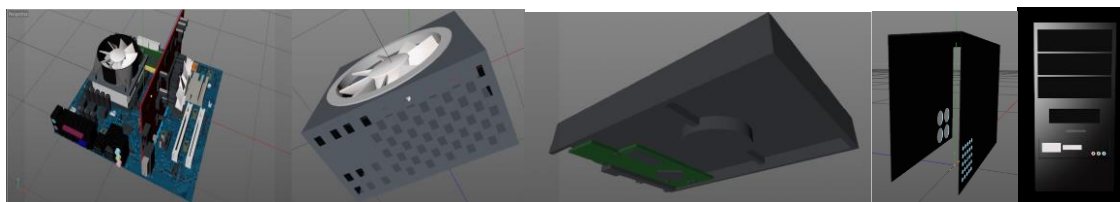
Cinema 4D е интуитивен, удобен както за професионални дизайнери така и за начинаещи. Изключително голям набор от функции и инструменти, с които времето за моделирането на обекта е значително по-кратко. Тя предоставя изчерпателна документация, в която е дадено пълно описание, предоставено от редица примери, както и видео и текстови уроци и сесии за обучение на живо. Също така предоставя „friendly“ интерфейс с което позволява лесно изграждане на тримерните модели. Не на последно място програмата предоставя възможност за лесно премоделиране на обектите с цел по-лесно изграждане на алтернативни тримерни обекти, което е една от целите на настоящата разработка от гледна точка на бъдещо развитие.

За реализацията на 3D модела на компютърната система (фиг. 1) се изграждат модели на основните елементи на системата – дънна платка, захранване, твърд диск и кутия (фиг. 2).

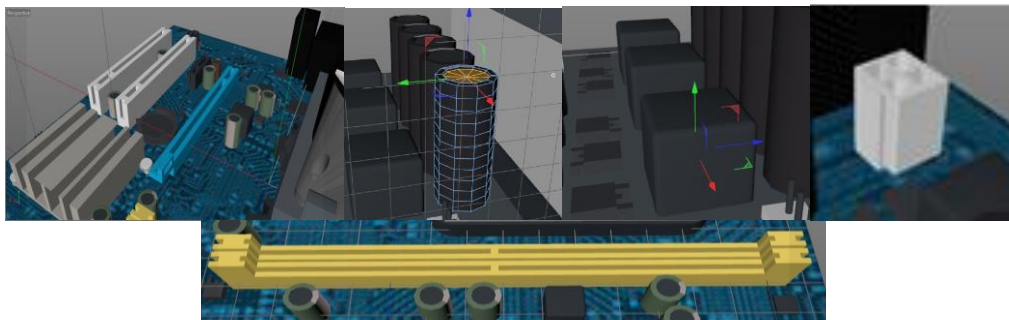


фиг. 1 Триизмерен модел на персонален компютър

За създаването на основните обекти се използват допълнителни подобекти (техни наследници), които са съставни части на обекта, и текстуриране на обекта. Дънната платка представлява съвкупност от различни видове слотове (фиг. 3) и разположените върху тези слотове компоненти – видеокарта, процесор, оперативна памет, охладители и др.(фиг. 4). А също така и различни дребни детайли – кондензатори и др.(фиг. 3)

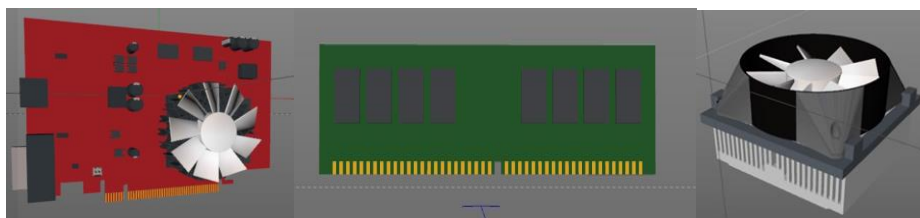


фиг. 2 Основни елементи на модела



фиг. 3 Основни елементи на модела на дънната платка

Всеки от тези компоненти може да се разглежда и като самостоятелен обект. Това позволява при моделиране на допълнителен набор от компоненти да се променя конфигурацията на компютърната система. По този начин може да се увеличи броя на конфигурациите, които да се предложат на обучаемите.



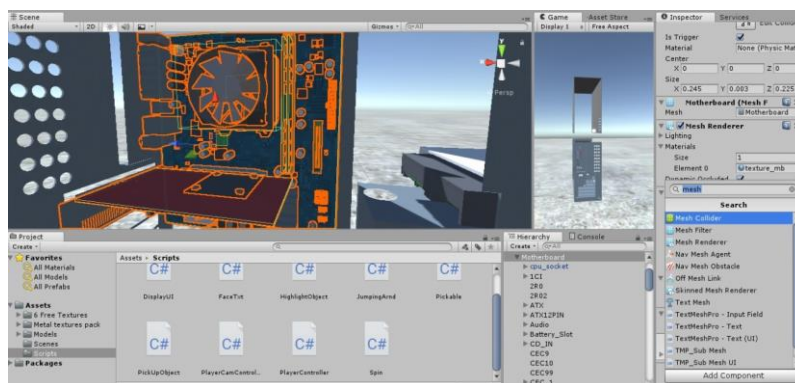
фиг. 4 Компоненти на компютърната система

4 Реализация на приложение за виртуална реалност

След успешното моделиране на обекта е ред на неговото импортиране във виртуалната реалност. Както и при избора на среда за моделиране е направен сравнителен анализ на различни софтуерни платформи за виртуализация. Сравнени са предимствата и недостатъците на най-разпространените продукти – Unity и Unreal Engine. За да бъде реализирана виртуалната сцена е избрана софтуерната платформа Unity [10]. Нейните предимства като отличен баланс на лекота на използване и мощност, вграден физически базиран рендеринг и висококачествени графични ефекти, активното развитие, редовно фиксирани бъгове и редовно пускани нови функции и т.н., определят нейния избор за реализиране на виртуалната сцена.

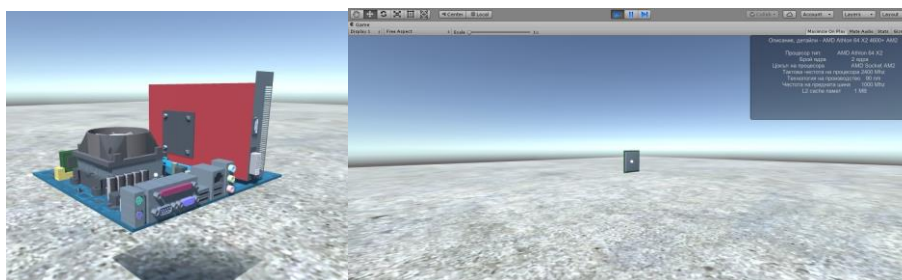
Крайната цел на приложението е асемблирането и деасемблирането на компютърна конфигурация във виртуална среда. За целта се експортира на модела от Cinema4D към

Unity и с помощта на вградените инструменти на Unity се импортира в програмата, като се запазва детайлността на тримерния модел. След успешното експортиране/импортиране отделните компоненти на модела да се „създадат“ като движещи се обекти за да може играча да ги вдига и оставя на различни позиции. На всички основни обекти на модела (капак, дънна платка, хард диск, захранващ блок, процесор, охладителя на процесора, РАМ паметта) се добавя по един нов компонент с име Mesh Collider (фиг. 5). Меш колизията се използва при сложни обекти с неточна форма, тя автоматично заема формата на обекта.



фиг. 5 Mesh collider върху компонентите на компютъра, тоест те вече могат да си взаимодействат с околната виртуална среда.

Модела и неговите подобекти се нуждаят от меш колизия за да може да бъдат вдигани, оставяни и т.н., в противен случай без колизията моделът би бил статичен. За да може отделните обекти на модела да се вдигат, оставят или въртят се създава C# скрипт. Чрез добавяне на функция в вече създадения скрипт се добавя информация към всеки обект, който е бил избран (фиг. 6).



фиг. 6 Дънната платка и процесора след изваждането и добавената информация

5 Заключение

Изучаването на учебни дисциплини с хардуерна насоченост, като „Компютърни архитектури“ и „Асемблиране“, предполага наличието на голям обем технически средства и ресурси. Тяхното представяне на студентите като хардуерни компоненти води до чисто практически затруднения от гледна точка на създаването на помещения и хранилища на този вид нагледни материали. За да се избегнат посочените неудобства могат да се използват възможностите на системите за изграждане на виртуална реалност.

Разработените виртуална сцена и свързания с нея тримерен модел могат да послужат като добра основа за последващото изграждане на цялостни виртуални системи за асемблиране и деасемблиране на компютри. С помощта на тези системи студентите биха могли, независимо от местоположението си, да получат необходимите им знания и умения за апаратната част на различни видове компютърни системи и техните параметри, характеристики и съвместимост. Предложеният тримерен модел лесно може да бъде модифициран с цел разширяването базата от компоненти за визуализиране и по този начин да се представи развитието на компютърната техника през годините с нагледни средства. Това е позволява настоящата разработка да се разглежда като първи етап от изграждането на цялостна система за виртуално обучение в областта на информационните технологии и хардуер.

ЛИТЕРАТУРА:

- [1] Haresh Khemani, Applications of CAD Software: What is Solid Modeling? www.brighthubengineering.com, 2008г.
- [2] Wisslar, Virginia. Illuminated pixels: the why, what, and how of digital lighting. Cengage Learning, 2013.
- [3] Gahan, Andrew. 3D Automotive Modeling: An Insider's Guide to 3D Car Modeling and Design for Games and Film. Focal Press, 2012.
- [4] Vaughan, William. Digital modeling. New Riders, 2011.
- [5] Straßer, Wolfgang, and Hans-Peter Seidel, eds. Theory and practice of geometric modeling. Springer Science & Business Media, 2012. Masjukov, A.V., Masjukov, V.V., Multiscale modification of Shepard's method for multivariate interpolation of scattered data. In: Mathematical Modeling and Analysis. Proceeding of the 10th International Conference MMA2005&CMAM2, 467-472.
- [6] KANA, Martiale G. ZEBAZE. "Introduction to solid modeling.", 1988.
- [7] Savelonas, Michalis A., Ioannis Pratikakis, and Konstantinos Sfikas. "An overview of partial 3D object retrieval methodologies." Multimedia Tools and Applications 74.24 (2015): 11783-11808.
- [8] Earnshaw, Rae A., ed. Virtual reality systems. Academic press, 2014.
- [9] Kaminar, Aaron. Instant Cinema 4D Starter. Packt Publishing Ltd, 2013.
- [10] Roedavan, Rickman. "Unity Tutorial Game Engine." Bandung: Informatika 2014.

Стоян Василев Стоянов

Шуменски Университет „Еп. Константин Преславски“
E-mail: 404i90@gmail.com

Станимир Кунчев Железов

Шуменски Университет „Еп. Константин Преславски“
E-mail: s.zhelezov@shu.bg