

# ОПЫТ СОЗДАНИЯ ГРАФИЧЕСКИХ ПРИМИТИВОВ ДЛЯ ОФОРМЛЕНИЯ ТЕЛЕВИЗИОННОГО КОНТЕНТА «ТРК «РАДОНЕЖЬЕ» В КОМПЬЮТЕРНОЙ СРЕДЕ ТРЕХМЕРНОГО МОДЕЛИРОВАНИЯ BLENDER® С ПРИМЕНЕНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON™

**PRACTICE OF DRAWING PRIMITIVES  
FOR «RADONEJE» BROADCASTING  
COMPANY CONTENT DESIGN IN 3D —  
MODELING COMPUTER ENVIRONMENT  
BLENDER® BY USING PYTHON™  
PROGRAMMING LANGUAGE**

**V. Taran  
V. Pakshina**

*Summary.* Technical issues of computer design based on drawing primitives (meshes) for making up the content of TV broadcasting streaming are discussed in the article. Being illustrated by logo of regional TV and radio company «Radoneje» the experience of 2D and 3D-geometric design (subsequently taking into account their rotation) by using method of redistribution of light streams and hiding when extrusion of clipping outline of the structure in Blender® 3D-modeling environment with partial using of Python™ programming language resources is analyzed. The method above proved to be very helpful for accurate reproduction of rendered geometrical construction including light transitions and graduated fills. The method is universal allowing accurate reproduction of visual content independently from computer programming. General problem areas of computer 3D-modelling associated with 3D-objects modeling are pointed out. Sequence of operations when generating the effects related to drawing, extrusion and potential hiding of some layers by an opportunity of their visualization has been systemized. Based on symbolic algorithms and schematic illustrations the main capabilities of software packed 3D-visualization Blender® were revealed and certain conditions of its technical exploiting when modeling the objects and scenes of medium and high complexity were determined.

*Keywords:* computer 3D-modeling, visualization, automation of the modeled operations, methods of computer modeling, drawing primitives, computer geometry, computer graphic, design, sculpt, digital composition, extrusion, pseudo extrusion, light streams, Blender®, Python™ programming language.

**Таран Василий Васильевич**

К.культурологии, АНО ВО «Московский  
международный университет»; докторант, ФГБУН  
«Всероссийский институт научной и технической  
информации РАН»  
allscience@lenta.ru

**Пакшина Вероника Алексеевна**

АНО ВО «Московский международный университет»;  
МАУ «ТРК «Радонежье»  
pakshina-veronika@mail.ru

*Аннотация.* В статье рассматриваются технические вопросы компьютерного проектирования на основе графических примитивов (мешей) для оформления контента потокового телевизионного вещания. На примере логотипа регионального вещателя телерадиокомпании «Радонежье» анализируется опыт создания двухмерных и трёхмерных геометрических конструкций (в последствии учитывающих их ротацию) методом перераспределения световых потоков и скрытия слоёв при экструзии оброчных контуров конструкции в системе трёхмерного моделирования Blender® с применением некоторого арсенала языка программирования Python™. Обосновывается, что данный метод может быть очень полезен при точном воспроизведении отрисованной геометрической конструкции, включая световые переходы, цветовые переходы и градиентные заливки. Метод является универсальным и позволяет не зависимо от компьютерной программы воспроизводить визуальный контент с высокой степенью точности. Выделяются общие проблемные области компьютерного 3D-моделирования, связанного с моделированием объёмных геометрических объектов. Систематически изложены последовательности действий при создании эффектов, связанных с прорисовкой, экструзией и потенциальным скрытием некоторых слоёв при возможности их визуализации в кадре. На основе формульных алгоритмов действий и схематических иллюстраций показаны основные возможности программного пакета трёхмерной визуализации Blender® и определены некоторые условия его технической эксплуатации при моделировании объектов и сцен средней и высокой сложности.

*Ключевые слова:* трёхмерное компьютерное моделирование, визуализация, автоматизация моделируемых действий, методы компьютерного моделирования, графические примитивы, компьютерная геометрия, компьютерная графика, дизайн, скульптинг, цифровая композиция, экструзия, псевдоэкструзия, световые потоки, Blender®, язык программирования Python™.

**В** последнее время мы наблюдаем динамический рост различной видеопродукции, основным каналом реализации которой служит традиционное телевидение и Интернет. Так сложилось, что современная практика продвижения видеопродуктов различного назначения (рекламные ролики, отбивки, межпрограммные заставки, инфографика, короткометражные сериалы, телевизионные программы и т.д.) напрямую зависит от их стиля и наполнения. За эти две категории отвечает дизайн. С активным внедрением информационно-коммуникационных технологий в сферу дизайна появилось огромное количество возможностей по созданию дизайна графико-, видео- и анимационной продукции, что в свою очередь несомненно повлияло на всю индустрию дизайна. Благодаря достижениям компьютерных наук (в части разработки и усовершенствования алгоритмов обработки графической и видеоинформации) открылись новые методы и подходы к разработке различных оригинальных решений во всех сегментах дизайна.

Новые методы разработки дизайн-решений теперь прежде всего зависят от уровня развития информационно-коммуникационных технологий, через которые активно стали продвигаться программные продукты специального назначения. В этом смысле современный рынок программной продукции это фактически цифровой рынок опирающийся на главное достижение компьютерных наук и ИКТ 20-го века — глобальную сеть Интернет. Интернет сегодня наиболее развитая форма информационно-коммуникационных технологий, включающих в себя широкий спектр телекоммуникаций и позволяющий устанавливать интегративные мультиканальные связи между людьми (компьютерными пользователями), а также между машинами (искусственный интеллект). Ярким примером такой интегративности является интернет-вещей который посредством единого аппаратно-программного интерфейса фактически объединяет различные электронно-бытовые устройства (электрический чайник, тостер, холодильник, плеер, часы) в единую локальную сеть, шлюз которой подчиняется различным интернет-протоколам (TCP/IP)<sup>1</sup>.

Разумеется, дизайн этих вещей, включая аппаратную начинку и программный интерфейс проектируется в специализированных компьютерных программах, это различные САПР-системы и программные пакеты по трёхмерному моделированию. Одним из таких пакетов является Blender®. Blender® это открытое свободно-распространяемое программное обеспечение, ре-

гулируемое положениями (GNUGPL<sup>2</sup>) и объединяющее в себе целый комплекс инструментов, направленных на эффективное проектирование и моделирование геометрических объектов.



Функционал Blender® впечатляет, а его производительность местами поражает воображение, ведь благодаря чётко проработанной структуре и слаженности работы алгоритмов эффективность обработки данных становится довольно высокой. Blender® отлично подходит для решения различных задач в области компьютерного моделирования, приведем небольшой список наиболее значимых областей, где может применяться программа:

- ◆ Разработка геометрических объектов для промышленной архитектуры.
- ◆ Создание прототипов для городской и сельской архитектуры.
- ◆ Визуализация и анимация процессов и явлений в науке и культуре.
- ◆ Разработка графического дизайна компьютерных и видео игр.
- ◆ Создание интерфейсно-ориентированных компьютерных систем.
- ◆ Проектирование геометрических конструкций и примитивов для оформления телевизионного контента и видеопроизводства.

Вышеперечисленные пункты списка показывают, насколько широко может быть применен инструментарий данной программы. Последний пункт списка как раз хорошо иллюстрирует поставленную в нашей статье цель — разработки графических примитивов для оформления телевизионного контента. Объектом нашей цели выступает телеканал «Радонежье»<sup>3</sup>, и соответствен-

<sup>1</sup> TCP/IP (Transmission Control Protocol/Internet Protocol) — стек протоколов, определяющих иерархию разделения (дискретные пакеты) и перераспределения данных, с учётом описания формата пакета данных, контролирующей доставку и сохранность передаваемых данных. За описание формата пакетных данных (IP-пакеты) и за присвоение адресов пользователей компьютерной сети отвечает протокол IP.

<sup>2</sup> GNU/GPL («GNU's — Not Unix», GENERAL PUBLIC LICENSE) — Version 2, June 1991. Лицензия доступна по адресу: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html> (дата обращения к источнику: 27 мая 2020 г.). Открытый проект по разработке и распространению свободного программного обеспечения, получивший своё название от свободнораспространяемой операционной системы GNU, девизом которой стал слоган «GNU — не Unix». Официальный сайт ОС: <https://www.gnu.org/>

<sup>3</sup> МАУ «ТРК «Радонежье» Муниципальное автономное учреждение Сергиево-Посадского городского округа «Телерадиокомпания «Радонежье», в некоторых источниках (Муниципальное автономное учреждение Сергиево-Посадского муниципального района «Телерадиокомпания «Радонежье»). «Радонежье» основано в 1999 году, телерадиокомпания выступает в роли крупнейшего вещателя города Сергиев Посад и ближайших окрестностей. Телерадиокомпания является обладателем четырёх наград ТЭФИ. Официальный сайт телерадиокомпании: <https://tvr24.tv/>.

```

PYTHON INTERACTIVE CONSOLE 3.5.3 (default, May 20 2017, 19:10:40) [MSC v.1800 64 bit (AMD64)]
Command History:      Up/Down Arrow
Cursor:               Left/Right Home/End
Remove:               Backspace/Delete
Execute:              Enter
Autocomplete:        Ctrl-Space
Zoom:                 Ctrl +/-, Ctrl-Wheel
Builtin Modules:     bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, bpy.utils, bgl, blf, mathutils
Convenience Imports: from mathutils import *; from math import *
Convenience Variables: C = bpy.context, D = bpy.data

>>> |

```

Рис. 1. Встроенная интерактивная среда управления языка программирования Python™. Позволяет выполнять различные инженерные команды, облегчающие управление графическими проектами.

но для успешной реализации поставленной в научной статье цели следует конкретизировать несколько задач:

1. Создать двумерную геометрическую форму логотипа телеканала «Радонежье».
2. Разработать 3D-проекцию созданной геометрической формы, с учётом её потенциального импорта<sup>1</sup> в программы компьютерной визуализации (титровальная графика) и программы трёхмерного моделирования (САПР и редакторы 3D-моделирования).
3. Визуализировать элементы идентификации объектов в кадре.

Итак, приступая к решению поставленных задач, определимся с параметрами проекта. Проект имеет разрешение  $1920 \times 1080^2$  px. Для мониторов, имеющих соотношение 16:9. В качестве основы для распределения и компоновки двухмерных и трёхмерных геометрических конструкций в кадре, можно использовать схемы, изложенные на рисунках 18, 19. Хочется сразу отметить ориентацию интерфейса Blender® на широкоформатные видео-анимационные двухмерные и трёхмерные проекты<sup>3</sup>, впоследствии упрощающие процедуры рендеринга<sup>4</sup>. Логотип канала «Радонежье» представляет собой ге-

<sup>1</sup> Учёт потенциального экспорта подразумевает программно-независимый экспорт графических трёхмерных моделей, с целью их точной прорисовки и отображения в не зависимости от импортируемого формата.

<sup>2</sup> При создании проекта с высоким разрешением для стабильного отображения цветов и градиентных переходов рекомендуется использовать 24-х битный монитор.

<sup>3</sup> Интерфейс программы Blender® хорошо заточен под выполнение простых и сложных графических проектов, здесь имеется настройка камер и возможность вывода графических данных в следующих форматах: 1080 p. (DVCPRO HD), 720 p. (DVCPRO HD), 1080 p. (HDTV), 720 p. (HDTV), 1080 p. (HDV), 1080 p. (HDVNTSC), 1080 p. (HDVPAL), 16:9 (TV NTSC), 4:3 (TV NTSC), 16:9 (TV PAL), 4:3 (TV PAL). Также присутствует возможность ввода абстрактных параметров проекта.

<sup>4</sup> Рендеринг — процесс финальной отрисовки двухмерных и трёхмерных графических конструкций, подразумевающий создание целостной композиции, включая тона, цветовые переходы, тени и т.д.

ометрическую форму в виде неправильного выпуклого пятиугольника, располагаемого в левом углу кадра.

Схема представления графических элементов базируется на пятиугольной геометрической конструкции и дополняет её. Как правило, это горизонтальная линия, примыкающая к неправильному пятиугольнику. Канал имеет собственный фирменный стиль, выражающийся в оранжевом оттенке, который местами имеет градиентные переходы в серо-белые тона. Программа Blender® устроена таким образом, чтобы визуально-ориентированная часть интерфейса необходимая для качественного 2D и 3D-скульптинга, пространственного видео-анимационного 2D и 3D-композитинга сочеталась с функционалом современного объектно-ориентированного программирования. Это обстоятельство существенно расширяет границы 2D и 3D-проектирования, а также имитационного дизайн-моделирования за счёт повышения точности редактирования геометрических объектов и упрощает рутинные процедуры при проектировании сложных объектов:

- ◆ Разбивка геометрической фигуры с целью получения нового объекта.
- ◆ Автоматизация процессов анимации (уточнение и групповое перераспределение траектории движения слоёв и т.д.).
- ◆ Проведение компьютерного синтеза цветов и манипуляция производными оттенками при полигональном моделировании.
- ◆ Автоматическое создание текстур и сложных слоёв методом генерации числовых распределений.

За возможность проведения вышеперечисленных технических процедур и в целом за автоматизацию различных действий отвечает специальная инженерная консоль управления Python™ Interactive Console<sup>5</sup>.

<sup>5</sup> Python Interactive Console — встроенная интерактивная среда управления прикладными сценариями Blender, написанными на языке

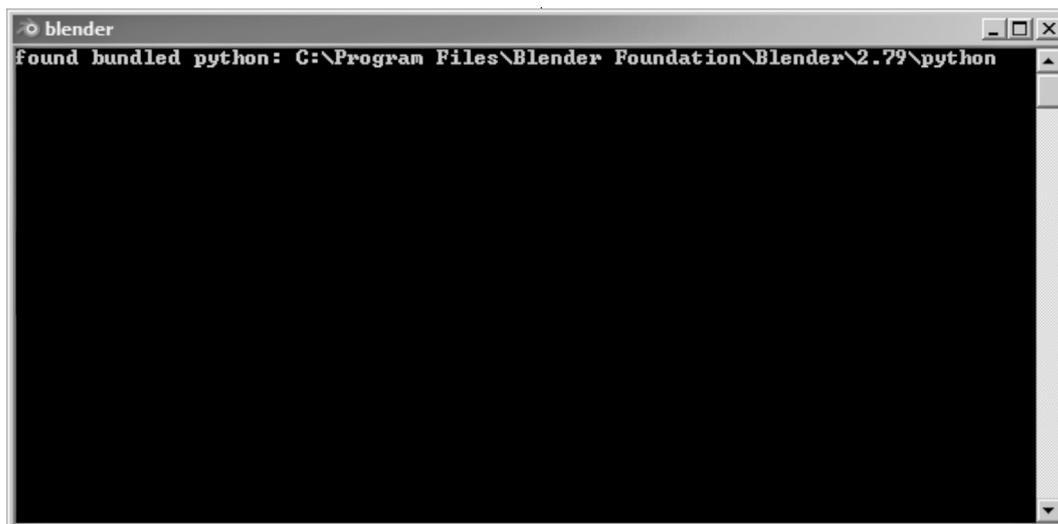


Рис. 2. Командная строка

```

453 if self.cir_top is True:
454     cir_top = create_torus("Column_cir_top", radio_top, self.cir_top_r, self.cir_top_z)
455     bpy.ops.object.select_all(False)
456     cir_top.select = True
457     bpy.context.scene.objects.active = cir_top
458     set_modifier_subsurf(cir_top)
459     set_smooth(cir_top)
460     cir_top.parent = mycolumn
461     cir_top.location.x = 0.0
462     cir_top.location.y = 0.0
463     cir_top.location.z = height - self.cir_top_z / 2

```

Рис. 3. Встроенный текстовый редактор ввода сценариев и макрокоманд, имеет функции подсветки синтаксиса

Опции консоли позволяют управлять техническими процессами внутри программы, ориентируясь на прорисовку деталей объекта и идентифицировать сложные элементы как символично-числовые значения. Blender® имеет также поддержку внешней программной среды Python™<sup>1</sup>, что значительно упрощает внешнюю регистрацию сообщений, для последующей корректировки.

Внутренняя объектно-ориентированная структура 3D-редактора Blender® имеет встроенный текстовый ре-

дактор для ввода программных сценариев, который как и окно моделирования сочетается с внутренним регистратором событий<sup>2</sup>.

Этот аспект позволяет расширить творческие горизонты 3D-проектировщика, за счёт гибкости настройки программного интерфейса и увеличить производительность выполняемых программой действий в не зависимости, на какой платформе она используется (Linux/Windows).

<sup>1</sup> Программирование Python™. Использование консоли позволяет упростить процедуры, связанные с выполнением рутинных действий при создании двумерных и трёхмерных геометрических конструкций, за счёт автоматизации заданных оператором программы действий.

<sup>1</sup> Прикладное использование языка Python™ — регулируется (Python Software Foundation, PSF), 2-й версией лицензионного соглашения. Подробнее о лицензиях и юридически правильном использовании языка программирования Python™, можно узнать по адресу: <https://docs.python.org/3/license.html#psf-license-agreement-for-python-release> (дата обращения к источнику: 27 мая 2020 г.).

<sup>2</sup> Внутренний регистратор событий иерархически отображает вносимые пользователем изменения в область моделирования и в другие режимы перехода (Scripting, Compositing and Animation), а также в составные части программы (VideoEditing, GameLogic), увеличивая общий коэффициент полезного действия интерфейса программы и развивая у пользователя навыки объектно-ориентированного программирования. Таким образом, любое действие в рабочей области программы (в режиме скриптинга) учитывается регистратором и отображается описаниями на языке программирования Python™ в журнале событий, что даёт возможность точно настраивать параметры проекта и автоматизировать некоторые рутинные действия, применяемые при дизайн-проектировании.

```

bpy.ops.paint.vertex_paint_toggle()
bpy.context.scene.tool_settings.unified_paint_settings.use_unified_color = True
bpy.context.scene.frame_end = 251
bpy.context.space_data.context = 'OBJECT'
bpy.context.space_data.context = 'CONSTRAINT'
bpy.data.brushes["Draw"].vertex_tool = 'SUB'
bpy.context.space_data.pivot_point = 'CURSOR'
bpy.ops.paint.vertex_paint_toggle()
bpy.ops.sculpt.sculptmode_toggle()
bpy.context.area.type = 'TIMELINE'

```

Рис. 4. Окно регистратора событий ведущего журнал изменений в режиме скриптинга.

Эти опции будут полезны и для нас при создании различных примитивов<sup>1</sup> для титровально-презентационной графики канала. Процесс создания 2D и 3D-графики в таком случае будет состоять из трёх ключевых звеньев:

Звено *первое* — скульптинг. Звено подразумевает непосредственно разработку двумерных и трёхмерных геометрических моделей, состоящих из замкнутых и незамкнутых линий, работу с мешами и т.д. На этом уровне будет произведена зарисовка необходимых графических объектов с применением интерфейсно-ориентированных функций (виртуальные кисти, маркеры, ручка) с уточнением углов и привязок к линиям объекта посредством списочных функций через интерактивную консоль языка Python™.

Звено *второе* — зарисовка текстур. Звено подразумевает зарисовку необходимых текстур для заливки, созданных в процессе скульптинга двумерных и трёхмерных геометрических моделей. Язык Python™<sup>2</sup> здесь может пригодиться для повышения точности опорных точек разметки текстуры, либо для автоматической её генерации с учётом заданных числовых координат. Использование интерфейса прикладного программирования (Blender® — Python™ API) позволит решать некоторые задачи более гибко и быстро. Ещё один немаловажный аспект, который позволяет производить манипуляции над отображаемыми сторонами геометрического объекта, это разработка световых текстур, (т.е. прорисовка необходимых граней будущей конструкции световыми потоками).

Звено *третье* — цифровая композиция. Звено характеризует процесс сопоставления графических эле-

ментов, произведённых на предшествующих этапах с целью оставления единой композиции. Упор делается на симметричность конструкций, их сочетание между собой и образования абстракций для более эффективных и привлекающих внимание иллюстраций. Помощь языка Python™ здесь может пригодиться в процессе сбора статистики создаваемых геометрических конструкций.



Начиная с первого звена (скульптинга), определимся с параметрами программы. Blender® устойчиво поддерживает практически все функции по скульптингу и имеет специальный интерфейсный режим (sculpt mode — настройка скульптинга). Переключив интерфейс программы в данный режим, мы получаем серьёзный арсенал для скульптинга и прорисовки различных линий свода.

В программе Blender®<sup>3</sup> имеется встроенная микробблиотека, представляющая собой меш-набор. В меш-набор традиционно входят стандартные геометрические конструкции — каркасы (окружность, цилиндр, куб и т.п.), на основе которых может быть выполнен скульптинг. Как правило (в прикладной компьютерной геометрии), как раз из простых объектов могут быть созданы сложные объекты. За трансформацию объектов в режиме скульптинга отвечает несколько разделов, в каждом из которых присутствуют контрольные опции. Это разделы: brush, texture, stroke, curve, symmetry/lock.

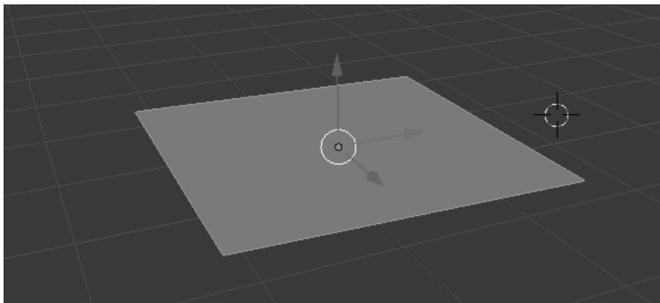
В первую очередь, нам необходим раздел brush. Раздел brush имеет встроенные контрольные опции: radius,

<sup>1</sup> Примитивы (в 3D-моделировании) это стандартные фигуры (блоки), являющиеся составными частями сложной 3D-модели. К примитивам также могут быть отнесены простые закрытые формы плоскости, кубы, пирамиды. Простые искривлённые формы сферы, цилиндры, конусы.

<sup>2</sup> Для более уверенной объектно-ориентированной работы и эффективной отладки скриптов необходимо установить соответствующую версию языка Python . Её можно загрузить с официального сайта: <https://www.python.org/>.

<sup>3</sup> При манипуляции с геометрическими конструкциями автораму использовалась система компьютерного 3D-моделирования Blender , версия — 2.79.

strength, autosmooth, pinch. Radius отвечает за размер кисти (измеряется в пикселях). С его помощью мы и будем корректировать созданную нами плоскость. Strength отвечает за силу нажима кисти. Опция важна при равномерном распределении обтравочных линий (контурах) на различных геометрических конструкциях. Autosmooth применяется для создания эффекта мягкости, важна при переходах и сглаживании линий и углов конструкции. Pinch устанавливает уровень сжатия. Из меш-набора мы выбираем плоскость — простой двухмерный примитив для образования неправильного выпуклого пятиугольника.



Каждое действие при желании может устанавливать программные связи с Python™ Interactive Console и адресно фиксировать командные действия кодовыми строчками сопровождения. Например, при организации мешей строчка обращения к программе будет выглядеть так

```
bpy.ops.mesh.primitive_plane_add()
```

Если необходимо подкорректировать координаты плоскости (plane) — например, при её перемещении в пространстве то результирующие значения могут выглядеть так

```
bpy.ops.transform.translate (3.74875, 0.932508, -6.13008) value
                             при перемещении
                             (1.56413, 4.57289, 0.836503) value
```

Соответственно при выборе типа меша для использования в проекте сквозная привязка будет иметь следующий вид:

```
bpy.ops.mesh.primitive_+_add()
H=plane_cube_circle_uv_sphere_ico_sphere_cylinder_cone_torus_grid_monkey_
```

Нижние индексы означают выбор потенциально используемого меша. Программа содержит восемь типов мешей:

- |          |              |               |                |
|----------|--------------|---------------|----------------|
| 1. Plane | 3. Circle    | 5. Ico Sphere | 7. Cone Torus  |
| 2. Cube  | 4. Uv Sphere | 6. Cylinder   | 8. Grid Monkey |

Каждый из них может быть использован 3D-проектировщиком по его усмотрению. Похожая история наблюдается и при выборе цифровых инструментов скульптинга. Каждая кисть отличается разновидностью и методом прорисовки, изменения объекта, адресная цепочка на языке программирования Python™, здесь будет следующей

Выбор, довольно внушительный, состоящий из двадцати настраиваемых образцов кистей, что вполне достаточно для реализации проектов повышенной сложности.

- |                |            |            |              |            |
|----------------|------------|------------|--------------|------------|
| 1. Blob        | 5. Draw    | 9. Inflate | 13. Pinch    | 17. Smooth |
| 2. Clay        | 6. Fill    | 10. Layer  | 14. Rotate   | 18. Shake  |
| 3. Clay Strips | 7. Flatten | 11. Mask   | 15. Scrape   | 19. Hook   |
| 4. Crease      | 8. Grab    | 12. Nudge  | 16. Simplify | 20. Thumb  |

Строчка начинается с обращения к модулю bpy (bpy.ops). Модуль bpy, как правило, содержит набор операторов для управления геометрическими объектами во внутренней системе 3D-моделирования. В рамках нашего исследования мы рассмотрим только основные модули (субмодули), затрагивающие процессы управления и манипулирования геометрией объектов. Поэтому на вводном этапе новые сцены можно создавать и упорядочивать через привязку bpy.data.scenes.new() [1]. Модуль обеспечивает объектно-ориентированное управление, процессами создания новых геометрических проекций, совмещая интерфейсно-ориентированное управление (перцепционный подход) с программным управлением (программный подход, с возможностью визуализации базовых фрагментов кода, необходимых для интерпретации дальнейших действий). Модуль является системообразующим звеном в программных связках языка Python™ и открывает дополнительные возможности по манипулированию сложными трёхмерными объектами, включая текстуру, контуры и рельеф [2].

Модуль bpy также оказывает влияние и на саму интерфейсную систему программы. Он может изменять традиционную схему представления объектов, регулировать и настраивать под предпочтения пользователя разделы просмотра и изменять параметры вывода графического содержимого из программы. В роли основных манипуляторов объектами выступают два класса (bpy.ops.object/bpy.ops.mesh). Класс object отвечает за основные функции по одновременному управлению несколькими wybranными объектами и позволяет производить селекторное редактирование трёхмерных изображений. Класс mesh включает функции для одновременного управления вершинами, рёбрами и гранями объектов, преимущественно в режиме Edit Mode. Модуль bpy.context используется для доступа к объектам и областям

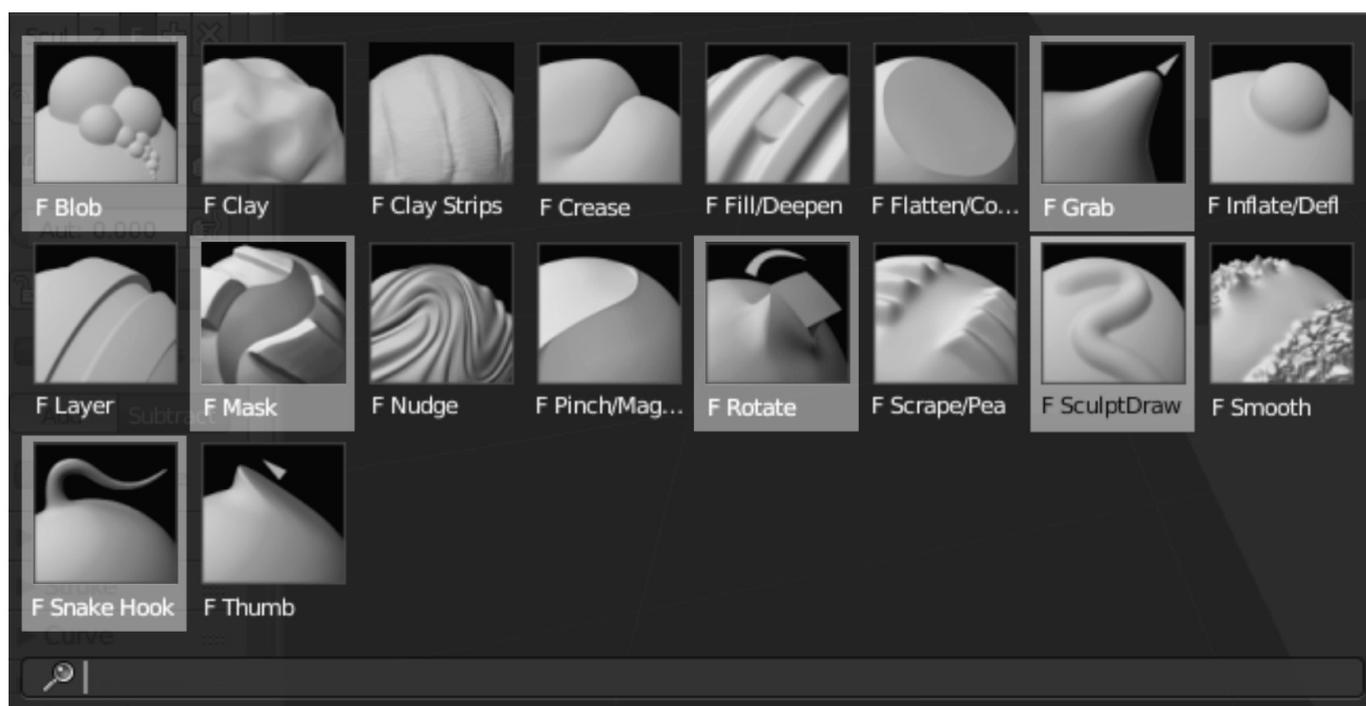


Рис. 5 Меню выбора кистей среды трёхмерного моделирования Blender®, опционально выделены кисти, которые применялись в проекте

программы Blender® по различным статусным критериям.

Главной функцией данного модуля является предоставление разработчикам Python™ средств доступа к текущим данным, с которыми работает пользователь программы. К примеру, если мы создаём и настраиваем кнопку, которая потенциально меняет порядок всех выбранных объектов, то это даёт возможность пользователю отбирать объекты на своё усмотрение, а затем переставить эти объекты в `bpy.context.select_objects`. Небольшой пример использования модуля (внутри архитектуры Blender® — субмодуля) для установки линзы

```
bpy.context.space_data.lens = 34.9 ...|34.5|...
```

При манипуляции с изображениями и сценами зачастую приходится использовать `bpy.context.scene` особенно, когда необходимо создать надстройки в силу обязательной загрузки определённых объектов в Blender®. Можно также использовать `bpy.context` для доступа к объектам, которые находятся в активном состоянии при переключении между режимами Object Mode — Edit Mode. В режиме проектирования `bpy.context` может также использоваться для приёма данных от (опций) карандаша. Модуль `bpy.data` служит проводником для доступа к внутренним данным программы Blender®.

Класс `bpy.data.objects` содержит все необходимые данные для определения формы и положения объекта. Если предыдущий субмодуль `bpy.context` хорошо подходит для выделения группы объектов, то имеется в виду, что классы `bpy.context` будут генерировать ссылки на блоки данных `bpy.data`-класса. Модуль `bpy.app` (`bpy.app.handlers`) субмодуль `handlers` содержит специальные функции для запуска полного цикла пользовательских функций в ответ на процессы в программе Blender®. Наиболее активно используется дескриптор смены кадров, который выполняет predetermined оператором функцию каждый раз, когда обновляется экран просмотра трёхмерных изображений (после смены кадра). Здесь стоит упомянуть, что язык программирования Python™ располагает большим арсеналом прикладных средств, направленных на визуализацию различных мультимедийных данных. Код языка Python™ может оптимизироваться в соответствии с поставленными задачами в рамках компьютерной геометрии и топологии геометрических объектов [3].

В режиме скульптинга опционально выбираем кисти, пригодные для нашего проекта<sup>1</sup>

<sup>1</sup> Большинство действий в интерфейсно-ориентированном режиме можно выполнять вручную, арсенал традиционных средств моделирования позволяет выполнять даже очень сложные графические проекты, однако когда у проекта присутствует определённая специфика (настройка эффектов,

Это (F) Blob, (F) Grab, F Mask, (F) Rotate, (F) Sculpt Draw, (F) Snake Hook. Каждая из кистей имеет стандартные уровни подстройки (Radius of the brush) — устанавливает радиус кисти-щётки для определения масштаба сдвига, в выражении Python™ имеет следующий вид:

```
UnifiedPaintSettings.size
bpy.data.scenes["Scene"]. tool_settings.unified_paint_settings.size
```

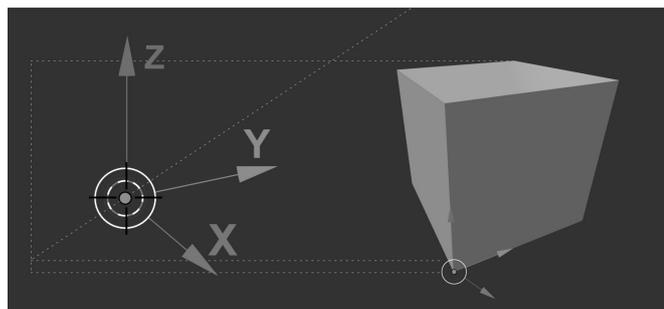
(How powerful the effect of the brush is when applied) — определяет насколько силен эффект при использовании кисти. Фактически уровень нажима кисти отвечает за степень прорисовки шага при пропорциональном изменении геометрической проекции. На языке Python™ выражается следующими строчками:

```
Brush.strength
bpy.data.brushes [«Blob»].strength
```

(Amount of smoothing to automatically apply to each stroke) — масштаб сглаживания автоматически применённый к каждому нажатию. Автоматически корректирует размер сглаживания, применяемый при нажатии кисти. На языке Python™ определяется:

```
Brush.auto_smooth_factor
bpy.data.brushes ["Blob"]. auto_smooth_factor
```

Для формирования проекции предусмотрен режим (Sculpt Plane), который позволяет визуально представлять проекцию по координатам.



Z-Plane — проектирование по оси «Z». Y-Plane — проектирование по оси «Y». X-Plane — проектирование по оси «X». View Plane — показывает конечный результат проекции. Area Plane — область проекции.

(How much the crease brush pinches) — регулирует насколько сильно растут изломы кисти. Устанавливает

определение количества слоёв, необходимых для повторения оригинального эффекта, не предусмотренного программой) становится очевидным, что гибкая настройка инструментария это единственный выход, особенно когда она опирается на высокоуровневое объектно-ориентированное программирование.

уровень излома кисти при нажатии на объект. Данный модуль также применим при уменьшении поверхности граф, лежащих в одной плоскости. Манипуляции по использованию компланарных вершин для уменьшения поверхности граф хорошо описаны в книге Криса Конлана (Chris Conlan), которая посвящена Python™ API в Blender® [4]. Языком Python™ предусмотрена следующая корректировка:

```
Brush.crease_pinch_factor
bpy.data.brushes ["Blob"]. crease_pinch_factor
```

(Brush only affects vertexes that face the viewer) — кисть воздействует на вершины только при прямом визуальном контакте. Опция полезна для масштабирования вершин на экранах разного типа (16:9 и 4:3). Выражения на Python™ в следующем виде:

```
Brush.use_frontface
bpy.data.brushes [«Blob»].use_frontface
```

(Add effect of brush) — добавление эффекта кисти. На Python™ выглядит так:

```
Brush.direction
bpy.data.brushes ["Blob"].direction
```

(Subtract effect of brush) — эффект вычитания кисти. На Python™ выражается так:

```
Brush.direction
bpy.data.brushes ["Blob"].direction
```

(Accumulate stroke daubs on top of each other) — последовательное накопление следа штриха. На Python™ выражается так:

```
Brush.use_accumulate
bpy.data.brushes ["Blob"].use_accumulate
```

Итак мы разобрались с инструментарием, необходимым для нашего проекта, с помощью которого создавались геометрические зарисовки, теперь необходимо определить его цветовые позиции и позиции групп оттеночных переходов, их можно представить в виде палитры цветов и градиентных переходов:

Можно также залить проект стандартными штатными текстурами, Blender® имеет достаточное их количество и гибкую настройку:

Прорисовка процедурными текстурами имеет свои нюансы в зависимости от сложности объекта. Например, бывает, что объект имеет сложную форму, определённое количество граней и т.д. Его можно заливать

Таблица 1. Числовые параметры распределения цветов (основные цвета) и точки их перехода, используемые в проекте.

Основные цвета	Средние оттенки	Производные оттенки
R #FF0E00 (255,14,0)	#EB4E0E (235,78,14)	#B5ADAA (181,173,170)
G #00FB38 (0,251,56)	#EE5D15 (238,93,21)	#FFD9C1 (C1D9FF)
B #001AF9(0,26,249)	#E7410D (231,65,13)	#EA4C0D (234,76,13)
	Цвет в точках перехода	Цвет в точках перехода
	#FFFFFF (255,255,255)	#FFFFFF (255,255,255)

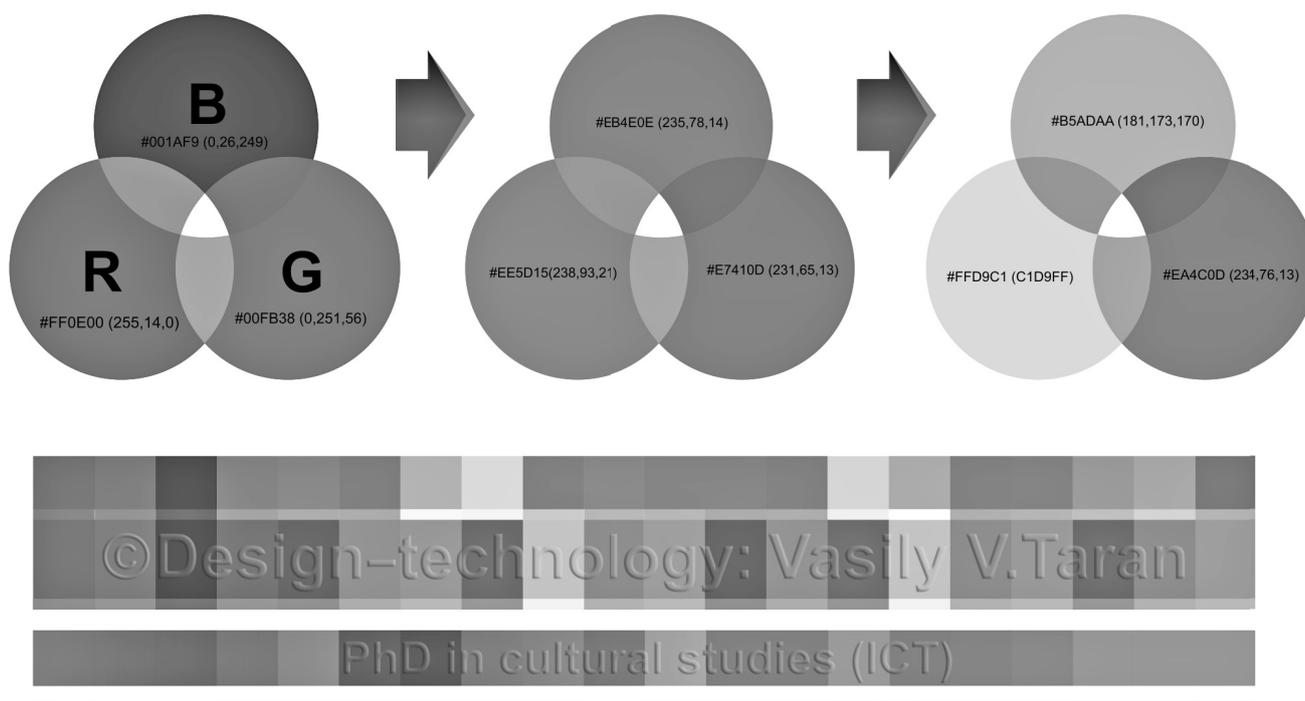


Рис. 6 Цветовая модель, отображающая используемые в проекте производные оттенки от палитры RGB.

целиком текстурой, и он впишется в композицию, другой вопрос, когда имеется необходимость регулировать каждую из частей объекта по отдельности, либо проецировать *двумерное изображение на трёхмерную модель*. Для этого предусмотрена технология UV-mapping<sup>1</sup>

<sup>1</sup> Прим. автора. (Таран В. В.) В русскоязычной научной литературе UV-преобразование либо развёртка в трёхмерной графике. UV-преобразование это процесс трёхмерного моделирования, связанный с проецированием двумерного изображения на поверхность трёхмерной модели для отображения текстуры. Буквы «U», «V» — обозначают оси двумерной текстуры, поскольку оси «X», «Y», «Z» — уже используются для обозначения осей трёхмерного объекта в пространстве модели, в то время как W в дополнении «XYZ» — используется при вычислении вращения кватернионов, что является обычной операцией в компьютерной графике. Кватернионы (по четыре), система гиперкомплексных чисел, образующая векторное пространство размерностью четыре над полем вещественных чисел. Обычно обозначается символом H, разработана У.Р. Гамильтоном (William Rowan

«U» — «V» картографирование, развёртка), процедура его настройки и применения хорошо изложена в книге Алана Брито (Allan Brito) «Невероятные машины». В книге на примере конкретных чертежей трёхмерных конструкций показаны основные опции по моделированию, отрисовке и анимации реалистичных машин с использованием программы трёхмерного моделирования Blender® [5].

Hamilton). В математике кватернионы — система чисел, которая расширяет комплексные (мнимые) числа. Они впервые применены У.Р. Гамильтоном (William Rowan Hamilton) в 1843 году к механике в трёхмерном пространстве. Особенностью кватернионов является то, что умножение двух кватернионов не коммутативно. Гамильтон определил кватернион, как *частное* от двух *направленных прямых* в *трёхмерном пространстве*, или что эквивалентно частному от двух векторов.

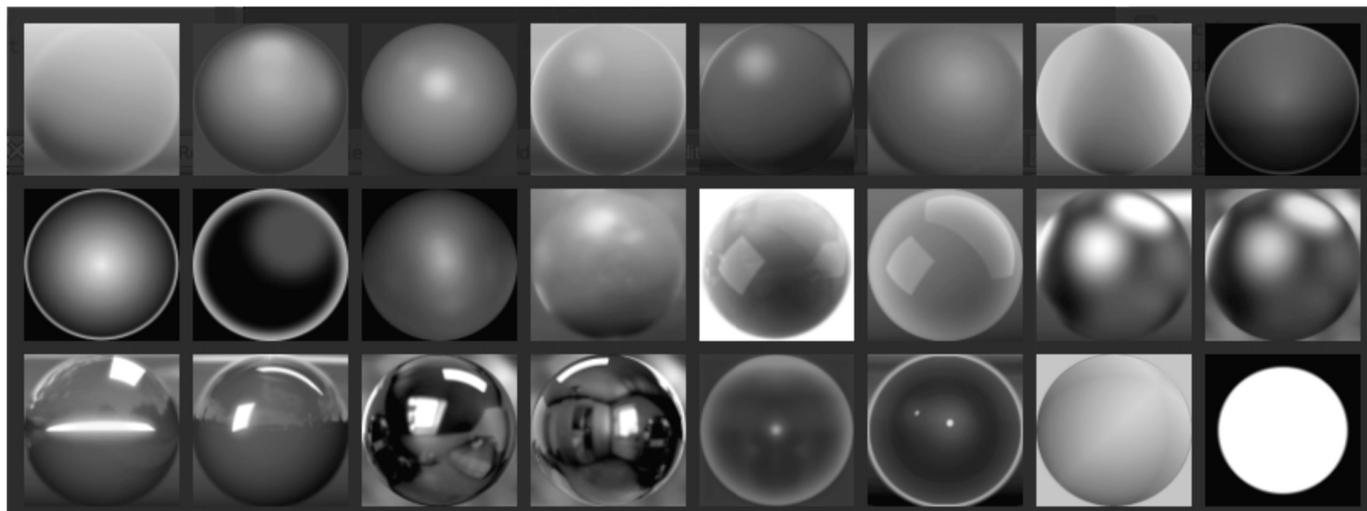
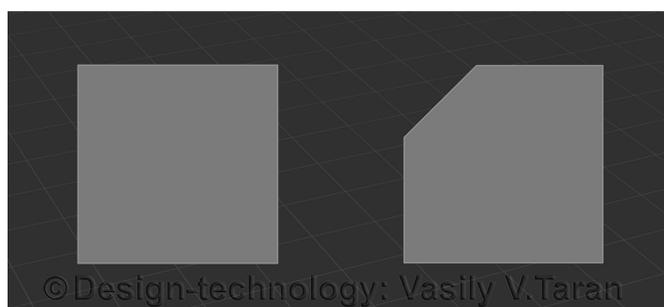


Рис. 7. Набор Внутренних текстур для заливки в программе Blender®

Далее (переходим в режим двухмерного редактирования) меняем положение плоскости на вертикальное и выставляем монтажные опорные точки конструкции таким образом, чтобы создать имитацию скоса<sup>1</sup> в левом верхнем углу плоскости. Из четырёхугольной плоскости получается пятиугольник.



Определяемся с группами слоёв (Group of layers), которые будут играть важную роль при ротации геометрической конструкции и при выполнении анимации будут скрывать в зависимости от перспективы внешнее и внутренне отражение некоторых углов. Схематично с процедурами анимации различных геометрических конструкций можно ознакомиться в книге Гордона Фишера (Gordon C. Fisher) под названием «Blender 3D Basics Beginner's Guide», несмотря на то, что книга написана для начинающих пользователей, автору удалось схематично показать основные аспекты анимации [6]. Здесь есть одно важное условие, что при переменности световых потоков и во время скрытия слоёв будет про-

<sup>1</sup> Имитация скоса на плоскости подразумевает создание пограничных точек пересечения с учётом будущего вращения объекта, который будет носить объёмный характер. То есть, эти точки будут видны только в двухмерной проекции, а в трёхмерной они будут скрыты.

исходить экструзия их обтравочного контура<sup>2</sup>. На плоскости те же слои будут скрыты (Значение LPS — Layers subject to hiding). Для более эффективного построения минискриптов<sup>3</sup> (создания классов) и оптимизации кода в целом можно использовать метаклассы [7]. Всего таких слоёв в первом случае будет 11 (L11) а во втором случае будет 12 (L12). В первом случае слои будут отвечать за внутреннее отражение перспективы (ER — External Reflexion), во втором случае слои будут отвечать за внешнее отражение перспективы (IR — Internal Reflexion). Визуально схема распределения и распространения световых потоков представлена на рисунках 12,13,14,15. Для начала представим объекты перспективы (двухмерного пространства) как простые списоч-

<sup>2</sup> Экструзия обтравочного контура здесь необходима для создания эффекта гладкости и скольжения ротируемого объекта. Если этот объект будет потенциально использоваться в видеозаставках на телевидении или же просто в титровой графике, то существует возможность импортирования этого объекта в другие программы 3D-проектирования, к примеру, в Autodesk 3ds Max (3Ds Studio MAX) или Maxon 4D CINEMA с последующим рендерингом в Maxon Redshift. И повторный экспорт графического объекта (в зависимости от типа рендеринга) может привести к изменению контурных линий, что скажется на не совсем корректном отображении объекта. Экструзия обтравочного контура служит своего рода страховочной маской особенно, если объект будет находиться в движении. Экструзия не позволит сужать линии до критического порога, и будет постоянно выдавливать обтравочные контуры наружу, после чего будет происходить плавное попеременное скрытие слоёв.

<sup>3</sup> **Прим. автора** (Таран В. В.) В книге Сунила Капила (Sunil Kapil) Clean Python: Elegant Coding in Python (Чистый Питон: Элегантное кодирование в Питоне) в главе 4-й «Работа с модулями и метаклассами» (Working with Modules and Metaclasses), довольно подробно рассмотрены схемы создания метаклассов см. стр.123–137. Некоторые примеры (листинги программ) к данной книге опубликованы по адресу: <https://github.com/Apress/clean-python> (дата обращения к источнику: 27 мая 2020 г.). Конкретно по главе 4, по адресу: <https://github.com/Apress/clean-python/tree/master/chapter04> (дата обращения к источнику: 27 мая 2020 г.).

$$['A', 'B', 'C', 'D'] = \text{Plane}$$

$$['A', 'B', 'C', 'D', 'E'] = \text{irregular pentagon}$$

$$P_{ER} = L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11} \xrightarrow[\text{LSH}]{L=L_{11}} ['A', 'B', 'C', 'D'];$$

$$P_{IR} = L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}, L_{12} \xrightarrow[\text{LSH}]{L=L_{12}} ['A', 'B', 'C', 'D'];$$

$$ip_{ER} = L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11} \xrightarrow[\text{LSH}]{L=L_{11}} ['A', 'B', 'C', 'D', 'E'];$$

$$ip_{IR} = L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}, L_{12} \xrightarrow[\text{LSH}]{L=L_{12}} ['A', 'B', 'C', 'D', 'E'];$$

Формула 1

$\frac{\text{Plane} = (AD + B + C)}{1}$	$\frac{\text{Plane} = (AD + BC)}{0,5}$	$\frac{\text{Plane} = (A + D + B + C)}{1}$	$\frac{\text{Plane} = (ADB + C)}{1}$
$\frac{\text{Plane} = (CD + B + A)}{0,5}$	$\frac{\text{Plane} = (CD + BA)}{3,5}$	$\frac{\text{Plane} = (C + D + B + A)}{0,5}$	$\frac{\text{Plane} = (CDB + A)}{0,5}$
$\frac{\text{Plane} = (BD + A + C)}{3,5}$	$\frac{\text{Plane} = (BD + AC)}{3,5}$	$\frac{\text{Plane} = (B + D + A + C)}{1}$	$\frac{\text{Plane} = (BDA + C)}{3,5}$
$\frac{\text{Plane} = (BD + C + A)}{5}$	$\frac{\text{Plane} = (BD + CA)}{1}$	$\frac{\text{Plane} = (B + D + C + A)}{0,5}$	$\frac{\text{Plane} = (BDC + A)}{5}$

Формула 2

ные конструкции<sup>1</sup> для упорядочивания объектов (формула 1).

Когда углы упорядочены, настраиваем слои отражения и устанавливаем геометрию распределения — обёртывания<sup>2</sup> и отрисовки<sup>3</sup> слоями граней и углов проекции

<sup>1</sup> Списочные конструкции показывают последовательность выполняемых действий, в нашем случае это распределение слоёв по степени их пограничной прозрачности (т.е. та прозрачность, которая будет возникать, когда на основе плоскости (неправильного пятиугольника) будет построена объёмная проекция. Последовательность на языке Python обуславливается кавычками как двухпарными, так и одинарными, которые могут чередоваться попеременно. Фактически кавычки — это разделитель последовательности.

<sup>2</sup> Обёртывание слоями (Repeated layer wrapping) — важная процедура, которая позволяет превратить трёхмерный сложный геометрический объект в обтекаемую форму, скрывающую грани углов при вращении по часовой и против часовой стрелок. В формулах сокращённо употребляется аббревиатура (RLW).

<sup>3</sup> Отрисовка слоями (Re-drawing of layers) — это техническая процедура, подразумевающая фрагментарную контурную отрисовку граней трёхмерного объекта (по заданной точками траектории) в момент его вращения. Фактически это изначально заложенная карта контуров, по которым прорисовывается каркас трёхмерного объекта.

в перспективах. Отражение имеет важное значение при перераспределении световых потоков, которые создают псевдоэкструзию обтравочного контура. Внутренне отражение слоёв (обычной плоскости)<sup>4</sup> может иметь следующий вид (формула 2).

Здесь распределены световые потоки, проходящие по сторонам плоскости, имеющие четырёхступенчатый проход, индексы — {0,5/1/3,5/5} устанавливают степень псевдопрозрачности для плавных переходов световых потоков друг к другу. Фактически получается световое обёртывание и отрисовка слоями граней и углов проекции в перспективах.

В первом случае (схематично) значения переходов будут выглядеть следующим образом (формула 3).

Во втором случае (формула 4).

<sup>4</sup> Плоскость используется для отражения трассированных лучей и при определённых координатах образует псевдоперспективу неправильного пятиугольника.

$$L_1 L_2 L_3 L_4 \left[ \begin{array}{c} ['AD', 'B', 'C,'] \\ [01, 02, 03, 04, 05] \\ 05, 04, 03, 02, 01 \end{array} \right] \rightarrow^* L_5 L_6 L_7^* \begin{array}{c} \text{Return } t = 05, 04, 03, 02, 01 \\ ; L_8 L_9 L_{10} L_{11} L_{12} \end{array}$$

$$\text{Plane} = \begin{array}{c} \angle A, B, C, D \\ CB + BD \rightarrow B \rightarrow A \\ (\text{Reflexion} = 1, 2 + 2, 1 (L_8, L_9)) \end{array}$$

$$['AD', 'B', 'C,'] + ['CD', 'B', 'A'] + ['BD', 'A', 'C'] + ['BD', 'C', 'A']$$

Формула 3

$$L_1 L_2 L_3 L_4 \left[ \begin{array}{c} ['AD', 'B', 'C,'] \\ [01, 02, 03, 04, 05] \\ 05, 04, 03, 02, 01 \end{array} \right] \rightarrow^* L_5 L_6 L_7^* \begin{array}{c} \text{Return } t = 05, 04, 03, 02, 01 \\ ; L_8 L_9 L_{10} L_{11} L_{12} \end{array}$$

$$\text{Plane} = \begin{array}{c} \angle A, B, C, D \\ CD + DA \rightarrow A \rightarrow B \\ (\text{Reflexion} = 1, 2 + 2, 1 (L_8, L_9)) \end{array}$$

$$['AD', 'BC'] + ['CD', 'BA'] + ['BD', 'AC'] + ['BD', 'CA']$$

Формула 4

$$L_1 L_2 L_3 L_4 \left[ \begin{array}{c} ['AD', 'B', 'C,'] \\ [01, 02, 03, 04, 05] \\ 05, 04, 03, 02, 01 \end{array} \right] \rightarrow^* L_5 L_6 L_7^* \begin{array}{c} \text{Return } t = 05, 04, 03, 02, 01 \\ ; L_8 L_9 L_{10} L_{11} L_{12} \end{array}$$

$$\text{Plane} = \begin{array}{c} \angle A, B, C, D \\ CA + AB \rightarrow D \rightarrow B \\ (\text{Reflexion} = 1, 2 + 2, 1 (L_8, L_9)) \end{array}$$

$$['A', 'D', 'B', 'C,'] + ['C', 'D', 'B', 'A'] + ['B', 'D', 'A', 'C'] + ['B', 'D', 'C', 'A']$$

Формула 5

В третьем случае (формула 5).

В четвёртом случае (формула 6).

Каждый из перечисленных случаев фрагмента светового перехода показывает изменения обтравочных контуров геометрической конструкции, как видно из четырёх формульных схем, ряды списочных значений меняются соответственно распределению световых потоков. Поскольку из плоскости при её ротации мы получаем не-

правильный пятиугольник, то внутренняя схема отображения светов будет выглядеть следующим образом

Пример № 1 (рис. 8).

AD — находится в позиции вектора D<sup>1</sup>, позиция вектора A — соответствует AD = d — a (формула 7).

<sup>1</sup> Позиционность векторов можно менять, а также увеличивать (либо уменьшать) в зависимости от граней и углов обёртывания конструкции,

$$L_1 L_2 L_3 L_4 \left[ \begin{array}{c} [AD', 'B', 'C'] \\ [01, 02, 03, 04, 05] \\ 05, 04, 03, 02, 01 \end{array} \right] \xrightarrow{L_5 L_6 L_7} \begin{array}{c} [+, '-', '+'] \\ \text{Return } t=05, 04, 03, 02, 01 \end{array} ; L_8 L_9 L_{10} L_{11} L_{12}$$

$$\text{Plane} = \underset{\substack{\angle A, B, C, D \\ (\text{Reflexion}=2, 1+1, 2 (L_8, L_9))}}{DC + BD} \rightarrow D \rightarrow A$$

$$['ADB', 'C'] + ['CDB', 'A'] + ['BDA', 'C'] + ['BDC', 'A']$$

$$A = \begin{bmatrix} a_7 & \dots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{p1} & \dots & a_{sp} \end{bmatrix}, D = \begin{bmatrix} d_7 & \dots & d_{1s} \\ \vdots & \ddots & \vdots \\ d_{p1} & \dots & d_{sp} \end{bmatrix}$$

$$A + D = \begin{bmatrix} a_7 + d_7 & \dots & a_{1s} + d_{1s} \\ \vdots & \ddots & \vdots \\ a_{p1} + d_{p1} & \dots & a_{sp} + d_{sp} \end{bmatrix}$$

$$A = \begin{bmatrix} a_7 & \dots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{p1} & \dots & a_{sp} \end{bmatrix}, D = \begin{bmatrix} d_7 & \dots & d_{1s} \\ \vdots & \ddots & \vdots \\ d_{p1} & \dots & d_{sp} \end{bmatrix}, A + D = \begin{bmatrix} a_7 + d_7 & \dots & a_{1s} + d_{1s} \\ \vdots & \ddots & \vdots \\ a_{p1} + d_{p1} & \dots & a_{sp} + d_{sp} \end{bmatrix}$$

Формула 6

$$AC = c - a$$

$$AD \times AC = (d - a) \times (c - a)$$

$$= d \times c - d \times a - a \times c + a \times a \quad (\because a \times a = 0)$$

$$= a \times d + d \times c + c \times a$$

$$\Rightarrow = \frac{1}{2} |a \times d + d \times c + c \times a| = 0$$

$$\text{Следовательно площадь } \Delta ADC = \frac{1}{2} |AD \times AC|$$

$$\Rightarrow = \frac{1}{2} |a \times d + d \times c + c \times a| = 0$$

$$\Rightarrow = |a \times d + d \times c + c \times a|$$

$$\Rightarrow = a \times d + d \times c + c \times a$$

Формула 7

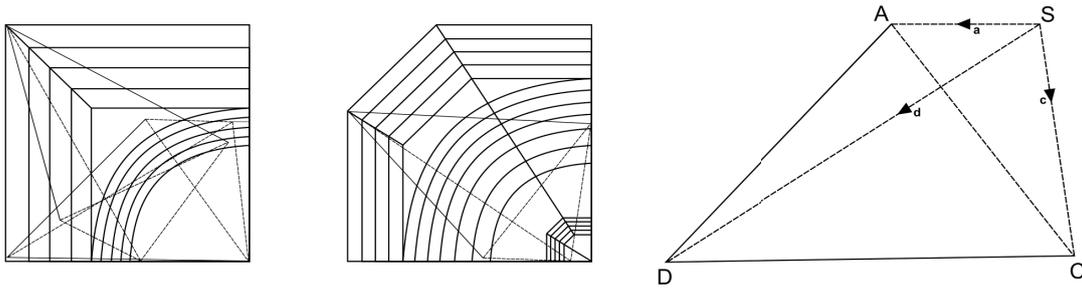


Рис. 8. Геометрия распределения и обёртывания слоями пограничных зон внутри конструкции.

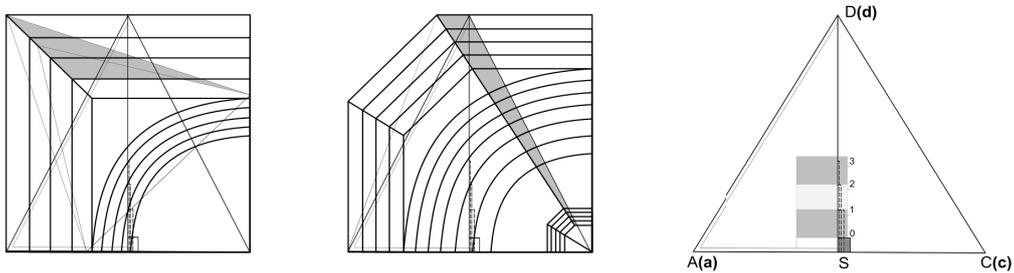


Рис. 9 Геометрия распределения и обёртывания слоями пограничных зон фронтальных частей конструкции.

$$\begin{aligned} \therefore \frac{1}{2}|AC||DS| &= \frac{1}{2}|a \times b + b \times c + c \times a| \\ \Rightarrow DS &= \frac{|a \times c + c \times d + d \times a|}{|c - a|} \end{aligned}$$

['AC', 'C', 'B', 'E'] + ['DB', 'B', 'A', 'E'] + ['BE', 'D', 'C', 'A'] + ['DE', 'B', 'A', 'C'] + ['BC', 'A', 'E', 'D']			
$\frac{ip = (AC + C + B + E)}{1}$	$\frac{ip = (CB + AC + B)}{1}$	$\frac{ip / RLW = (C + B + A + C + B)}{1}$	$\frac{ip = (CBA + C + E)}{1}$
$\frac{ip = (DB + B + A + E)}{0,5}$	$\frac{ip = (BA + AE + A)}{0,5}$	$\frac{ip / RLW = (B + A + A + E + A)}{0,5}$	$\frac{ip = (AEC + D + A)}{0,5}$
$\frac{ip = (BE + D + C + A)}{3,5}$	$\frac{ip = (BD + DC + A)}{3,5}$	$\frac{ip / RLW = (B + D + D + C + A)}{3,5}$	$\frac{ip = (BDE + A + D)}{3,5}$
$\frac{ip = (DE + B + A + C)}{5}$	$\frac{ip = (BE + BC + D)}{5}$	$\frac{ip / RLW = (B + E + B + C + D)}{5}$	$\frac{ip = (ECA + A + B)}{5}$
$\frac{ip = (BC + A + E + D)}{0,1}$	$\frac{ip = (CA + BD + E)}{0,1}$	$\frac{ip / RLW = (C + A + B + D + E)}{0,1}$	$\frac{ip = (CAE + D + E)}{0,1}$

Формула 9

$$L_1L_2L_3L_4 \left[ \begin{array}{c} [AC, C, B, E] \\ [01, 02, 03, 04, 05] \\ 05, 04, 03, 02, 01 \end{array} \right] \rightarrow^* L_5L_6L_7^* \begin{array}{c} [+, +, +, +] \\ \text{Return } t=05, 04, 03, 02, 01 \end{array} ; L_8L_9L_{10}L_{11}L_{12}$$

$$ip = \begin{array}{c} \triangle A, B, C, D, E \\ AC + DB \rightarrow B \rightarrow E \\ (\text{Reflexion}=1, 2+2, 1 (L_{10}, L_{11})) \end{array}$$

$$[AC, C, B, E] + [DB, B, A, E] + [BE, D, C, A] + [DE, B, A, C] + [BC, A, E, D]$$

Формула 10

$$L_1L_2L_3L_4 \left[ \begin{array}{c} [AC, C, B, E] \\ [01, 02, 03, 04, 05] \\ 05, 04, 03, 02, 01 \end{array} \right] \rightarrow^* L_5L_6L_7^* \begin{array}{c} [+, +, +, +] \\ \text{Return } t=05, 04, 03, 02, 01 \end{array} ; L_8L_9L_{10}L_{11}L_{12}$$

$$ip = \begin{array}{c} \triangle A, B, C, D, E \\ CB + BA \rightarrow B \rightarrow A \\ (\text{Reflexion}=1, 2+2, 1 (L_{10}, L_{11})) \end{array}$$

$$[CB, AC, B] + [BA, AE, A] + [BD, DC, A] + [BE, BC, D] + [CA, BD, E]$$

Формула 11

$$L_1L_2L_3L_4 \left[ \begin{array}{c} [AC, C, B, E] \\ [01, 02, 03, 04, 05] \\ 05, 04, 03, 02, 01 \end{array} \right] \rightarrow^* L_5L_6L_7^* \begin{array}{c} [+, +, +, +] \\ \text{Return } t=05, 04, 03, 02, 01 \end{array} ; L_8L_9L_{10}L_{11}L_{12}$$

$$ip = \begin{array}{c} \triangle A, B, C, D, E \\ CB + AE \rightarrow D \rightarrow A \\ (\text{Reflexion}=1, 2+2, 1 (L_{10}, L_{11})) \end{array}$$

$$[CBA, C, E] + [AEC, D, A] + [BDE, A, D] + [ECA, A, B] + [CAE, D, E]$$

Формула 12

При условии, что точки A, D, C — коллинеарны, соответственно сектор межевания A, D, C = 0.

Перпендикулярное расстояние d от присоединения к точкам a и c

$$\frac{|a \times d + d \times a + a \times c|}{|c - a|}$$

Пусть ABC будут треугольником, пусть a, b, c будут позицией его вершин A, B, C и пусть CM лежат перпендикулярно C на AB. Когда площадь:

$$\Delta ACD = \frac{1}{2} (AC) \cdot (DS) = \frac{1}{2} |AC| |DS|$$

$$\Delta ABC = \frac{1}{2} |a \times c + c \times d + d \times a|$$

Пример№ 2 (рис. 9, формула 9).

Здесь, в первом случае (формула 10).

Во втором случае (формула 11).

В третьем случае (формула 12).

$$L_1 L_2 L_3 L_4 \left[ \begin{array}{c} [AC, 'C', 'B', 'E'] \\ [01, 02, 03, 04, 05] \\ 05, 04, 03, 02, 01 \end{array} \right] \rightarrow^* L_5 L_6 L_7^* ; L_8 L_9 L_{10} L_{11} L_{12}$$

Return t=05,04,03,02,01

ip / RLW =  $\begin{matrix} \angle A, B, C, D, E \\ CB + AE \rightarrow D \rightarrow A \\ \text{(Reflexion=1,2+2,1 (L_{10}, L_{11} \leftrightarrow L_{12}))} \end{matrix}$

['C', 'B', 'A', 'C', 'B'] + ['B', 'A', 'A', 'E', 'A'] + ['B', 'D', 'D', 'C', 'A'] + ['B', 'E', 'B', 'C', 'D'] + ['C', 'A', 'B', 'D', 'E']

$$\text{Plane} = (AC + AD + AE) + CB + BD + EB)$$

Reflexion=1,2+2,1 (L<sub>8</sub>+L<sub>9</sub>)

$$= (AC + CB) + (AD + DB) + (AE + EB)$$

Reflexion=2,1+1,2 (L<sub>7</sub>+L<sub>9</sub>)

$$= AB + \underset{R=2,1(L_5)}{AB} + AB = 3AB$$

$$AB = DB + AC = \frac{1}{5} a + 3f$$

Формула 13

$$\begin{array}{cccccc} L_1 = A+B+C & L_3 = B+C+D & L_5 = B+E+D & L_7 = A+E+D & L_9 = A+D+B \\ \text{[LDA CL]} & \text{[LDA CL]} & \text{[LDA CL]} & \text{[LDA CL]} & \text{[LDA CL]} \\ L_2 = A+C+D & L_4 = B+C+D & L_6 = C+D+E & L_8 = A+C+B & L_{10} = D+E+A \\ \text{[LDA CL]} & \text{[LDA CL]} & \text{[LDA CL]} & \text{[LDA CL]} & \text{[LDA CL]} \\ L_{11} = \text{LIGHT } [A+B+C+A+C+D+B+C+D+B+C+D+B+E \\ +D+C+D+E+A+E+D+A+C+B+A+D+B+D+E+A] \\ \text{[LDA CL]} \end{array}$$

$$| \text{Smooth\_flow}(S) | = \left| -\frac{1}{8}n \right| = \frac{1}{8} | p | \Rightarrow | c | > | a | \xrightarrow{\leftarrow} \left| \frac{AB}{ED} + \frac{CA}{CD} \right|$$

Формула 14

В четвёртом случае (формула 13)

После определения соотношения слоёв перейдём к распределению света вдоль линии обтравки. Слои обтравочного контура обозначаются как LDA CL (Light distribution along the clipping line) и необходимы для более гладкой световой прорисовки границ вращения проекции (формула 14).

При расширении (ext) объекта конструкции по направлениям светового потока так, чтобы вершины избежали выхода за критический обтравочный слой края (радиус окружности края, при этом R=95), можно задать дополнительные расширения (формула 15).

Гладкая световая прорисовка<sup>1</sup> позволяет смягчить контуры ротируемых сторон объекта, при этом равномерно распределяя световые потоки. При рендеренге все контуры ротируемых слоёв должны просматриваться и иметь ширину 0,5 (формула 16).

Встраиваемый фрагмент кода (после описанных выше действий) будет выглядеть так (формула 17).

<sup>1</sup> При необходимости настройки световых потоков в соответствии с координатами RGB (может пригодиться при композиционном видеомонтаже) ввод данных в сценарий может осуществляться через: r\_... = ...[r\_...], g\_... = ...[g\_...], b\_... = ...[b\_...]. Например, для введения зелёного свечения и производных от него оттенков значение примет следующий вид g\_set = ips [g\_set].

$$L_1 = \frac{A+B+C}{4.40439, 11.4668, 1.21199} \quad [LDA\text{CL}]$$

$$L_2 = \frac{A+C+D}{-7.69592, -16.348, 0.874572} \quad [LDA\text{CL}]$$

$$L_3 = \frac{B+C+D}{4.08646, 5.06537, -3.39744} \quad [LDA\text{CL}]$$

$$L_4 = \frac{B+C+D}{3.5508, 7.72708, -0.253974} \quad [LDA\text{CL}]$$

$$L_5 = \frac{B+E+D}{-0.485498, 0.889475, 1.61357} \quad [LDA\text{CL}]$$

$$L_6 = \frac{C+D+E}{-0.939774, -2.51842, -0.31694} \quad [LDA\text{CL}]$$

$$L_7 = \frac{A+E+D}{-0.264682, -0.295853, 0.246158} \quad [LDA\text{CL}]$$

$$L_8 = \frac{A+C+B}{-0.533492, -1.27225, -0.052202} \quad [LDA\text{CL}]$$

$$L_9 = \frac{A+D+B}{2.08609, 0.82976, -3.15903} \quad [LDA\text{CL}]$$

$$L_{10} = \frac{D+E+A}{-2.07469, -3.54453, 0.935606} \quad [LDA\text{CL}]$$

LIGHT

$$\frac{[A+B+C+A+C+D+B+C+D+B+C+D+B+E]}{[LDA\text{CL}]}$$

$$B+C+A+C+D \rightarrow \begin{matrix} \mathbf{D, B, C, D} \\ \downarrow \downarrow \downarrow \downarrow \\ \begin{matrix} -1 & 1 & 1 & 0 & 1 \\ (-1, -1), (-1, 1), (1, -1), (1, 1), (1, 0) \\ (1, -0), (-0, 1), (0, -1), (1, 1), (0, 1) \end{matrix} \end{matrix}$$

$$L_{11} = \frac{= 0,1,2,3,4,5...}{+D+C+D+E+A+E+D+A+C+B+A+D+B+D+E+A} \quad [LDA\text{CL}] \quad [r = .95]$$

Формула 15

$$[LDA\text{CL}] = \frac{\text{bpy.context.scene.render.resolution}_x = 1920}{\text{bpy.context.scene.render.resolution}_y = 1080} \frac{[A+B+C+A+C+D+B+C+D+B+C+D+B+E+D+C+D+E+A+E+D+A+C+B+A+D+B+D+E+A]}{[A+B+C+A+C+D+B+C+D+B+C+D+B+E+D+C+D+E+A+E+D+A+C+B+A+D+B+D+E+A]}$$

Формула 16

После небольшой автоматизации некоторых рутинных действий следует приступить к прорисовке текстового содержания на неправильном выпуклом пятиугольнике. Здесь устанавливаем координатные значения внутренней обтравочной линии с каждой стороны по 0,16 дюймов. Получается внутренний холст для зарисовки наименования телекомпании. Внутри холста

располагаются цифры «24», и сокращённое наименование канала «ТВР».

Буквы должны быть расположены внутри холста, в противном случае анимация обтравочного контура будет скрывать те части, которые выходят за границы холста. Это обстоятельство касается и смежных сторон не-

```

{
IP_ER [0,1] [0,5] = IR [1] [5] * 0 + 1;
IP_ER [0,5] [3,5] = IR [5] [1] * 1 + 0;
IP_ER [0,5] [0,1] = IR [1] [5] * 0 + 1;
IP_ER [3,5] [0,5] = IR [5] [1] * 1 + 0;
IP_ER [3,5] [0,1] = IR [1] [5] * 0 + 1;
}

```

Формула 17

правильного пятиугольника, эти стороны должны быть также обтекаемы холстом.



Буквы и цифры отрисовываются вручную, на данном этапе холст автоматически размечает точками грани соединения и углы наклона



В режиме Brush (кисть) выбираем опцию Vertex Paint (Рисование вершинами) на Python™ это выражение

```

Brush.use_paint_vertex
bpy.data.brushes["Blob"].use_paint_vertex

```

Экструзивным методом выталкиваем поверхностные грани в пределах внутреннего каркаса конструкции. Далее следуем прорисовке шага движения мыши и выбираем опцию точек (Dots)

```

Brush.stroke_method
bpy.data.brushes["Blob"].stroke_method
Инструмент скульптинга регулируется
Brush.sculpt_tool
bpy.data.brushes["Blob"].sculpt_tool

```

Точки позволяют динамично изменять обтравочный контур. Более того эти же точки можно использовать для точной настройки плавного перетекания и выпадания

теней вдоль объекта. Фронтальная часть в миниатюре будет иметь следующий вид



Расстановка контурных точек позволит производить сложные манипуляции и дробить трёхмерный объект на двумерные конструкции, каждая из которых может быть соединена точками, либо представлена в виде объединённого каркаса.

Рисунок 10 фрагментарно показывает основные процессы преобразования проектных фигур с помощью точек. Для установления баланса плотности точек<sup>1</sup> можно ввести следующие значения (формула 18).

После создания обтравочного контура необходимо произвести настройку внутренних и внешних теней для качественного объёмного отображения данной проекции на мониторе компьютера и для телевизионного вещания. Для этих целей необходимо создать сетку переменных для теней во внутренних преломлениях.

<sup>1</sup> Баланс плотности точек может пригодиться при выполнении трансформации простых фигур в более сложные. В нашем случае  $L_a$  — главный слой прозрачности, на фрагменте рисунка 10, под номером 3<sup>a</sup> — обозначен как пограничный слой (L). aD (average density) — средняя плотность, объём средней плотности — (vol, "density"), pD (planimetric density) — контурная плотность.

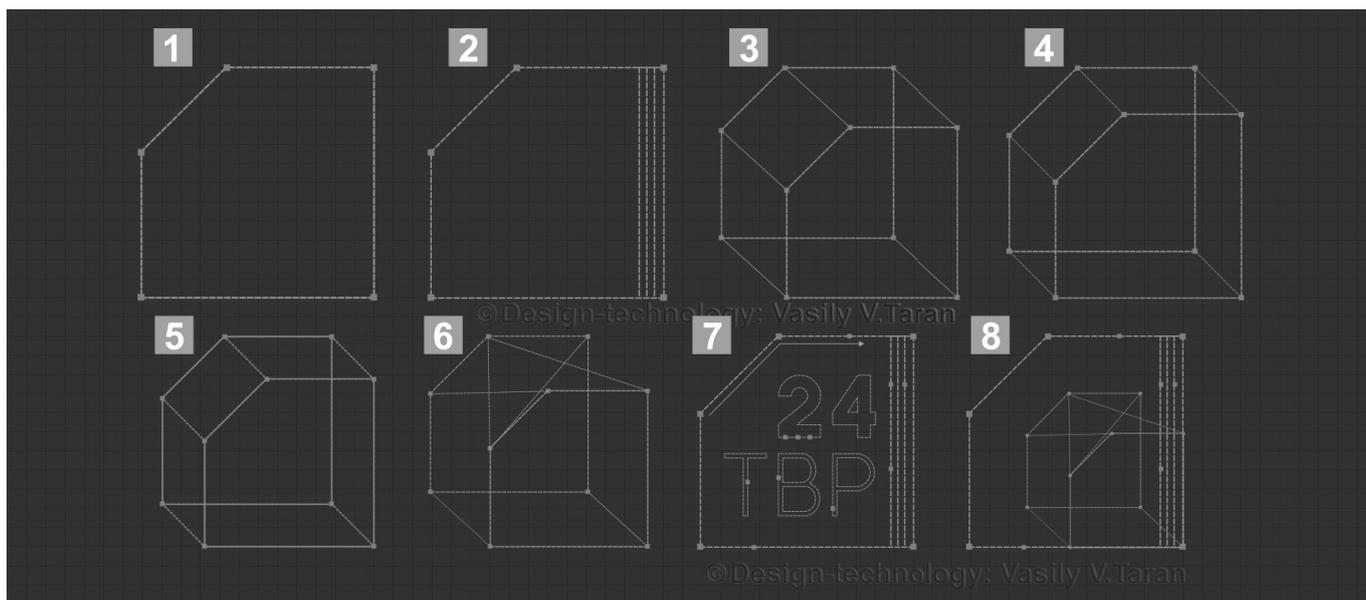
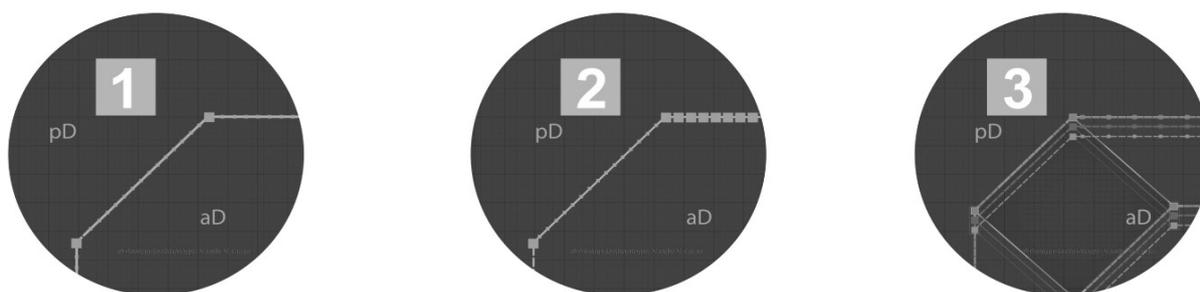


Рис. 10. Процедура, иллюстрирующая расстановку контурных точек для манипуляций конструкцией.



$$L_a [L]= aD, pD= 0.0, 1.0 \quad (1)$$

$$L_a [L]= aD, pD= 0.0, 3.0 \quad (2)$$

$$L_a [L]= aD, pD= 0.0, 5.0 \quad (3)$$

```
density.sim().prop(aD, "line", expand = True)
density.sim(percentage = 1.0)
density.sim().prop(pD, "line", expand = True)
density.sim(percentage = 0.3)
density.sim().prop(pD, "line", expand = True)
density.sim(percentage = 5.0)
```

Формула 18

```

if MODE==0:
    for v in me.verts:
        i= v.index
        if VSEL[i]:
            wA, wB = grad_weights[i]
            if wA != None: # and wB
                if TOALPHA:
                    # Do alpha by using the exiting weight for
                    try:         pickValB= vWeightDict[i][act_group]
                    except: pickValB= 0.0 # The weights not there? assume zero
                # Mix2 2 opaque weights
                vWeightDict[i][act_group]= pickValB*wA + pickValA*wB

```

Формула 19

```

else: # MODE==1 VCol
    for f in me.faces:
        if f.sel:
            f_v= f.v
            for i in xrange(len(f_v)):
                v= f_v[i]
                wA, wB = grad_weights[v.index]

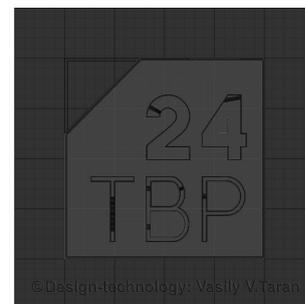
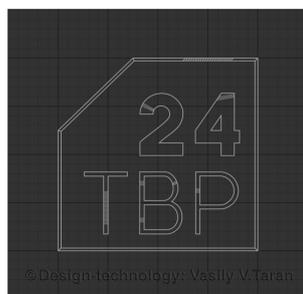
                c= f.col[i]

                if TOALPHA:
                    pickValB= c.r, c.g, c.b

                c.r = int(pickValB[0]*wA + pickValA[0]*wB)
                c.g = int(pickValB[1]*wA + pickValA[1]*wB)
                c.b = int(pickValB[2]*wA + pickValA[2]*wB)

```

Формула 20



После создания первичного слоя сетки переменных теней создадим сетку псевдообъёмности для сглаживания цветовых переходов.

После данной процедуры определимся с координатами прозрачности внутренних теней.

```

lineAB_length= (xA[1]-xB[1]).length

# normalzie
wA= wA/wTot
wB= wB/wTot

if ORTHO: # Con only use line length method with parellele lines
    if wTot > lineAB_length+eps:
        # vert is outside the range on 1 side. see what side of the grad
        if wA>wB:      wA, wB= 1.0, 0.0
        else:         wA, wB= 0.0, 1.0
    else:
        # PERSP, lineA[0] is the same origin as lineB[0]

        # Either xA[0] or xB[0] can be used instead of a possible x_mid between
        the 2
        # as long as the point is inbetween lineA and lineB it dosent matter.
        a= AngleBetweenVecs(lineA[0]-xA[0], line_mid)
        if a>line_angle:
            # vert is outside the range on 1 side. see what side of the grad
            if wA>wB:      wA, wB= 1.0, 0.0
            else:         wA, wB= 0.0, 1.0

    return wA, wB

grad_weights= [grad_weight_from_co(v) for v in me.verts]

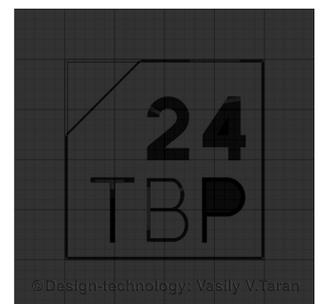
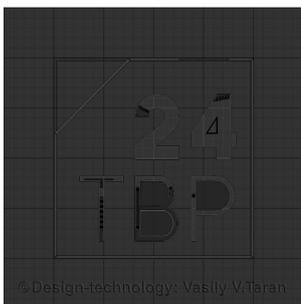
```

Формула 21

if wA > wB :

$$\begin{aligned}
 & \begin{matrix} wA, wB \rightarrow \\ \text{else:} \end{matrix} \begin{pmatrix} wA, wB = 1.0, 0.0 \\ wA, wB = 0.0, 1.0 \end{pmatrix} \\
 & \rightarrow \begin{pmatrix} wA, wB = 0.0, 1.0 \\ wA, wB = 1.0, 0.0 \end{pmatrix} \rightarrow \begin{pmatrix} wA, wB = 0.0, 0.0 \\ wA, wB = 0.0, 0.0 \end{pmatrix}
 \end{aligned}$$

Формула 22



За тем скорректируем координаты прозрачности внешних теней.

Немаловажную роль в настройке внешних и внутренних теней для качественного отображения трёхмерной

$$\begin{aligned}
 & [\text{'AD'}, \text{'B'}, \text{'C'}] + [\text{'CD'}, \text{'B'}, \text{'A'}] + [\text{'BD'}, \text{'A'}, \text{'C'}] + [\text{'BD'}, \text{'C'}, \text{'A'}] \\
 & [\text{'AD'}, \text{'BC'}] + [\text{'CD'}, \text{'BA'}] + [\text{'BD'}, \text{'AC'}] + [\text{'BD'}, \text{'CA'}] \\
 & [\text{'A'}, \text{'D'}, \text{'B'}, \text{'C'}] + [\text{'C'}, \text{'D'}, \text{'B'}, \text{'A'}] + [\text{'B'}, \text{'D'}, \text{'A'}, \text{'C'}] + [\text{'B'}, \text{'D'}, \text{'C'}, \text{'A'}] \\
 & [\text{'ADB'}, \text{'C'}] + [\text{'CDB'}, \text{'A'}] + [\text{'BDA'}, \text{'C'}] + [\text{'BDC'}, \text{'A'}] \\
 & [\text{'AC'}, \text{'C'}, \text{'B'}, \text{'E'}] + [\text{'DB'}, \text{'B'}, \text{'A'}, \text{'E'}] + [\text{'BE'}, \text{'D'}, \text{'C'}, \text{'A'}] + [\text{'DE'}, \text{'B'}, \text{'A'}, \text{'C'}] + [\text{'BC'}, \text{'A'}, \text{'E'}, \text{'D'}] \\
 & [\text{'AC'}, \text{'C'}, \text{'B'}, \text{'E'}] + [\text{'DB'}, \text{'B'}, \text{'A'}, \text{'E'}] + [\text{'BE'}, \text{'D'}, \text{'C'}, \text{'A'}] + [\text{'DE'}, \text{'B'}, \text{'A'}, \text{'C'}] + [\text{'BC'}, \text{'A'}, \text{'E'}, \text{'D'}] \\
 & [\text{'CB'}, \text{'AC'}, \text{'B'}] + [\text{'BA'}, \text{'AE'}, \text{'A'}] + [\text{'BD'}, \text{'DC'}, \text{'A'}] + [\text{'BE'}, \text{'BC'}, \text{'D'}] + [\text{'CA'}, \text{'BD'}, \text{'E'}] \\
 & [\text{'CBA'}, \text{'C'}, \text{'E'}] + [\text{'AEC'}, \text{'D'}, \text{'A'}] + [\text{'BDE'}, \text{'A'}, \text{'D'}] + [\text{'ECA'}, \text{'A'}, \text{'B'}] + [\text{'CAE'}, \text{'D'}, \text{'E'}] \\
 & [\text{'C'}, \text{'B'}, \text{'A'}, \text{'C'}, \text{'B'}] + [\text{'B'}, \text{'A'}, \text{'A'}, \text{'E'}, \text{'A'}] \\
 & + [\text{'B'}, \text{'D'}, \text{'D'}, \text{'C'}, \text{'A'}] + [\text{'B'}, \text{'E'}, \text{'B'}, \text{'C'}, \text{'D'}] + [\text{'C'}, \text{'A'}, \text{'B'}, \text{'D'}, \text{'E'}]
 \end{aligned}$$

Формула 23

$$\text{lineAB\_length} = (xA[.] - xB[.]) \cdot \frac{xA[0] \text{ or } xB[0]}{x\_mid \mid \text{line\_mid}} \rightarrow \left. \begin{array}{c} \text{lineA} \\ \updownarrow \\ [0] \\ \text{lineB} \end{array} \right\}$$

Формула 24

конструкции на мониторе (учитывая его степень отражательной способности — глянцевое покрытие или матовое) играет собирание информации о ступенчатых переходах, формирующих вершины, естественно они могут иметь разную степень сложности и уровень градиентности. Для снятия подобной информации, не предназначенной для прямого использования, можно использовать сценарий внешних ресурсов, в качестве примера приведём несколько фрагментов кода (формула 19).

Фрагмент кода иллюстрирует режим настройки веса градиента и установку степени его прозрачности (формула 20)..

Режим 1. (настройка) — индексно регулирует уровни веса градиента, устанавливает соотношения объёма градиентной прорисовки (Формула 21).

lineAB настраивает метод контроля длины строки параллельных линий, находящихся вне диапазона (формула 22).

$xA[0]$ ,  $xB[0]$  могут использоваться вместо  $x\_mid$ . Список всех списочных (конфигурируемых световых потоков) и их значений указанных в статье, представлены в формуле 23.

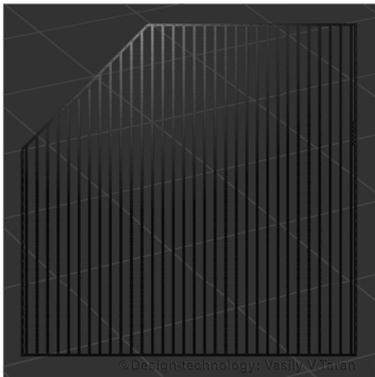
Метод контроля длины строки параллельных линий, также позволит упростить группировку объектов находящихся вне диапазона (формула 24).

Все процедуры распределения теней зависят от световых потоков и их переходов, в компьютерной среде трёхмерного моделирования Blender®, их можно имитировать различными видами освещения и типами ламп. Point, Sun, Spot, Hemi, Area — вот основной список источников для манипуляции светом. В нашем случае основными манипуляторами света выступают — «Sun» и «Spot». Опция Sun — подразумевает использование светового потока равной интенсивности вне зависимости от расположения объектов. Опция Spot, напротив устанавливает точечный световой поток и будет использоваться для подсветки проходящих вдоль линий. Источник ис-

пользует световой поток в виде «пятна» и очень удобен для манипуляций световой прорисовкой. Обе опции регулируются языком Python™

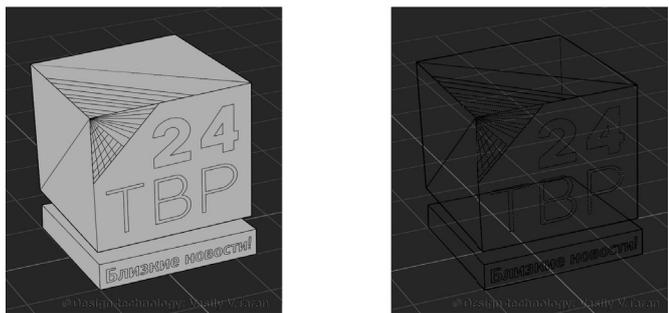
```
bpy.ops.object.lamp_add(type='POINT')
```

После настройки теней и теневых переходов в установленных границах можно перейти к созданию сетки заливки двумерной плоскости для установления переменных бликов в кадре телевидения.

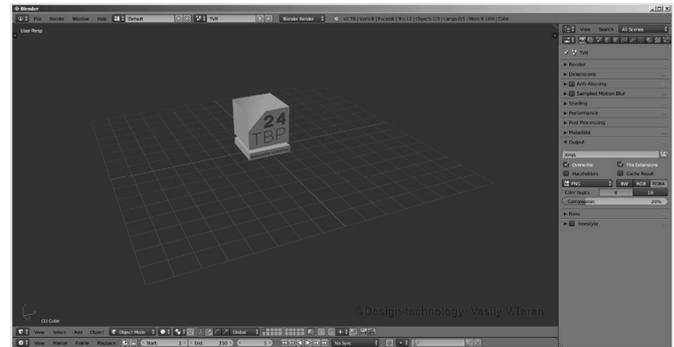


Сетка создаётся по аналогии с процессами плавного перетекания, необходимые точки устанавливаются таким образом, чтобы каждая из линий находилась равноудалённо друг от друга, кроме начальной и конечной позиций. Начальная и конечная позиции выделяются отдельными линиями, введёнными в конструкцию для уплотнения переходящего светового потока и повышения отражательной способности объекта. Переменные блики от источника света попадают на линии и в зависимости от их оптического приближения либо отдаления создают переменные блики.

По окончании процедуры мы получаем сетчатую конструкцию без горизонтальных перекрытий. Вертикальные линии позволяют высвободить блики при ротации объёмного изображения. Далее мы переходим к формированию трёхмерной модели на основе разработанной нами двумерной конструкции. Преобразуем двумерные координаты в объём с помощью увеличения точек пересечений. Получается следующая конструкция



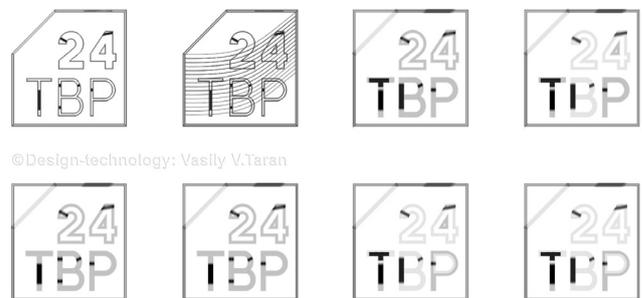
Получился куб с выделенными сторонами и дополнительными линиями пересечений. Справа отмечены точки координат, в результате которых происходит зарисовка псевдофронтальной проекции фигуры. По окончании данной процедуры выполняем настройку общего освещения, и получается следующая модель



Как видим, получается объёмная проекция со скошенным углом, фактически объёмный неправильный выпуклый пятиугольник (трёхмерный одиннадцатигуольник) неправильный объёмный семигранник.



Далее переходим к пропорциям светлых теней, образующихся в результате ротации объекта, их лучше представить на двумерной плоскости



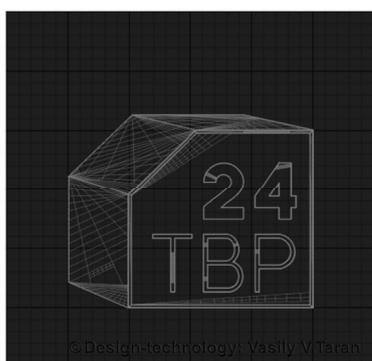
После объединения прозрачных слоёв, с фронтальной стороны (в зависимости от углов преломления светлых теней) объёмная проекция будет выглядеть следующим образом

```
aES = port.faces()
aES.port(line, "1")
aES.port(line, "2")
aES.port(line, "3")
aES.port(line, "4")
aES.port(line, "5")
```

Формула 25



Когда внутренние и внешние тени распределены и упорядочены, приступаем к сборке модельной конструкции.

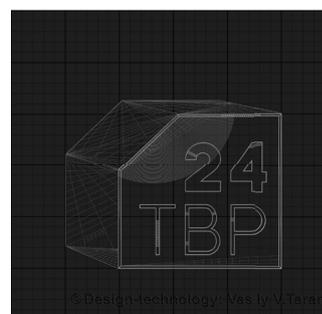


Модельная конструкция представляет собой сетчатую проекцию с изменяющимися углами, грани остаются на прежнем месте. Фактически это та же самая плоскость, имеющая вид неправильного пятиугольника. Неправильный пятиугольник получается в результате скрытия некоторых углов плоскости и путём преломления (благодаря точкам распределения света, и точкам регулировки интенсивности прорисовки граней) линий, образующих новые углы. Посмотрим на проекцию в усечённом виде



Далее следует процедура проверки прорисовки граней на дополнительные внешние тени<sup>1</sup>. Для этого можно использовать следующий код (формула 25).

Здесь aES — Additional external shades (дополнительные внешние тени). Port — Portrayal (прорисовка), port.faces (устанавливает грани прорисовки).



Блик, образуемый на конструкции, обозначает что тени прорисованы верно. Убедились, что внешние тени удовлетворяют проекции, теперь проследим, чтобы цифры были покрыты бликами сверху



<sup>1</sup> Дополнительные внешние тени могут образовываться в силу разных причин. Чаще всего данный эффект можно наблюдать при масштабировании объекта, выполняющего ротацию в результате задержек обновлений фреймов монитора, зависящих от тактовой частоты обновления экрана и глубины прорисовки объёмных фигур. От не совсем корректного представления цветов (битых пикселей) в результате нарушения цифрового цветосмешивания и дисбаланса аддитивных значений палитры RGB. А также от чисто оптических нарушений, влекущих за собой механические блики — в основном на мониторах глянцевого типа, при изменении угла наклона монитора, в условиях просмотра движущегося 3D-объекта и при других оптических эффектах.

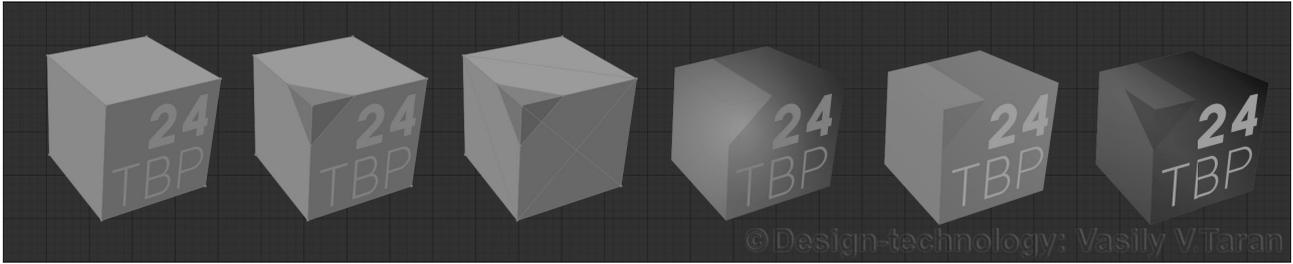
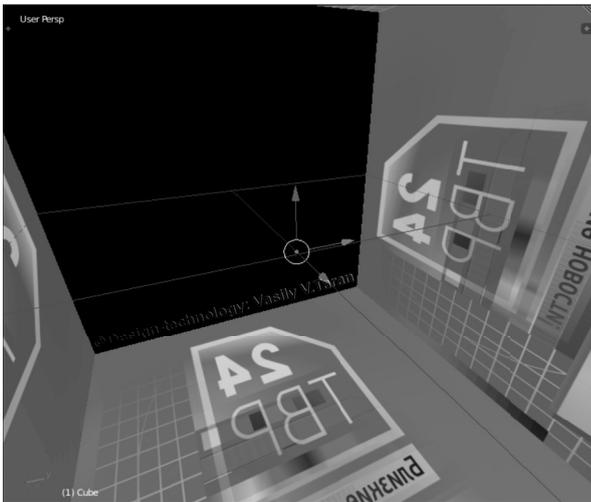
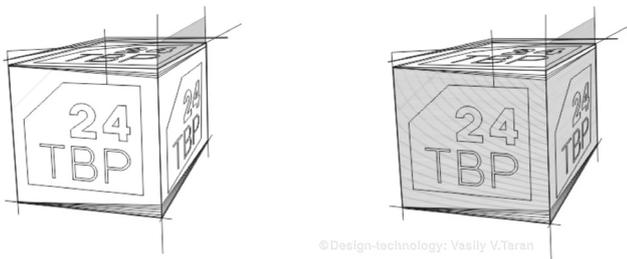


Рис. 11. Внешние части фигуры, подлежащие трансформации и отображающиеся послойно.

Убедившись, что цифры покрыты бликом, это необходимо для красоты логотипа, и создания эффекта переходящего блеска на цифрах, и теперь переходим к вывертке объёмной проекции

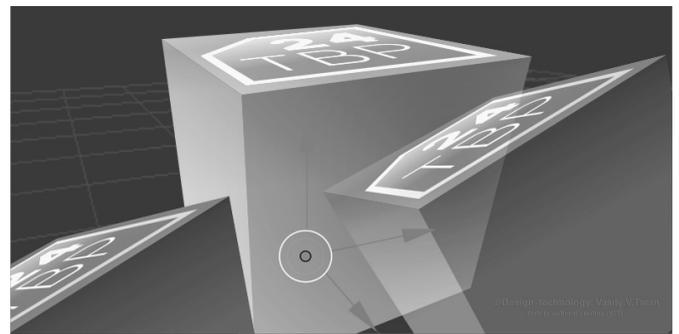


Видим, что стыки сторон пространства внутри куба имеют плавные границы перехода, внутренняя часть с внешней стороны (фактически вывертка) может выглядеть следующим образом



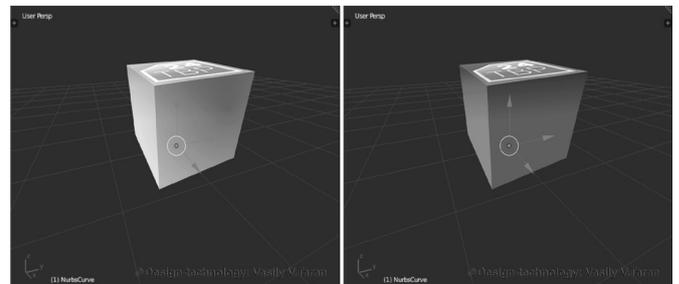
Зарисовка технически показывает, как выглядит фигура в перспективах. Слева объёмная проекция показывает отсутствие бликов, а справа на проекции блики присутствуют. Верхние части линий скрываются при трансформации объекта (в нашем случае это неправильный пятиугольник) и служат дополнительными прорисовочными линиями при изменении геометрии объекта

(включая анимацию). Внешние части фигуры будут отображены послойно следующим образом



На данном рисунке отображаются внешние поля вывертки, которые замыкаются в определённых точках при ротации. Ротация образует блики (и прозрачность), которые распределяются в соответствии со схемой ниже (Рисунок 12)

Верхние точки определяют световые потоки (в соответствии с описаниями) и распределяют их в соответствии с картой слоёв, описанной выше. Данная иллюстрация это фрагмент внутренней части нашей геометрической конструкции, которая впоследствии и будет преобразовываться в неправильный пятиугольник. Логотип с внутренней стороны находится в начальной стадии и не отображает блики. Конструкция (и её стороны) в объёмном виде до её фактического преобразования будет выглядеть так



Все стороны конструкции в соответствии с распределёнными бликами:

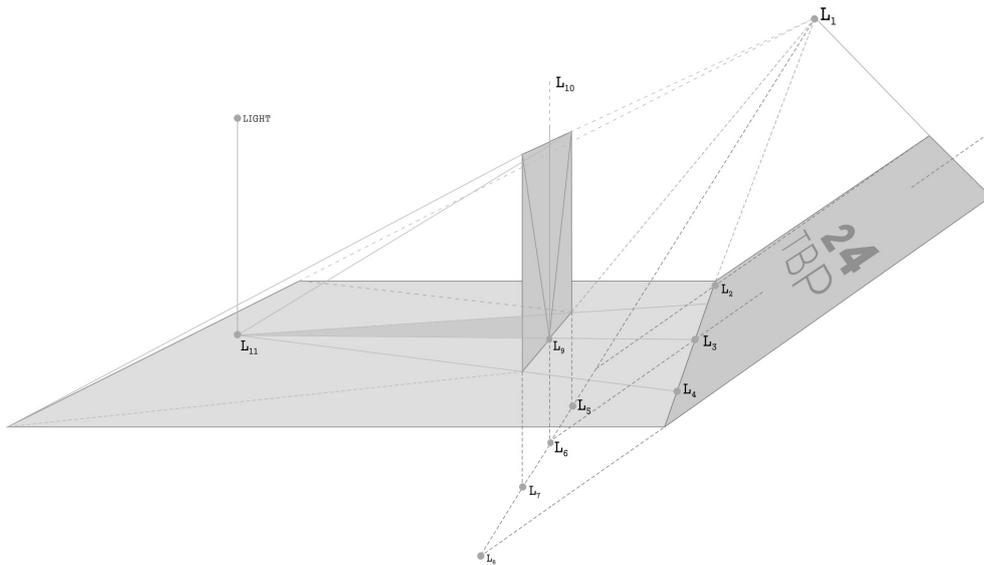
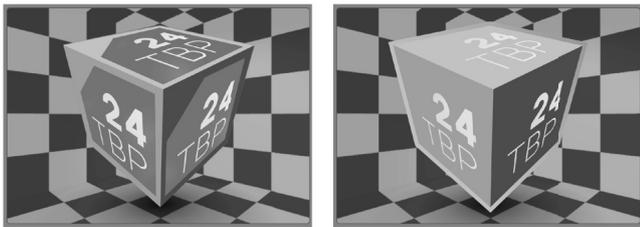
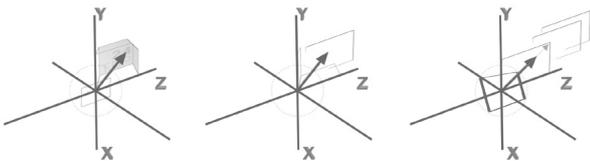


Рис. 12. Фрагмент геометрической конструкции (начальная стадия распределения световых потоков), иллюстрирующий внутреннее отражение перспективы (ER).



В перспективах распределения бликов участвуют разные слои, слои динамически зависят друг от друга и могут отражаться от разных узловых точек



Координатные конструкции показывают откат удалённой перспективы для подгонки конструкции к разным программно-аппаратным решениям. Особенно, когда речь заходит об экспорте (сохранении) спроектированной конструкции и вновь её воспроизведении (прорисовки) в другой системе трёхмерного моделирования. Ниже представлен развёрнутый вид конструкции, показывающий распределение световых потоков, образующих внешнее отражение перспективы

Рисунок 12 схематично иллюстрирует распределение световых потоков внешнего отражения перспективы. Фронтальная часть логотип «ТВР24» покрывается

бликами, образуемыми в результате внешнего отражения перспективы, и постепенно логотип приобретает блеск. Процесс отражения и отражательная способность искусственных лучей финальной замыкающей стадии представлена на рисунке 13.

Из рисунка 13 следует, что после распределения световых потоков, на начальной стадии образуются дублирующие тени, которые делают световые потоки более объёмными. Такой эффект необходим для плавной огранки объекта, её результаты отражаются на внешних сторонах объекта и выглядят как плавный обтравочный контур, который может преобразовываться благодаря экструзии. Пример такого объёмного преобразования и панорамирования можно наблюдать ниже.



В качестве примера двумерной трансформации и распределения света можно продемонстрировать титр с логотипом канала:

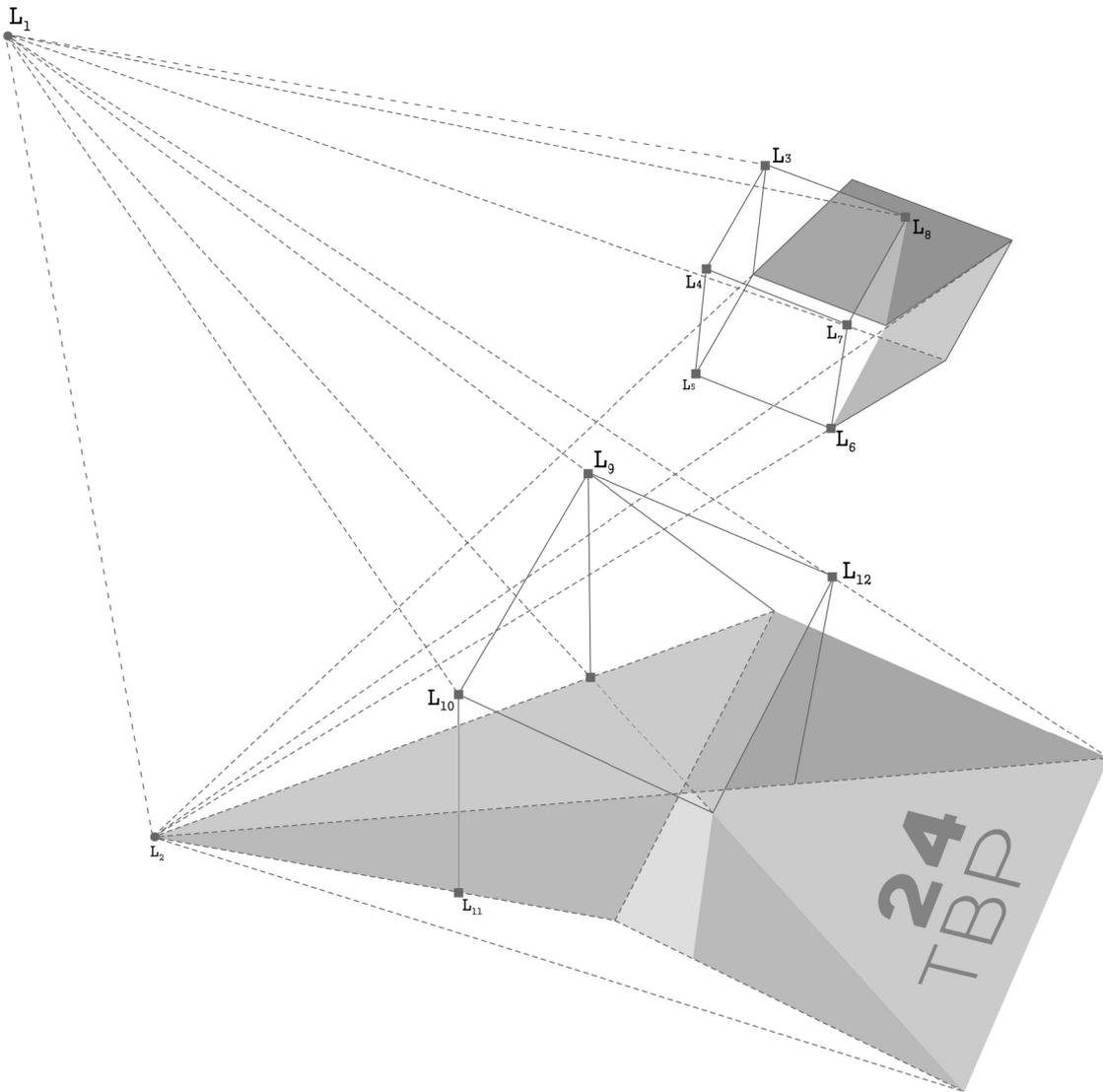
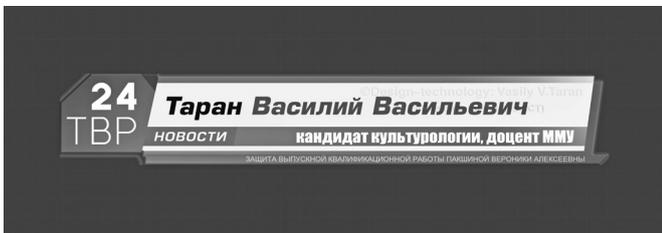


Рис. 13. Фрагмент геометрической конструкции (начальная стадия распределения световых потоков), иллюстрирующий внешнее отражение перспективы (IR).



С внешней стороны конечная стадия отражения перспектив будет выглядеть следующим образом

Рисунок 14 показывает внешнее отражение перспективы на конечной стадии распределения световых потоков. Из рисунка хорошо видно, что после описываемых выше процедур и технических формулировок логотип

«ТВР24» стал обёрнут бликами. И этот эффект в таком варианте останется вместе с фигурой в не зависимости от используемых каналом программно-аппаратных решений. Нижеследующая иллюстрация визуализирует схему (рисунок 15) конечной стадии распределения световых потоков, так как это будет представлено в кадре



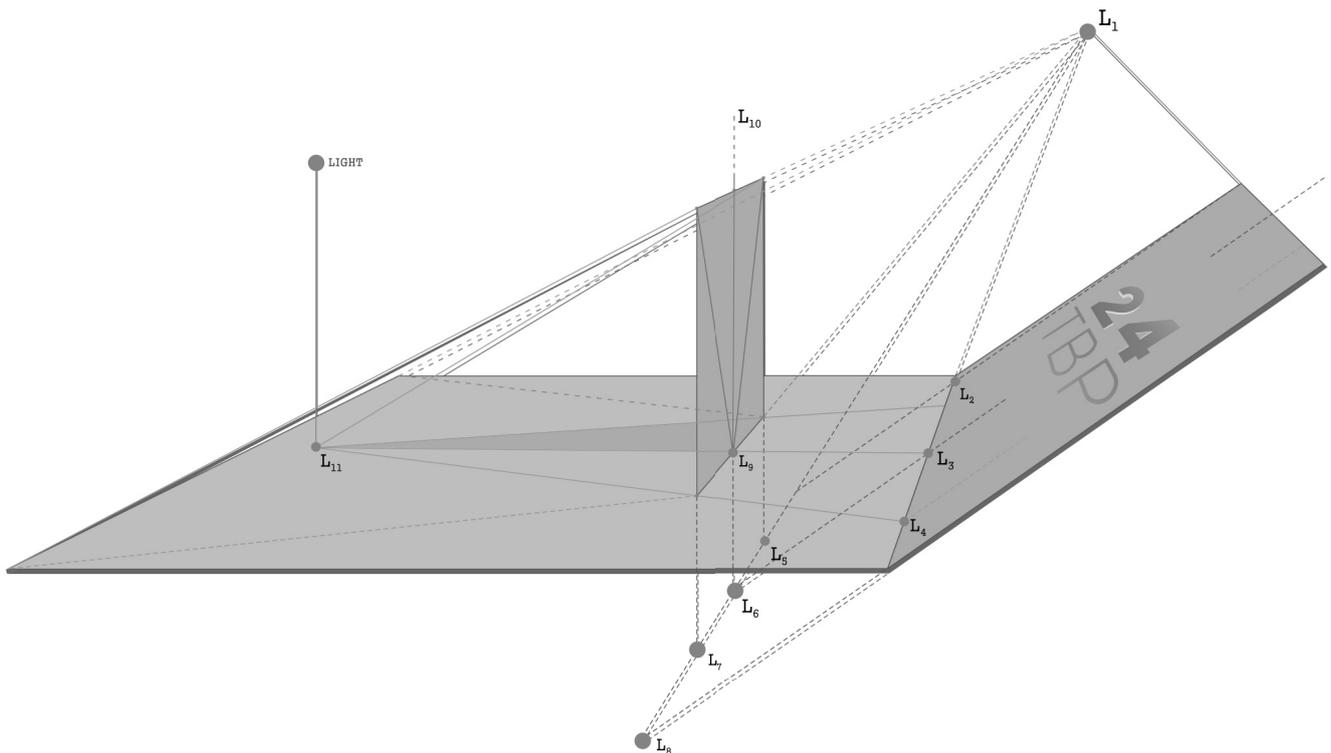
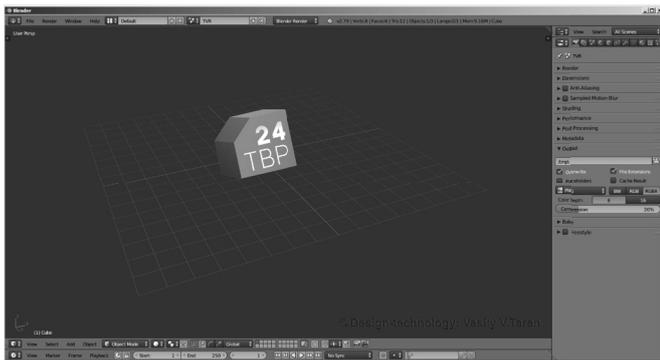


Рис. 14. Фрагмент геометрической конструкции (конечная стадия распределения световых потоков), иллюстрирующий внутреннее отражение перспективы (ER).

После необходимых манипуляций, принцип которых изложен нами выше (настройка цвета и коррекция отражательной способности объектов в кадре), можно посмотреть на результаты:



Рендеринг готовых объектов, пригодных для оформления кадра (логотип объёмный), прорисовка из шести разновидностей неправильного пятиугольника. Периоды трансформации световых потоков на геометрии объекта (в разложенном виде) иллюстрируют рисунки 16,17. При выполнении анимации разметка кадра может быть как 16:9, так и 4:3. Рисунки 18,19 иллюстрируют возможную разметку кадра для плашечных двухмерных и трёхмерных конструкций. В соответствии с данной

разметкой (А, В, С, В — поля кадра) можно использовать отрисованные двухмерные плашечные конструкции, показанные в приложении № 1

Промежуточная обрисовка светом конструкций, демонстрирующая мягкую обтавку граней световым потоком в условиях прозрачности (рис. 20).

Возможная карта настройки редактора узлов применительно к цветам проекта (рис. 21).

Логотип в движении, результат рендеринга промежуточной анимации в среде компьютерного моделирования Blender®



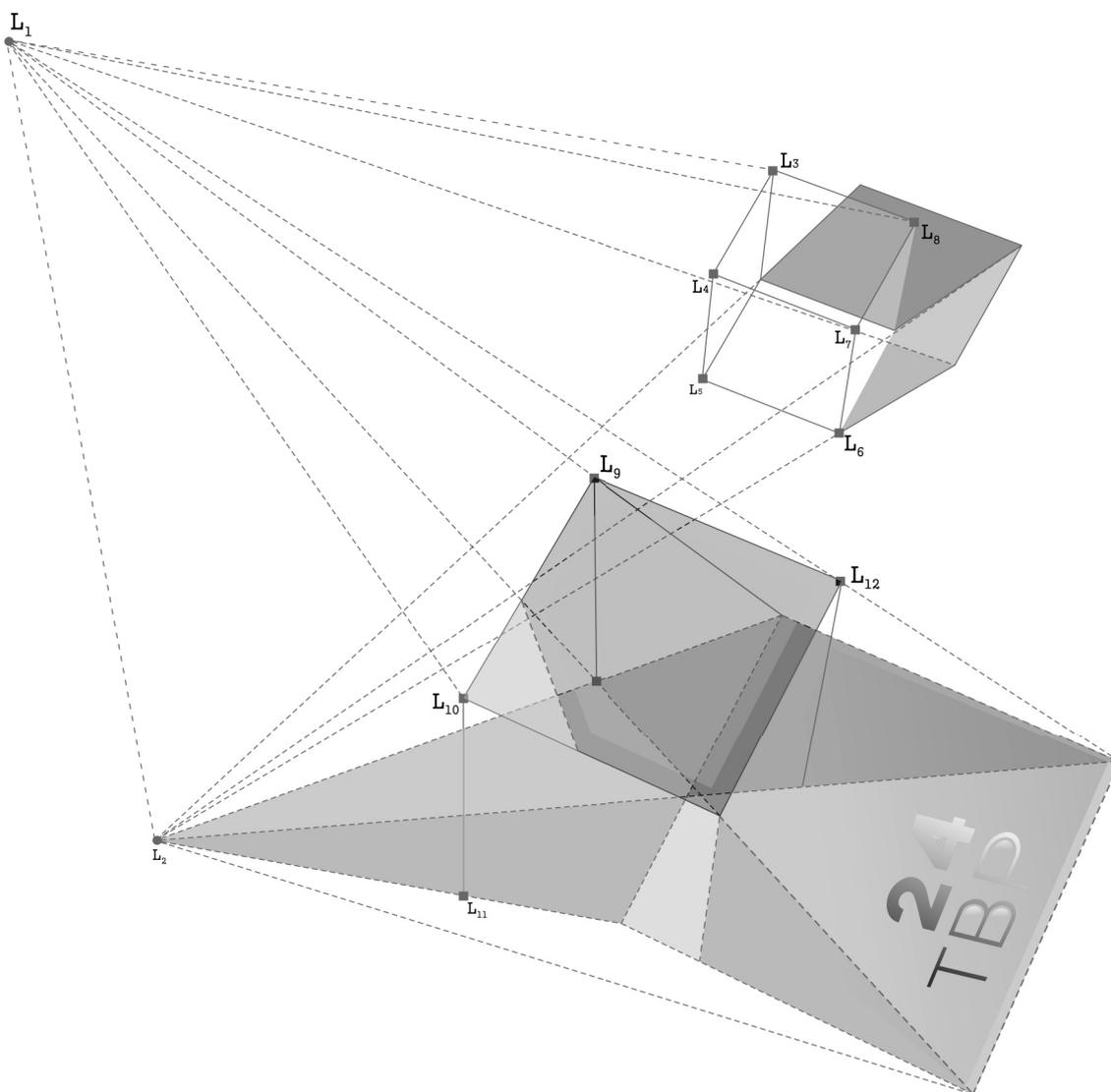


Рис. 15. Фрагмент геометрической конструкции (конечная стадия распределения световых потоков), иллюстрирующий внешнее отражение перспективы (IR).

Логотип статический, объёмная проекция, может использоваться как текстурная подложка в заставке (см. приложение 2)



Проекция для дополнительной ротации с фронтальным изменением неправильного пятиугольника и двухмерная конструкция (лицевая часть логотипа)



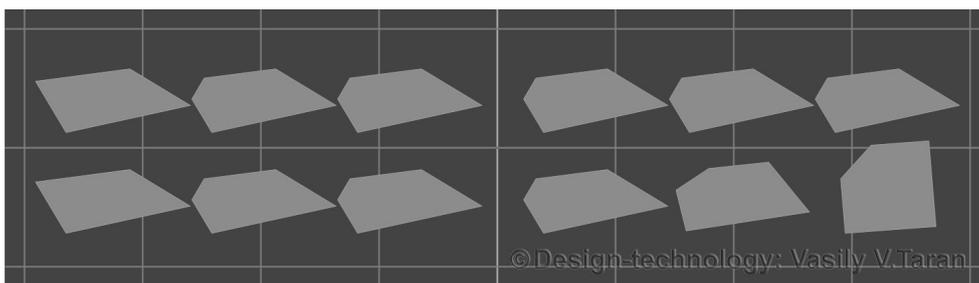


Рис. 16. Трансформация геометрии объекта при одноцелевом потоке.

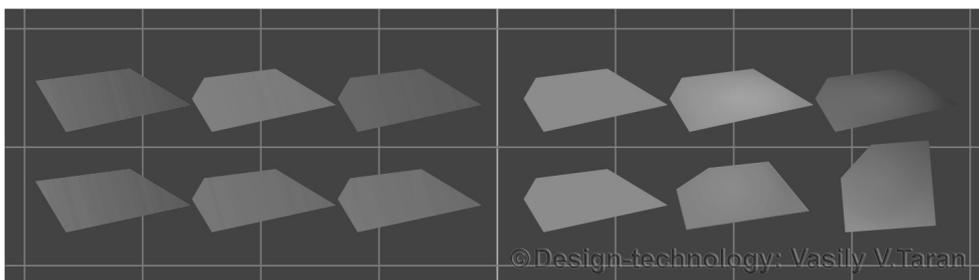


Рис. 17. Трансформация геометрии объекта при многоцелевых потоках.

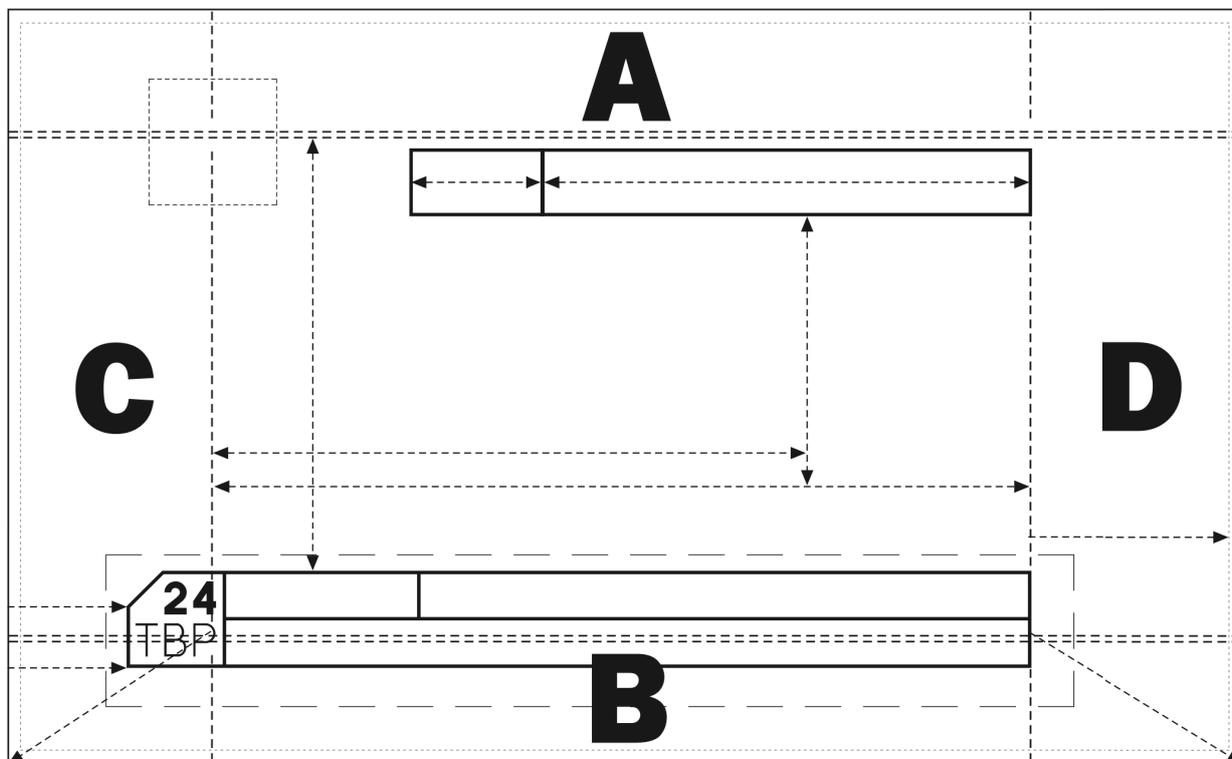


Рис. 18. Схема отображения двухмерных и трёхмерных графических данных с пропорциями 16:9.

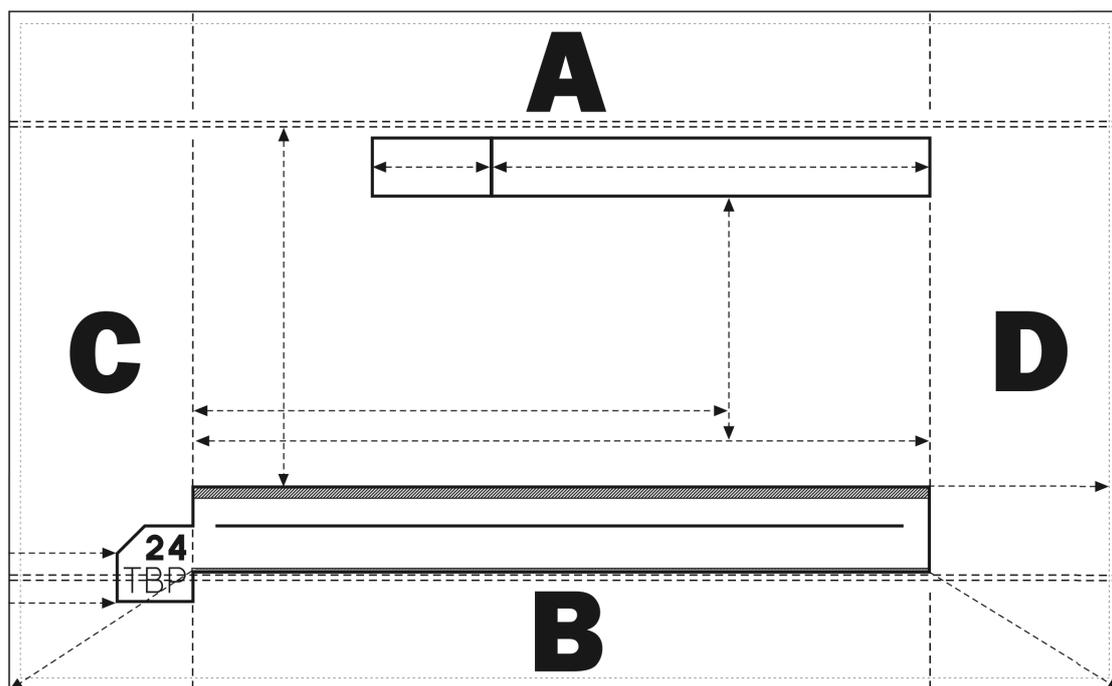


Рис. 19. Схема отображения двухмерных и трёхмерных графических данных с пропорциями 4:3.

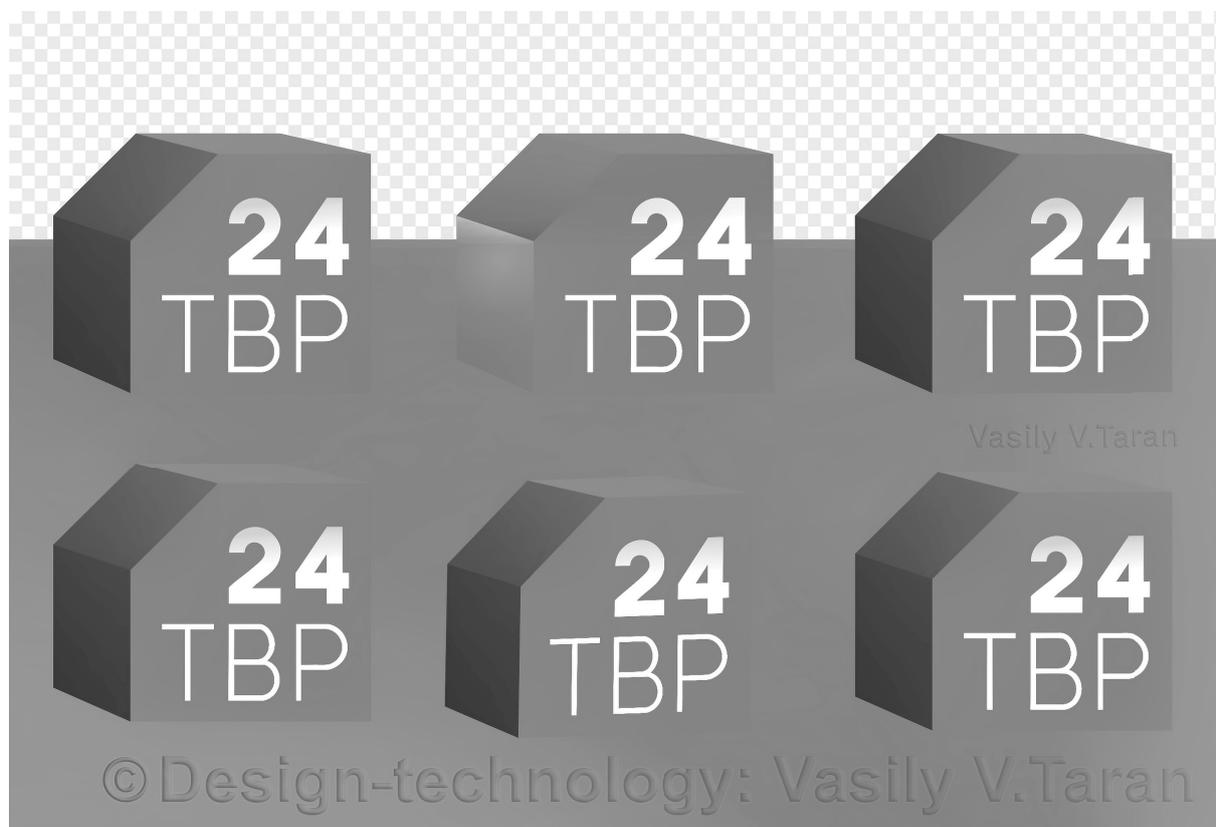


Рис. 20

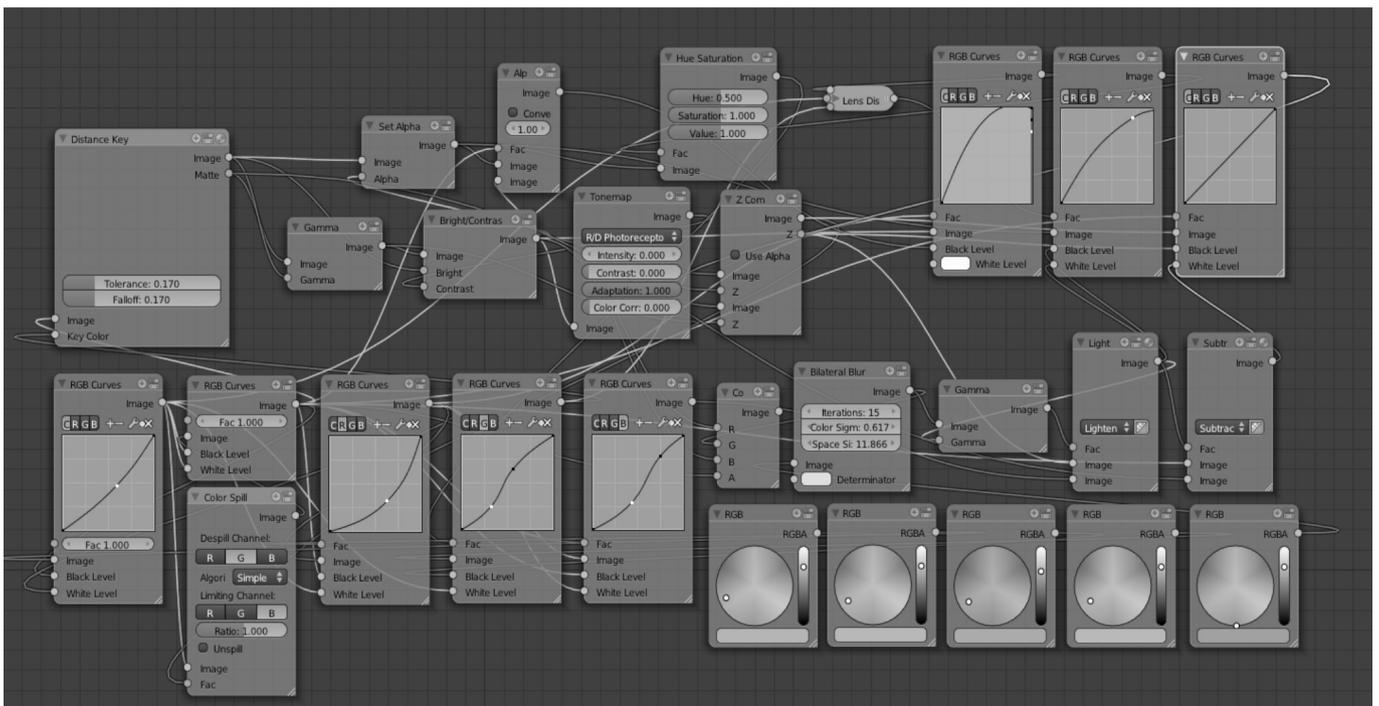


Рис. 21



Рис. 22

Проекция с объёмными цифрами (полученная методом световой экструзии) может использоваться для новостных заставок, и передач новостной тематики (рис. 22).

Проекция с усечёнными углами и плавными переходами, для межпрограммных вставок рекламы, а также объёмная проекция для фронтальных заставок



Возможность применения рассмотренного нами метода в области проектирования двумерных иллюстраций (рис. 23).

Метод может успешно применяться и в двумерном моделировании для исправления различных компонентов композиции, ниже акцентированы области изменения композиции (рис. 24).

Для демонстрации статичных форм смоделированных изображений (устанавливаем уровень их компрессии) и экспортируем их в обычный JPEG-файл. Обращение для Python™ будет следующим

```
RenderSettings.filepath
bpy.data.science[«Scene»].render.filepath
```

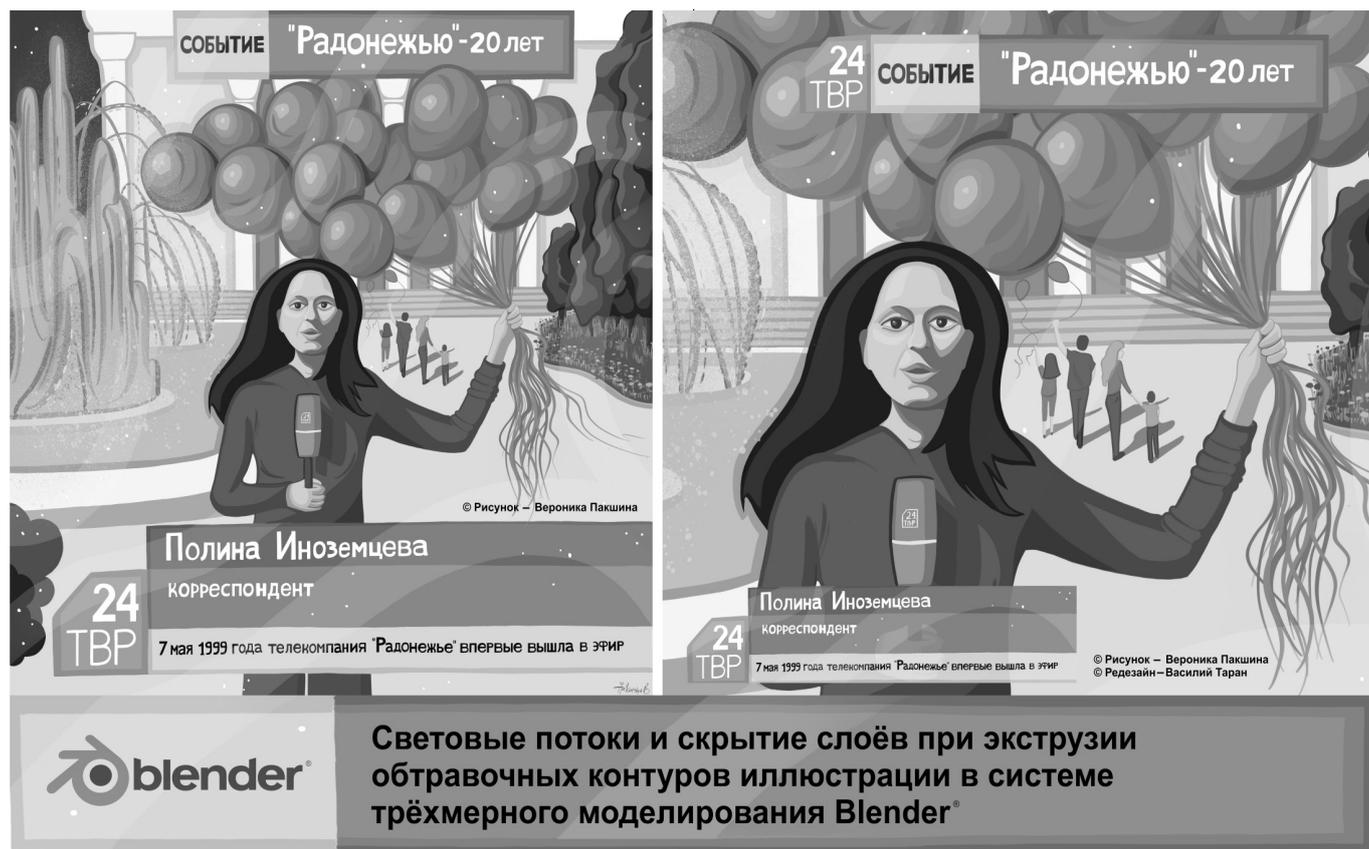


Рис. 23



Рис. 24

При экспорте изображения следует учесть степень прорисовки отражённых отскоков, визуально они идентифицируются как глянцевые эффекты. Степень отскоков должна составлять 300%. Регулирующие опции языка Python™ здесь следующие

```
CyclesRenderSettings.glossy_bounces
 bpy.data.scenes["Scene"].cycles.glossy_bounces
```

При фиксированных позициях камеры (формула 26).

Степень отскоков для экспорта статичного трёхмерного изображения очень важна, потому как она влияет на точное перераспределение световых потоков, значения ниже 300% могут не учитывать некоторые потоки, тем самым скрытые слои будут присутствовать, не выполняя никакой функции, а только увеличивая объём

$$xvec = \text{Vector}(1,0,0) * \text{camera\_matrix}$$

$$yvec = \text{Vector}(0,1,0) * \text{camera\_matrix}$$

0,1,2,3,4,5/\*0\*

additional values

$$\text{render\_cam\_ob.LocZ} = 1.0 \text{ [0.0][z]}$$

$$\text{render\_cam\_ob.LocX} = 0.5 \text{ [0.0][x]}$$

$$\text{render\_cam\_ob.LocY} = 0.5 \text{ [0.0][y]}$$

Формула 26

экспортируемого файла. В вопросах регулировки качества изображения и его степени сжатия может пригодиться следующее обращение на Python™

```
ImageFormatSettings.quality
 bpy.data.scenes ["Scene"]. render.image_settings.quality
```

Таким образом, мы рассмотрели прикладную часть вопросов, связанных с созданием и моделированием геометрической конструкции (логотипа) в трёхмерной среде компьютерного моделирования Blender®, пригодных для оформления потокового телевизионного вещания телерадиокомпании «Радонежье».

Центральным вопросом научной статьи (в аспекте разработки геометрической конструкции) послужила возможность точного воспроизведения (прорисовки) спроектированного логотипа в других компьютерных редакторах трёхмерного моделирования, композитинга, программах логогенерации и компьютерного монтажа видеоматериала, а также прочих утилит, задействованных в представлении титровальной графики вне зависимости от файла хранения трёхмерных данных. Создать такую конструкцию удалось благодаря описанному в статье методу перераспределения световых потоков и скрытию слоёв при экструзии обтравочных контуров конструкции. В основе метода лежит принцип разграничения слоёв, отражающих световые потоки, в последствии образующие обтравочный контур, который может быть трансформирован. Конечно это полностью не снимает общей проблемы совместимости и качественной визуализации компьютерных форматов хранения трёхмерных данных. Вопрос является открытым и носит дискуссионный характер, однако изложенная в статье практика может оказаться полезной для инженеров-проектировщиков и прочих технических специалистов, чья область деятельности затрагивает компьютерное моделирование. Метод является универсальным и программно независим, что позволяет успешно применять его и при других условиях трёхмерной визуализации данных, таких как статистика, проектирование сложных компьютерных

графических интерфейсов, составление динамичной инфографики.

Некоторые связи программы Blender® с языком Python™ (реализуемые через Python™ Interactive Console и текстовый редактор подсветки и редактирования кода) позволят достигнуть лучших результатов, касающихся точности моделирования, данный язык программирования хорошо вписывается в программно-интерфейсную основу и позволяет разрешать различные утилитарные задачи с высокой точностью. К тому же на компьютерах, которые в силу своей бюджетной аппаратной комплектации не могут конкурировать с рабочими станциями премиум класса, язык программирования Python™ пригодится как вспомогательный инструмент по визуализации сложных трёхмерных данных.

Все разработанные трёхмерные конструкции могут быть выведены путём прямого экспорта из программы, либо с применением специальных систем рендеринга, авторы рекомендуют использовать LuxRender.

Авторы выражают благодарность редакции и всем сотрудникам телерадиокомпании «Радонежье» за согласие выступить в качестве экспериментальной базы для научно-технических разработок и применения новых методов визуализации в компьютерной графике и за оказанное доверие по трёхмерной визуализации логотипа, а также некоторых элементов фирменного стиля компании.

Авторы надеются, что их наработки в области компьютерной графики, дизайна и трёхмерного моделирования найдут применение в прикладной практике оформления и визуализации новых элементов фирменного стиля телерадиокомпании «Радонежье». Внедрение данных наработок в техническую деятельность телерадиокомпании повысят точность и эффективность подготовки трёхмерного иллюстративного материала для оформления потокового вещания, межпрограммных блоков и отдельных видеосюжетов.

Приложение № 1

**24** **НОВОСТИ** **ПАКШИНА ВЕРОНИКА АЛЕКСЕЕВНА**  
ТВР **ТВР «РАДОНЕЖЬЕ»**

**НОВОСТИ** **ТАРАН ВАСИЛИЙ ВАСИЛЬЕВИЧ**  
**НАУЧНЫЙ РУКОВОДИТЕЛЬ, КАНДИДАТ КУЛЬТУРОЛОГИИ**

**24** **НОВОСТИ** **ТВР «РАДОНЕЖЬЕ»**  
ТВР ЗАЩИТА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ ПАКШИНОЙ ВЕРОНИКИ АЛЕКСЕЕВНЫ

**24** **ТАРАН ВАСИЛИЙ ВАСИЛЬЕВИЧ**  
ТВР **НАУЧНЫЙ РУКОВОДИТЕЛЬ, КАНДИДАТ КУЛЬТУРОЛОГИИ**

**24** **ТАРАН ВАСИЛИЙ ВАСИЛЬЕВИЧ**  
ТВР **НАУЧНЫЙ РУКОВОДИТЕЛЬ, КАНДИДАТ КУЛЬТУРОЛОГИИ**

**24** **ТАРАН ВАСИЛИЙ ВАСИЛЬЕВИЧ**  
**НАУЧНЫЙ РУКОВОДИТЕЛЬ, КАНДИДАТ КУЛЬТУРОЛОГИИ**  
**24** **ТВР** ЗАЩИТА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ ПАКШИНОЙ ВЕРОНИКИ АЛЕКСЕЕВНЫ

**24** **Пакшина Вероника**  
ТВР **НОВОСТИ** **СОТРУДНИК ТВР «РАДОНЕЖЬЕ»**  
ЗАЩИТА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ ПАКШИНОЙ ВЕРОНИКИ АЛЕКСЕЕВНЫ

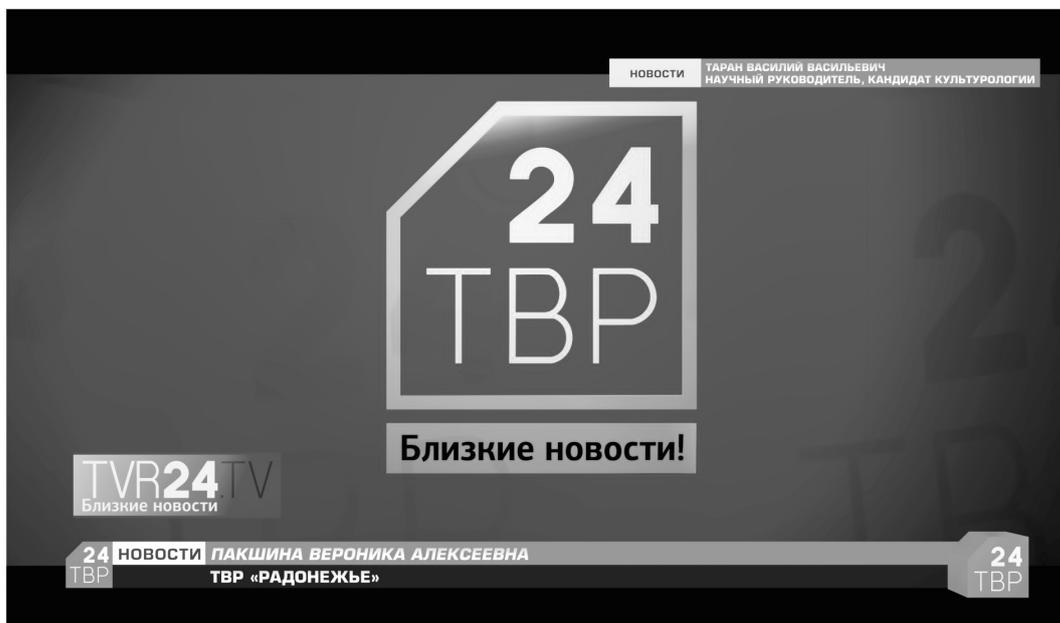
**24** **Таран Василий Васильевич**  
ТВР **кандидат культурологии, доцент ММУ**

**TVR24** TV  
Ближкие новости

**TVR24** TV  
Ближкие новости

**TVR24** TV  
Ближкие новости

Приложение № 2



ЛИТЕРАТУРА

1. Филиппов С. В. Программная платформа Blender как среда моделирования объектов и процессов естественно-научных дисциплин // Препринты ИПМ им. М. В. Келдыша. 2018, № 230, 42 с. (DOI:10.20948/prepr-2018-230).
2. Флоринский И.В., Филиппов С. В. Трёхмерное моделирование рельефа: применение пакета Blender // ИнтерКарто / ИнтерГИС 24, Материалы Международной конференции, Петрозаводск, Бонн, Анкоридж, 19 июля — 1 августа 2018, Ч. 2. Петрозаводск: КарНЦ РАН, 2018, с. 250–261.
3. Zhi Hao Zuo, Yi Min Xie A simple and compact Python code for complex 3D topology optimization//Advances in Engineering Software 85 (2015) p.p. 1–11. (<http://dx.doi.org/10.1016/j.advengsoft.2015.02.006>)
4. Chris Conlan The Blender Python API: Precision 3D Modeling and Add-on Development/ Bethesda, Maryland USA, 138 p. [ISBN-13 (pbk): 978–1–4842–2801–2], [ISBN-13 (electronic): 978–1–4842–2802–9] (DOI: 10.1007/978–1–4842–2802–9).
5. Allan Brito Blender 3D2.49 Incredible Machines Modeling, rendering, and animating realistic machines with Blender 3D Published by Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK, 316 p. [ISBN978–1–847197–46–7 2009].
6. Gordon C. Fisher Blender 3D Basics Beginner's Guide (The complete novice's guide to 3D modeling and animation) Published by Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 468 p. [ISBN978–1–84951–690–7].
7. Sunil Kapil Clean Python: Elegant Coding in Python / Sunnyvale, CA, USA [ISBN-13 (pbk): 978–1–4842–4877–5], [ISBN-13 (electronic): 978–1–4842–4878–2] (DOI: <https://doi.org/10.1007/978–1–4842–4878–2>).

© Таран Василий Васильевич ( [allscience@lenta.ru](mailto:allscience@lenta.ru) ), Пакшина Вероника Алексеевна ( [pakshina-veronika@mail.ru](mailto:pakshina-veronika@mail.ru) ).

Журнал «Современная наука: актуальные проблемы теории и практики»



Московский международный университет