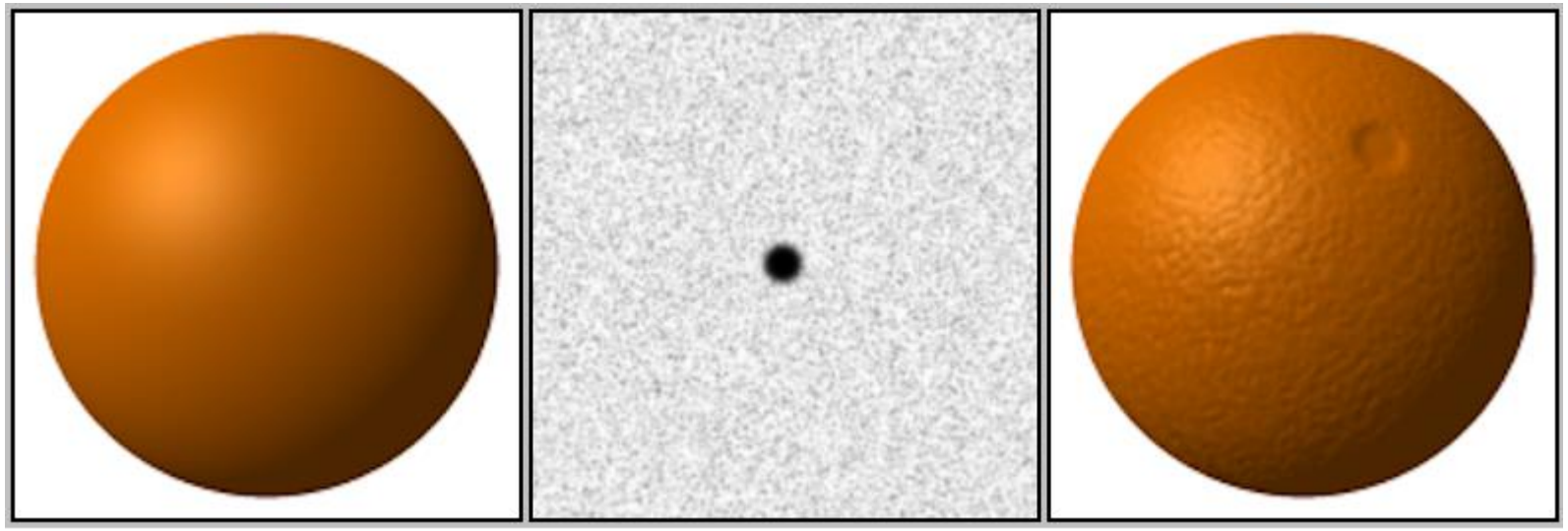
# Компьютерная графика
## Рельефное текстурирование

Лекция 5

Демяненко Я.М. ЮФУ 2023 MAGnUS

# Рельефное текстурирование — что это?



Рельефное текстурирование — метод в компьютерной графике для придания более реалистичного и насыщенного вида поверхности объектов

# В чем отличие от обычного текстурирования?

Рельефное текстурирование отражает реальное положение источника света в сцене и даже изменение его местоположения

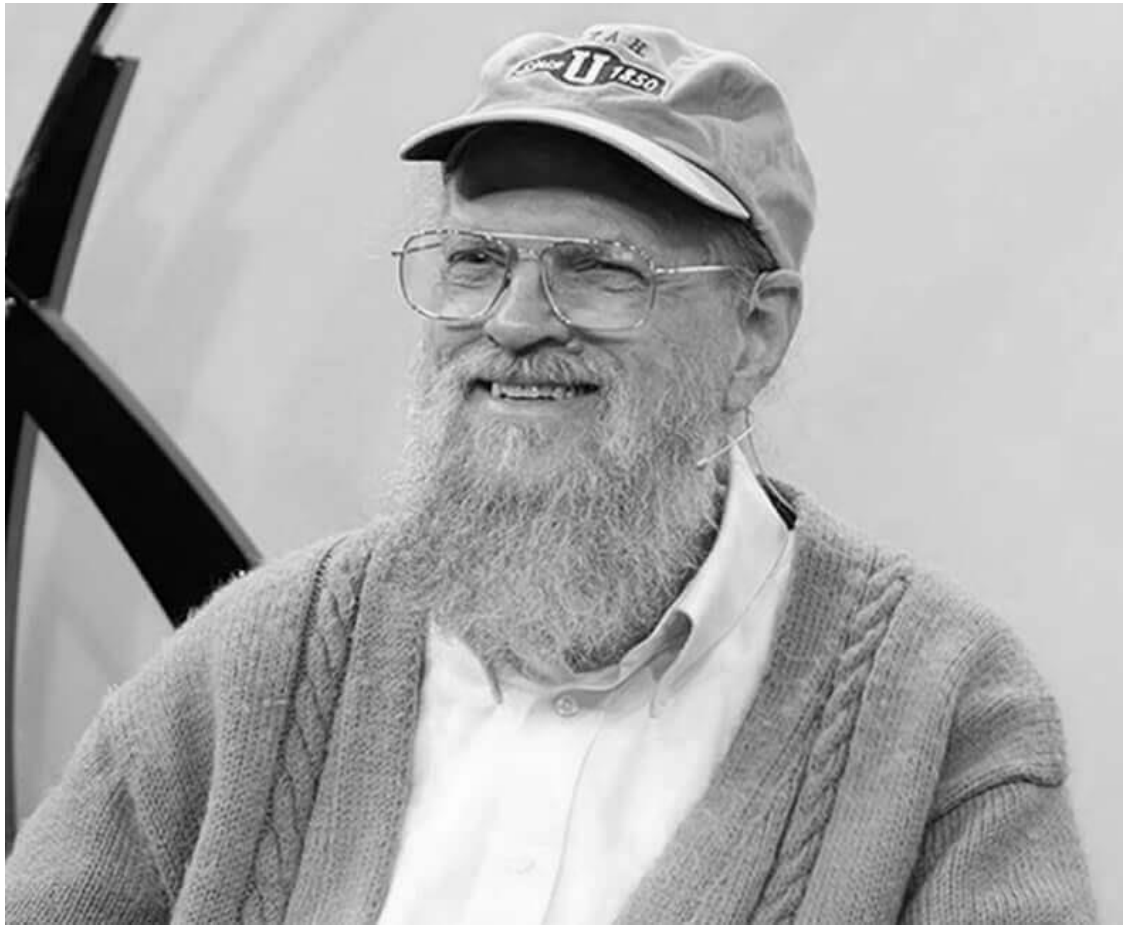
# Виды рельефного текстурирования

- Bump mapping
- Normal mapping
- Displacement mapping
- Parallax mapping
- Parallax occlusion mapping

# Немного из истории Bump mapping

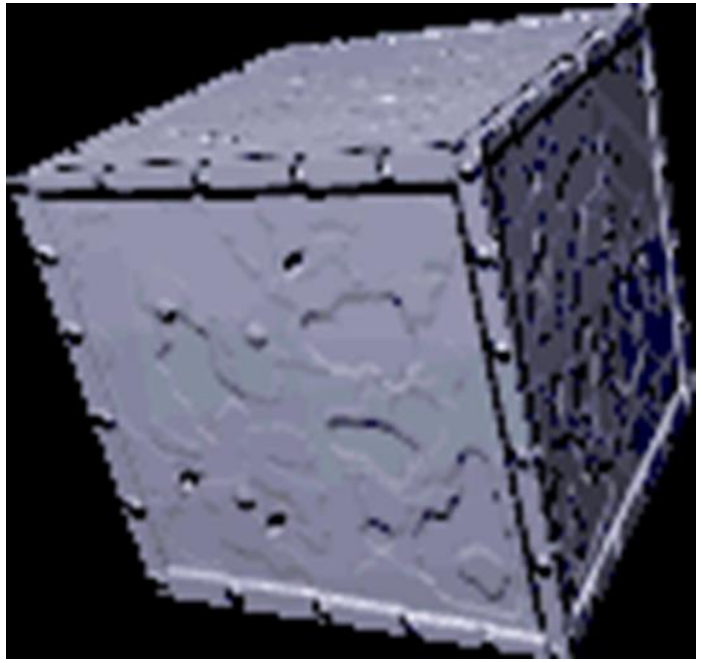

Бампмаппинг был разработан Блинном (Blinn) еще в 1978 году

# Jim Blinn

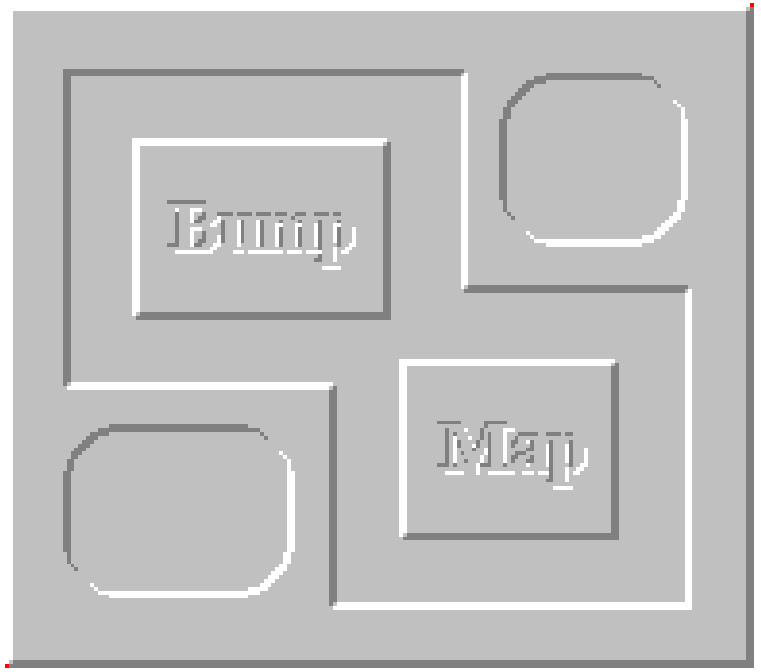


Born: 1949
Alma mater: University of Utah; University of Michigan
Awards: Macarthur fellowship (1991); NASA Exceptional Service Medal
Fields: Computer science
Institutions: NASA's Jet Propulsion Laboratory; Microsoft Research

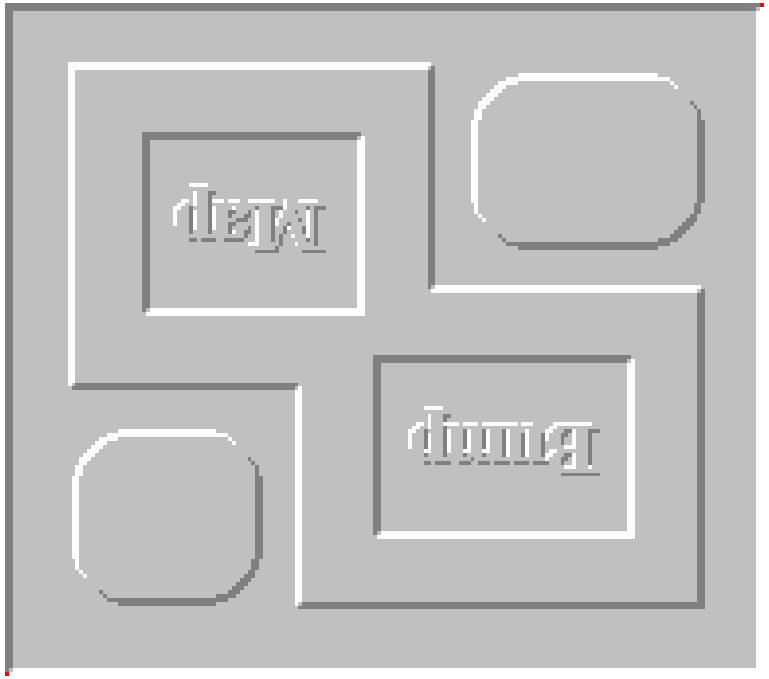
# Просто кубик



# Где выпуклости?



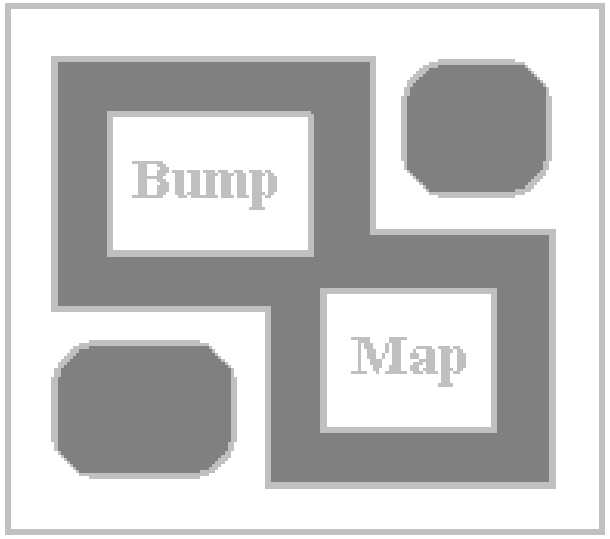
# А если так?



# Что такое рельефная карта?

Рельефная карта (текстура) — это обычная **текстура**, только **в отличие** от первой, несущей информацию о **цвете** определенных участков,
**рельефная карта** несет информацию о **неровностях**

# Карта высот



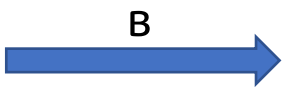
# Что имитировать

- Можно неровности и шероховатости
  - дерево
  - камень
  - шелушащуюся краску
  - и т.д.
- Нельзя крупные впадины и возвышенности

# Что нужно?

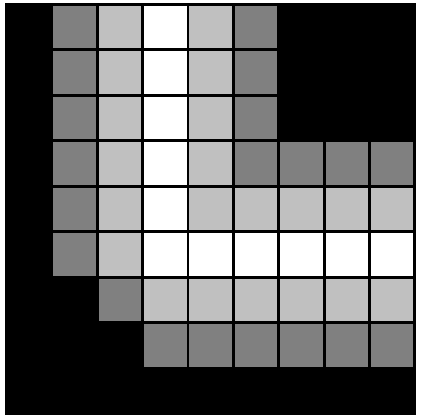
Преобразовать

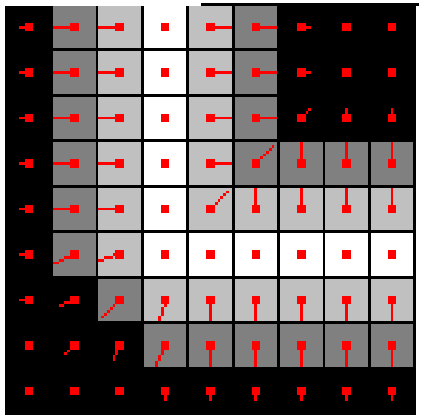
информацию о высоте неровностей на карте высот

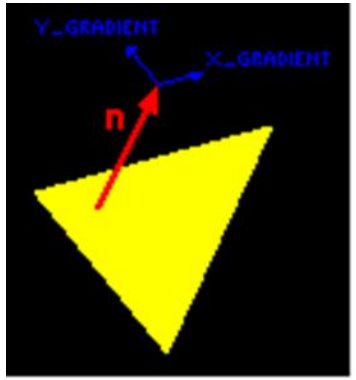


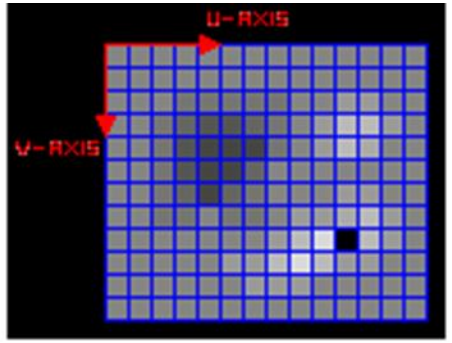

информацию о величине подстройки вектора нормали

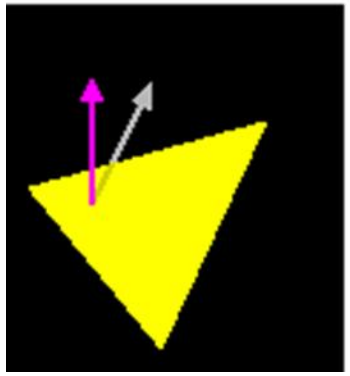
# Как реализовать?

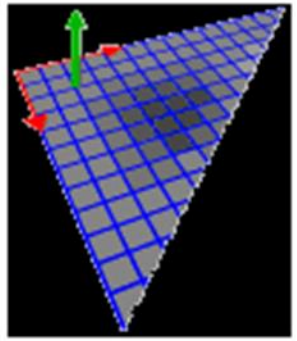






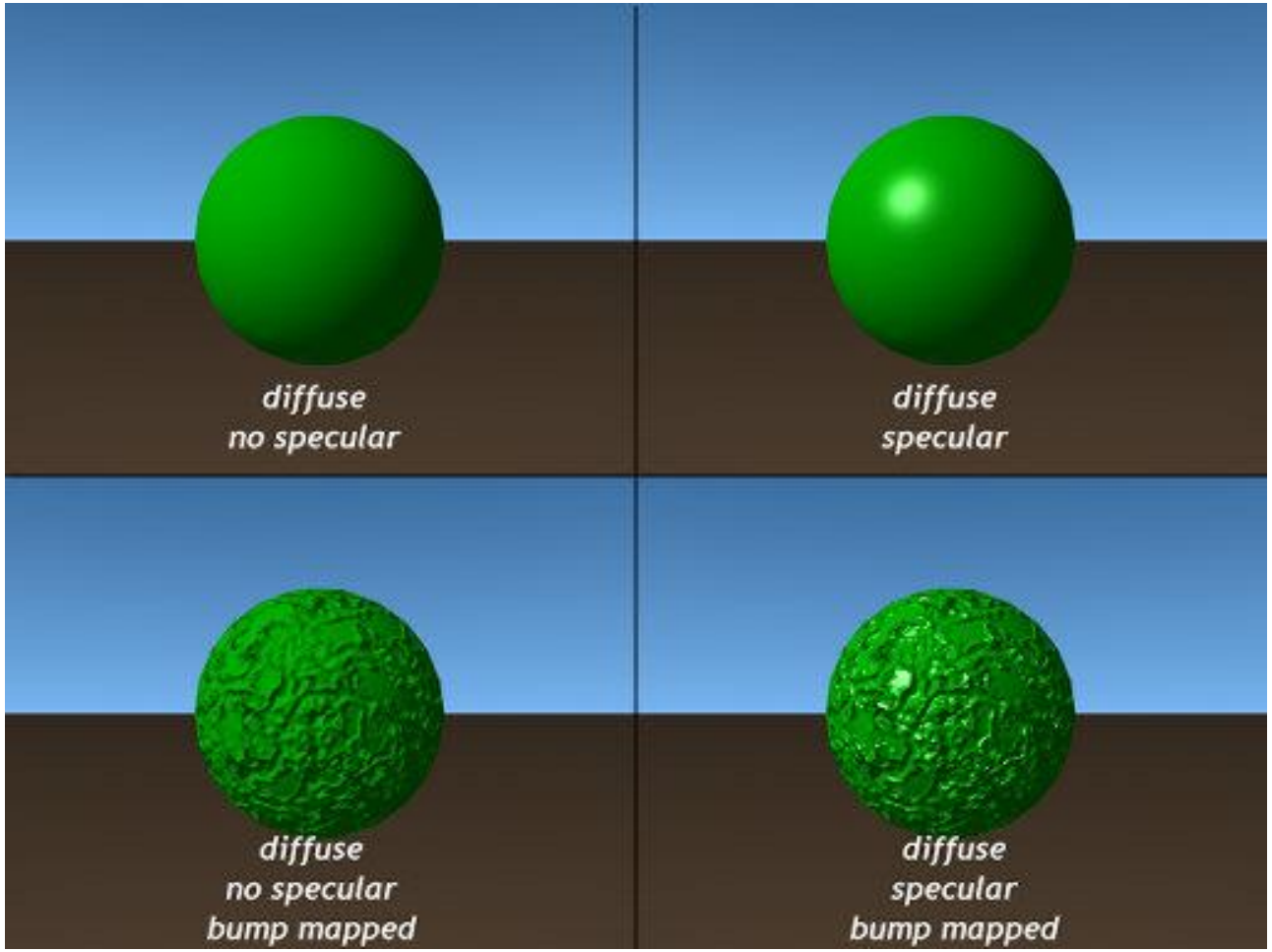





x_gradient = pixel(x-1, y) — pixel(x+1, y)
y_gradient = pixel(x, y-1) — pixel(x, y+1)

New_Normal=Normal+(U*x_gradient)+(V*y_gradient)

# Bump mapping



# Bump mapping в играх

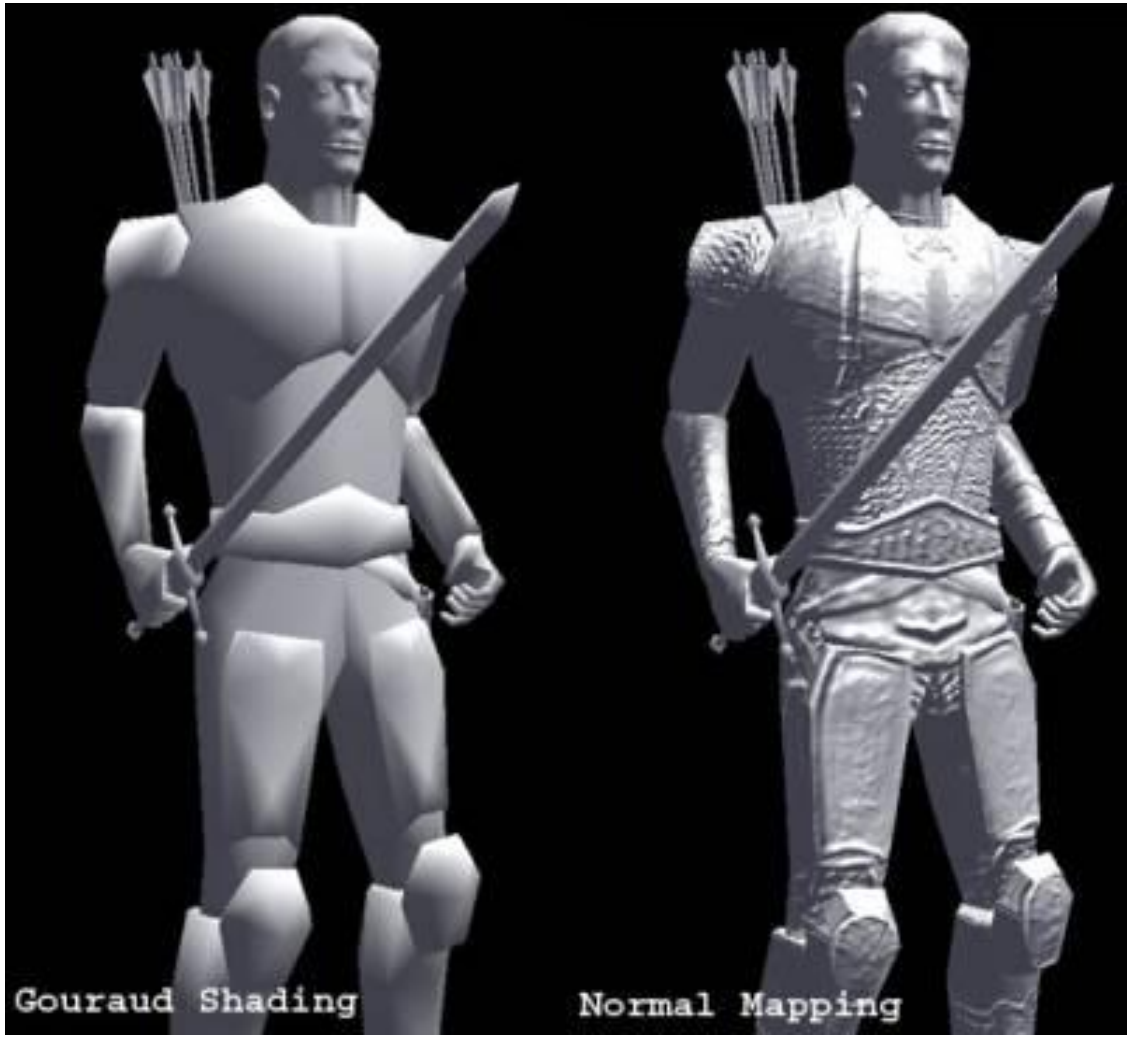
рендеринг без бампмаппинга

рендеринг с бампмаппингом
без бликовой составляющей

# Карта нормалей (normal map)

- Это дальнейшее развитие технологии бамп (bump)
- Иногда карты нормалей называют «3-х точечная бамп-карта» (Dot3 bump mapping)

# Normal Mapping



# Немного из истории

- Впервые идея о том, что можно «брать» детали с высокополигональной модели и переносить на низкополигональную, была предложена на SIGGRAPH в 1996 во время доклада "Использование сглаженных поверхностей для добавления плотности полигональной сетке" авторов Кришнамурати и Левои

- SIGGRAPH (The Special Interest Group for Computer Graphics)

- Тогда речь шла об использовании для Displacement Mapping

- Предлагалось использовать созданные на основе NURBS карты смещения (displacement maps) для увеличения детализации низкополигональной поверхности

- Реализация появилась гораздо позже

# И ещё из истории

- В 1998 году опять на SIGGRAPH были представлены 2 доклада о перенесении деталей в виде карт нормалей от высокополигональных моделей в низкополигональные

- Один из них: «Appearance Preserving Simplification». Авторы: Jonathan Cohen, Marc Olano, Dinesh Manocha

- www.graphicon.ru/oldgr/library/siggraph/98/papers/cohen/cohen.pdf

# Из статьи Appearance Preserving Simplification

Figure 1: Bumpy Torus Model. *Left*: 44,252 triangles full resolution mesh. *Middle and Right*: 5,531 triangles, 0.25 mm maximum image deviation. *Middle*: per-vertex normals. *Right*: normal maps

# Appearance Preserving Simplification

| 249,924 triangles | 62,480 triangles | 7,809 triangles | 975 triangles |
|---|---|---|---|
| 0.05 mm max image deviation | 0.25 mm max image deviation | 1.3 mm max image deviation | 6.6 mm max image deviation |

Figure 12: Close-up of several levels of detail of the armadillo model. *Top*: normal maps *Bottom*: per-vertex normals

# Общее: bump и normal -mapping

- Увеличивает визуальную детализацию 3d-модели без добавления дополнительных полигонов
- Карта рассчитывается для каждого пикселя текстуры

# Различия: bump и normal -mapping

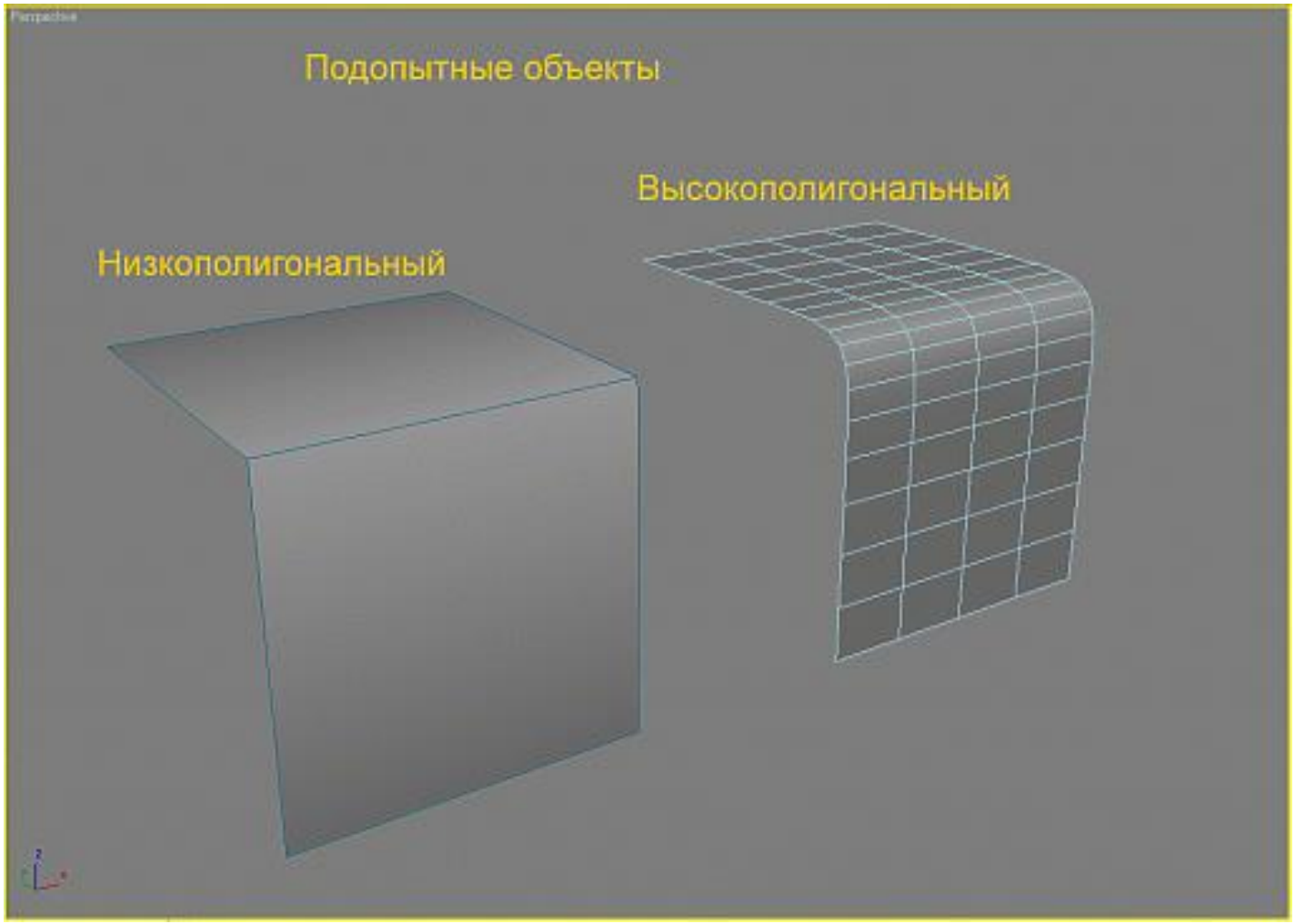
- Бамп рассчитывается на основе текстуры с одним каналом (grayscale — оттенки серого)

- Карты нормалей используют многоканальную RGB-текстуру, поэтому метод дает большую точность, чем Bump mapping

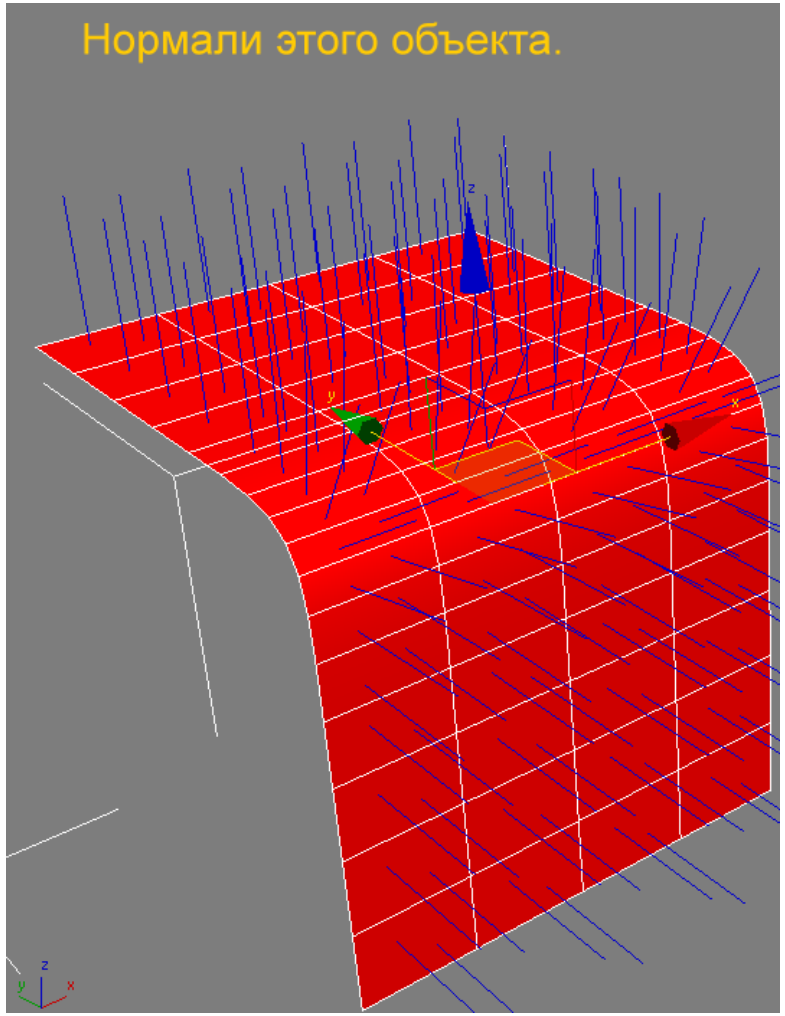
Bump Map

Normal Map

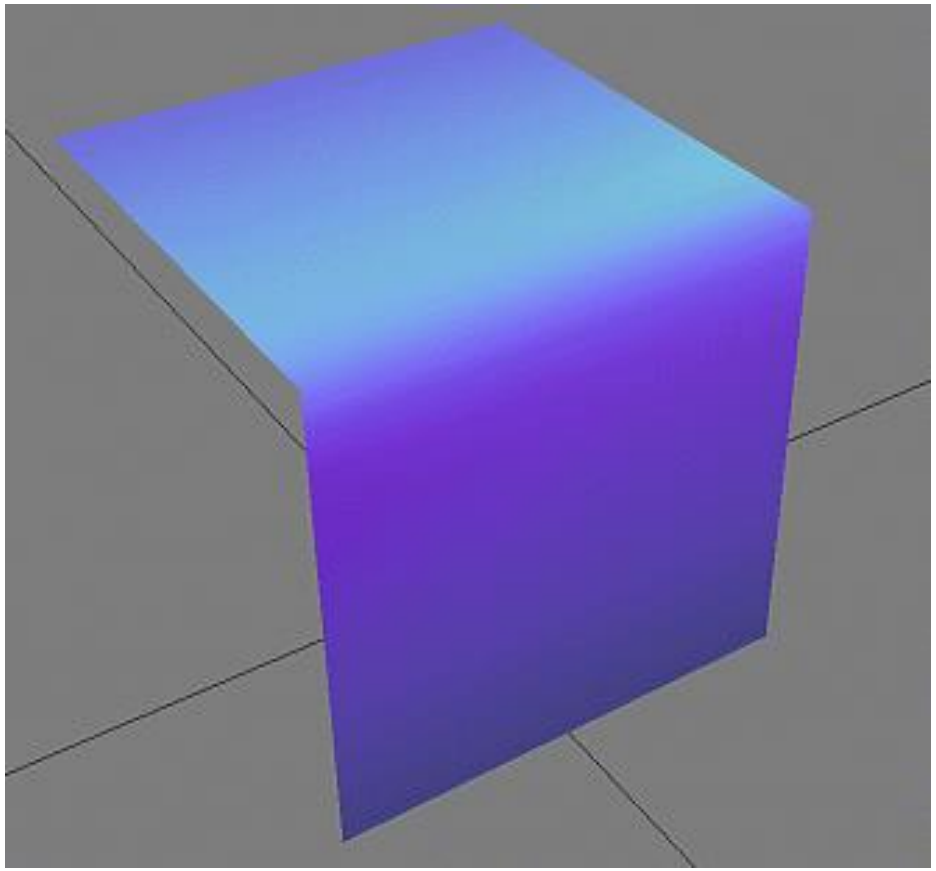
# Объекты


Подопытные объекты
Высокополигональный
Низкополигональный

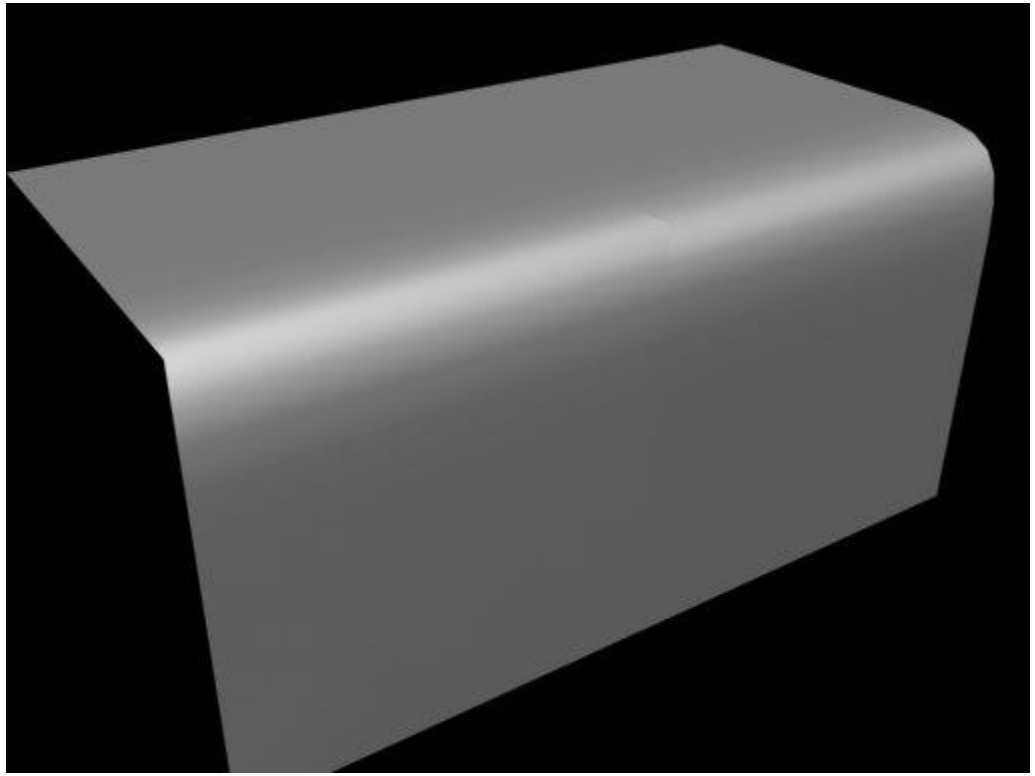
# Нормали высокополигонального объекта



# Карта нормалей как текстура

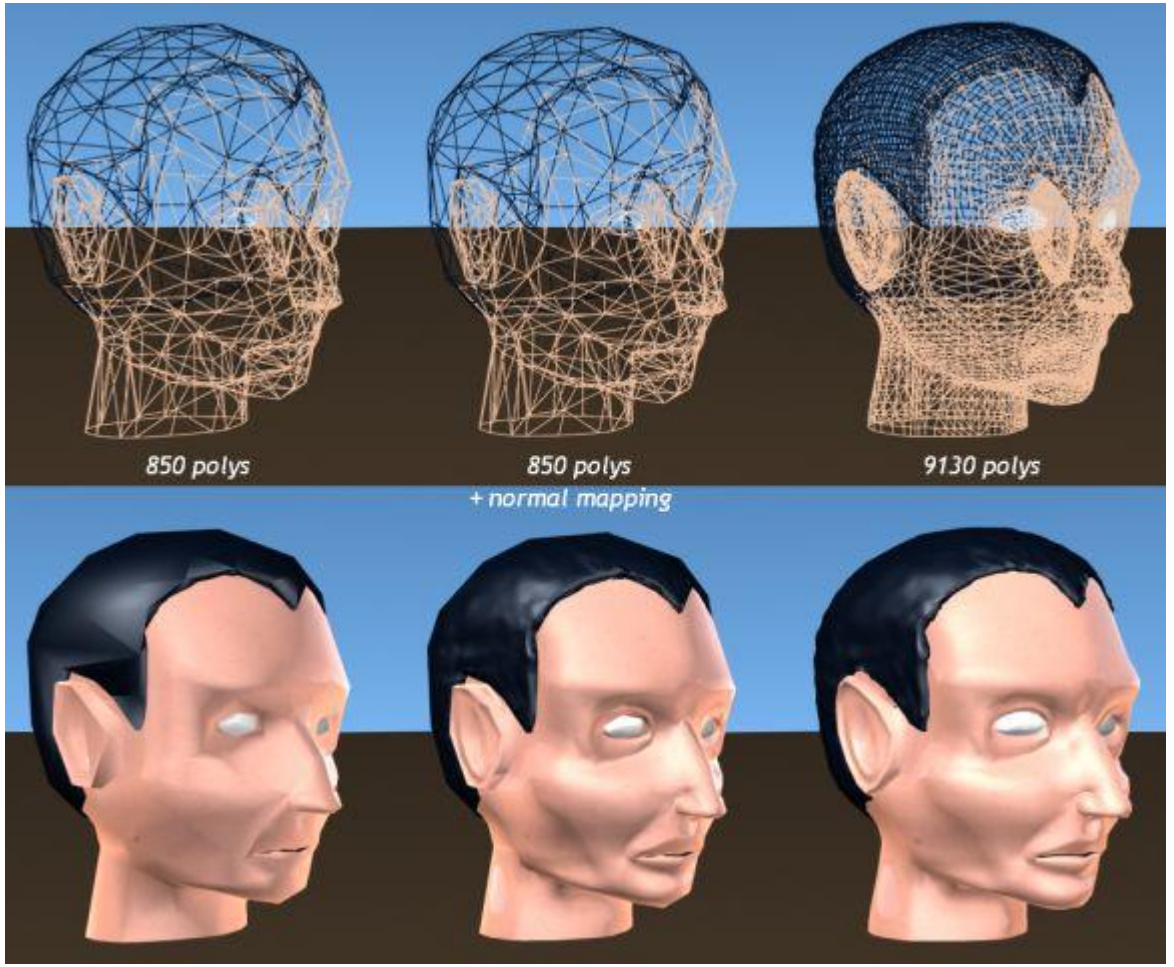


# Результат



# Где Normal Mapping?

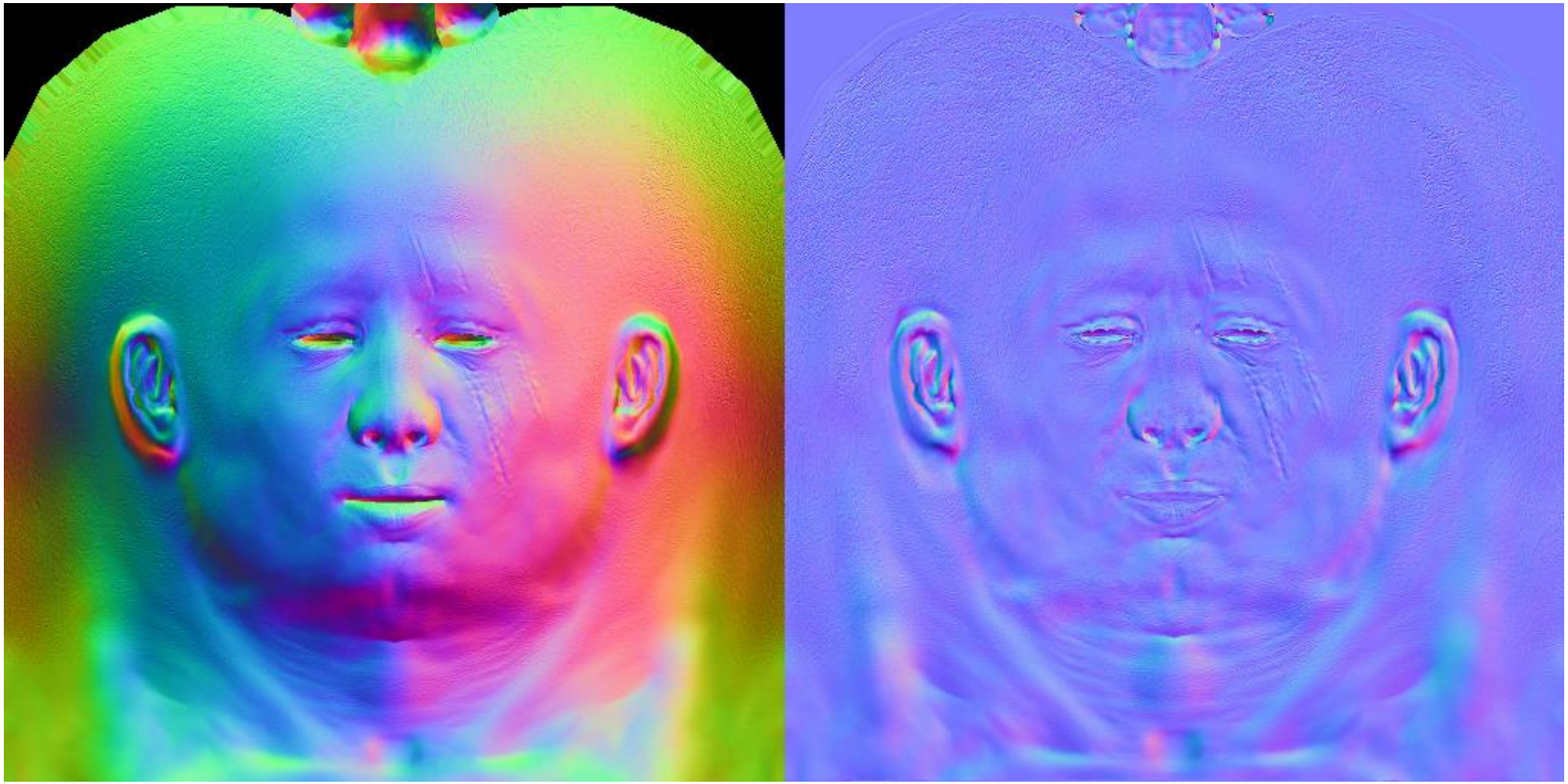
# Normal Mapping: количество полигонов



# Два вида карт нормалей

- В объектном пространстве (object-space): общей системе координат
- В касательном пространстве (tangent-space): локальной системе координат

# Где какая?



# Где какая используется и почему?

- Используется для не деформирующихся объектов, таких как стены, двери и т.п.
- Применяется для возможности деформировать объекты, например, персонажей

# Общая и локальная система координат

# Входные данные для вершинного шейдера

- Позиция камеры в локальной системе координат модели
- Позиция источника света в локальной системе координат модели

# Алгоритм для вершинного шейдера

- Вычислить вектор источника света
- Нормализировать его
- Трансформировать вектор источника света в пространство касательных
- Вычислить вектор камеры
- Нормализировать его
- Просчитать H вектор (модель освещения Блинна; H — half-angle vector)
- Трансформировать H вектор в пространство касательных

# Выходные данные для вершинного шейдера

Вектор источника света в пространстве касательных

H вектор в пространстве касательных

# Входные данные для фрагментного шейдера

- Вектор источника света в пространстве касательных
- H вектор в пространстве касательных
- Color — цвет

# Алгоритм для фрагментного шейдера

- Нормализировать вектор источника света и H вектор в пространстве касательных
- Прочитать нормаль из normal текстуры и нормировать
- Посчитать скалярное произведение вектора нормали из normal текстуры и вектора источника света в пространстве касательных
- Просчитать скалярное произведение вектора нормали из normal текстуры и H вектора в пространстве касательных
- Просчитать цвет

# Применение нормалмаппинга

Сделать низкополигональную модель достаточно детализированной для того, чтобы сохранялась основная форма объекта, и использовать карты нормалей для добавления более мелких деталей.

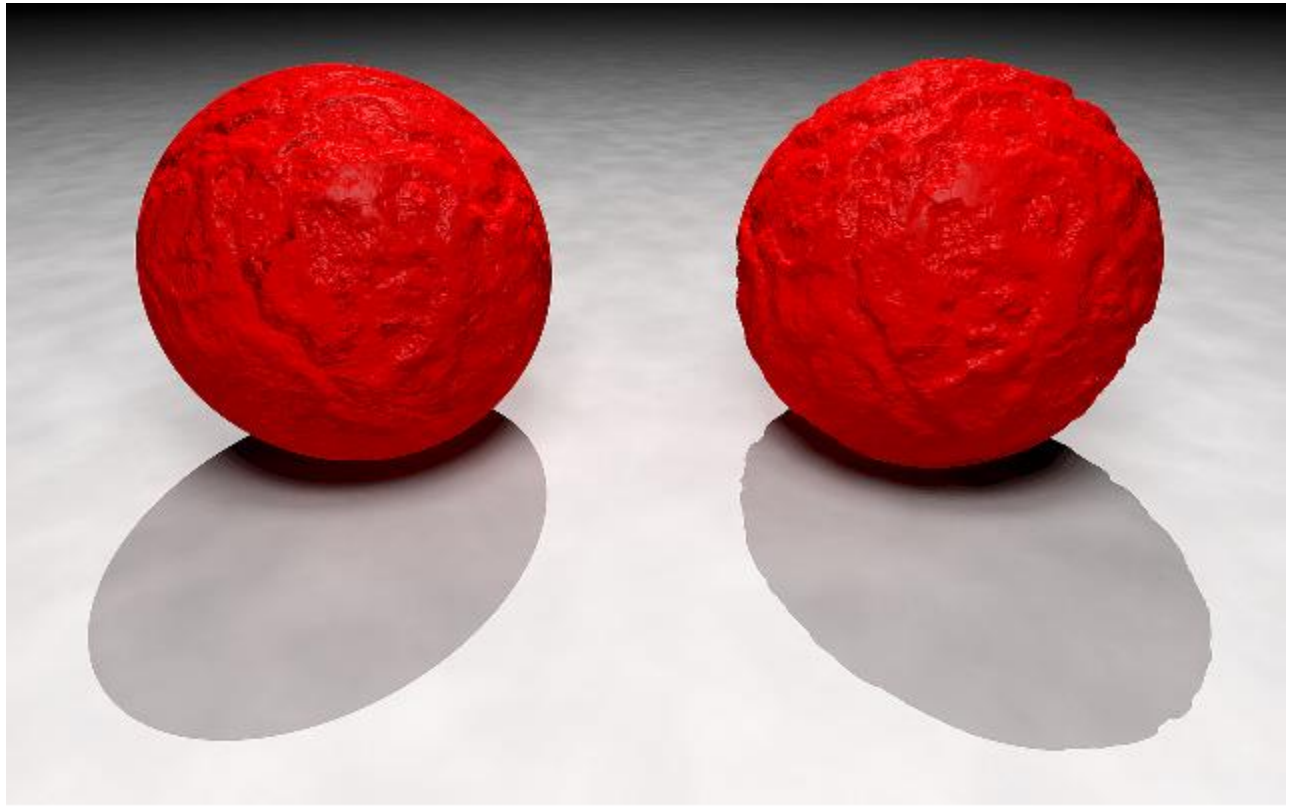
# Применение бампмаппинга

Можно использовать дополнительно и bump map для очень мелких деталей, которые даже в высокополигональной модели не смоделировать (поры кожи, другие мелкие углубления)

# Displacement mapping

- Изменяет **геометрию** поверхности по заданной карте высот, обычно передающейся в вершинный шейдер через текстуру
- Освещение считается обычным способом (пиксельный шейдер может быть практически любым), но требует высокую детализацию модели

# Bump mapping и Displacement mapping



# Displacement mapping

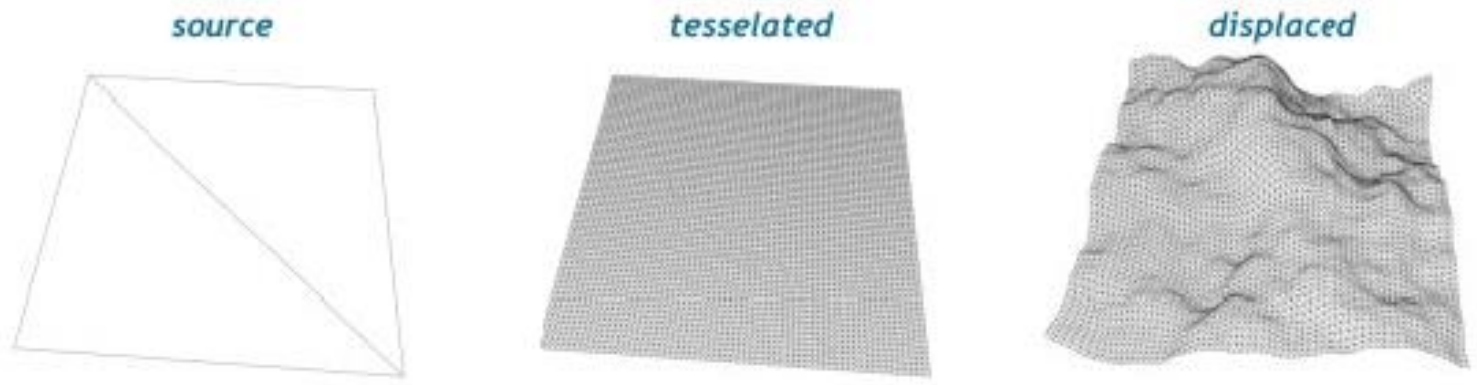
- Метод изменяет положение вершин треугольников, сдвигая их по нормали на величину, исходя из значений в картах смещения

- Карта смещения (displacement map) — это обычно полутоновая текстура, и значения в ней используются для определения высоты каждой точки поверхности объекта (значения могут храниться как 8-битные или 16-битные числа), схоже с bumpmap.

# Создание ландшафта

- Часто карты смещения используются (в этом случае они называются и картами высот) для создания земной поверхности с холмами и впадинами

- Исходными были 4 вершины и 2 полигона, в итоге получился полноценный кусок ландшафта



# Сложный 3D объект

# Сжатие геометрии

- Наложение карт смещения впервые получило поддержку в DirectX 9.0
- Наложение карт смещения можно по существу считать методом сжатия геометрии
- Использование карт смещения снижает объем памяти, требуемый для определенной детализации 3D модели
- Громоздкие геометрические данные замещаются простыми двухмерными текстурами смещения, обычно 8-битными или 16-битными
- Это снижает требования к объему памяти и пропускной способности, необходимой для доставки геометрических данных к видеочипу

# Преимущества

- Позволяет использовать намного более сложные геометрически 3D модели
- Позволяет ускорить анимацию моделей
- Улучшить внешний вид модели, применив более сложные комплексные алгоритмы и техники, вроде имитации тканей (cloth simulation)

# Ограничения

- Гладкие объекты, не содержащие большого количества тонких деталей, будут лучше представлены в виде стандартных полигональных сеток или иных поверхностей более высокого уровня, вроде кривых Безье
- Очень сложные модели, такие как деревья или растения, также нелегко представить картами смещения
- Почти всегда требует специализированных утилит, ведь очень сложно напрямую создавать карты смещения (если речь не идет о простых объектах, вроде ландшафта)

# Parallax mapping

- Parallax mapping был представлен Томомити Канэко (англ. Tomomichi Kaneko) в 2001 году www.researchgate.net/publication/228583097_Detailed_shape_representation_with_parallax_mapping
- В 2004 году Welsh продемонстрировал применение параллаксмаппинга на программируемых видеочипах
- Parallax mapping полностью исполняется на графических процессорах видеокарты как пиксельный шейдер

# Другие названия

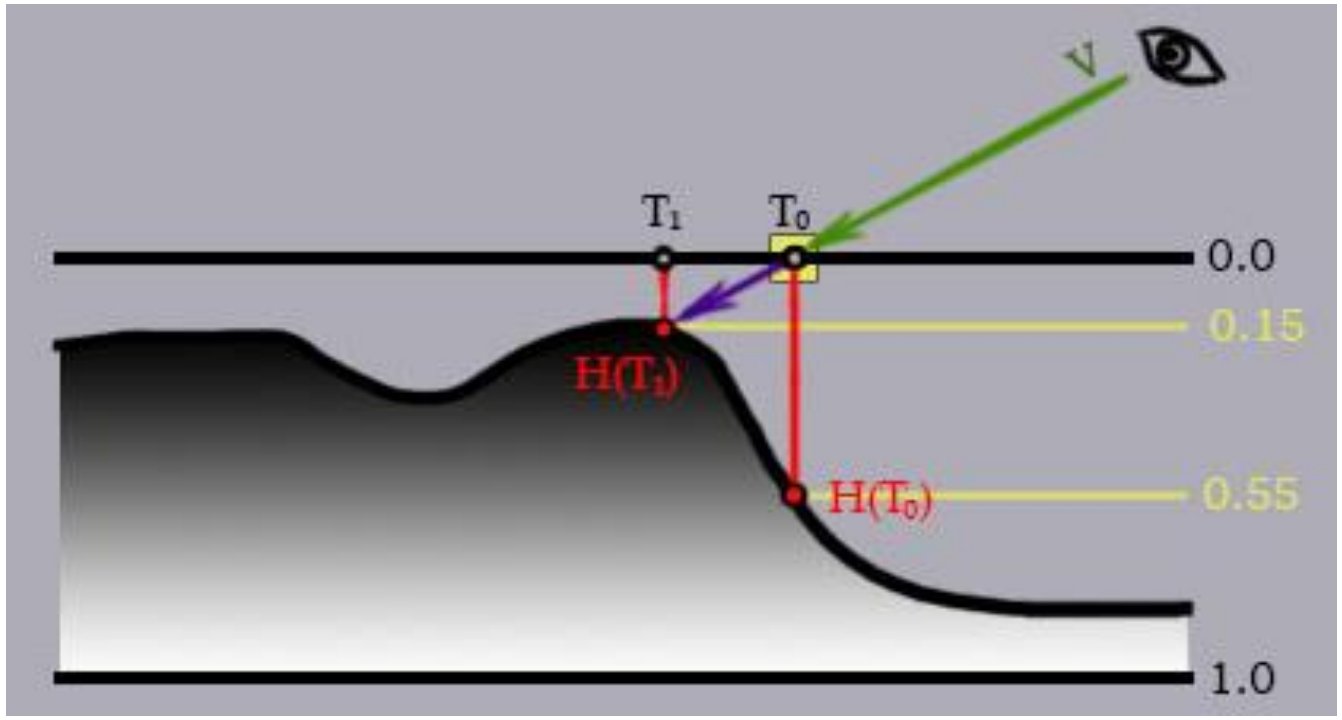
- Parallax Mapping
- Offset Mapping
- Virtual Displacement Mapping
- Per-Pixel Displacement Mapping

# Parallax mapping

Parallax mapping осуществляется смещением текстурных координат так, чтобы поверхность казалась объёмной

Скриншот из демонстрационного бенчмарка графического движка Irrlicht Engine

# Идея Parallax mapping




Идея метода состоит в том, чтобы возвращать текстурные координаты той точки, где видовой вектор пересекает поверхность

Это требует просчета лучей (рейтрейсинг) для карты высот

# Parallax mapping и аппрокимация

- Если карта высот не имеет слишком сильно изменяющихся значений («гладкая» или «плавная»), то можно обойтись аппроксимацией без использования рейтрейсинга
- Parallax mapping условно можно назвать «2.5D»
- Parallax mapping похож одновременно на наложение карт смещения и нормалмаппинг, это нечто среднее между ними

# Displacement mapping и Parallax mapping

Главное отличие **parallax mapping** от **displacement mapping** в том, что в нём все расчеты **попиксельные**, а не **повершинные**

# Normal mapping и Parallax mapping

- Данная технология также использует карты нормалей
- В отличие от normal mapping, она реализует не только освещение с учётом рельефа, но и сдвигает координаты диффузной текстуры
- Этим достигается наиболее полный эффект рельефа, особенно при взгляде на поверхность под углом

- Parallax mapping отличается от normal mapping всего тремя инструкциями пиксельного шейдера: две математические инструкции и одна дополнительная выборка из текстуры
- После того, как вычислена новая текстурная координата, она используется дальше для чтения других текстурных слоев: базовой текстуры, карты нормалей и т. п

# Применение в играх

# Применение в играх

# Недостатки Parallax mapping

- Использование обычного параллаксмаппинга ограничено картами высот с небольшой разницей значений.
- "Крутые" неровности обрабатываются алгоритмом некорректно, появляются различные артефакты, "плавание" текстур и пр.

# Модификации Parallax mapping

Несколько исследователей (Yerex, Donnelly, Tatarchuk, Policarpo) описали новые методы, улучшающие начальный алгоритм.

Почти все идеи основаны на трассировке лучей в пиксельном шейдере для определения пересечений деталей поверхностей друг другом.

Модифицированные методики получили несколько разных названий: Parallax Mapping with Occlusion, Parallax Mapping with Distance Functions, Parallax Occlusion Mapping.

# Parallax occlusion mapping

- Parallax occlusion mapping (сокращённо POM)
- Усовершенствованный вариант техники «parallax mapping»
- Позволяет корректно выполнять определение перспективы и самозатенение в реальном времени

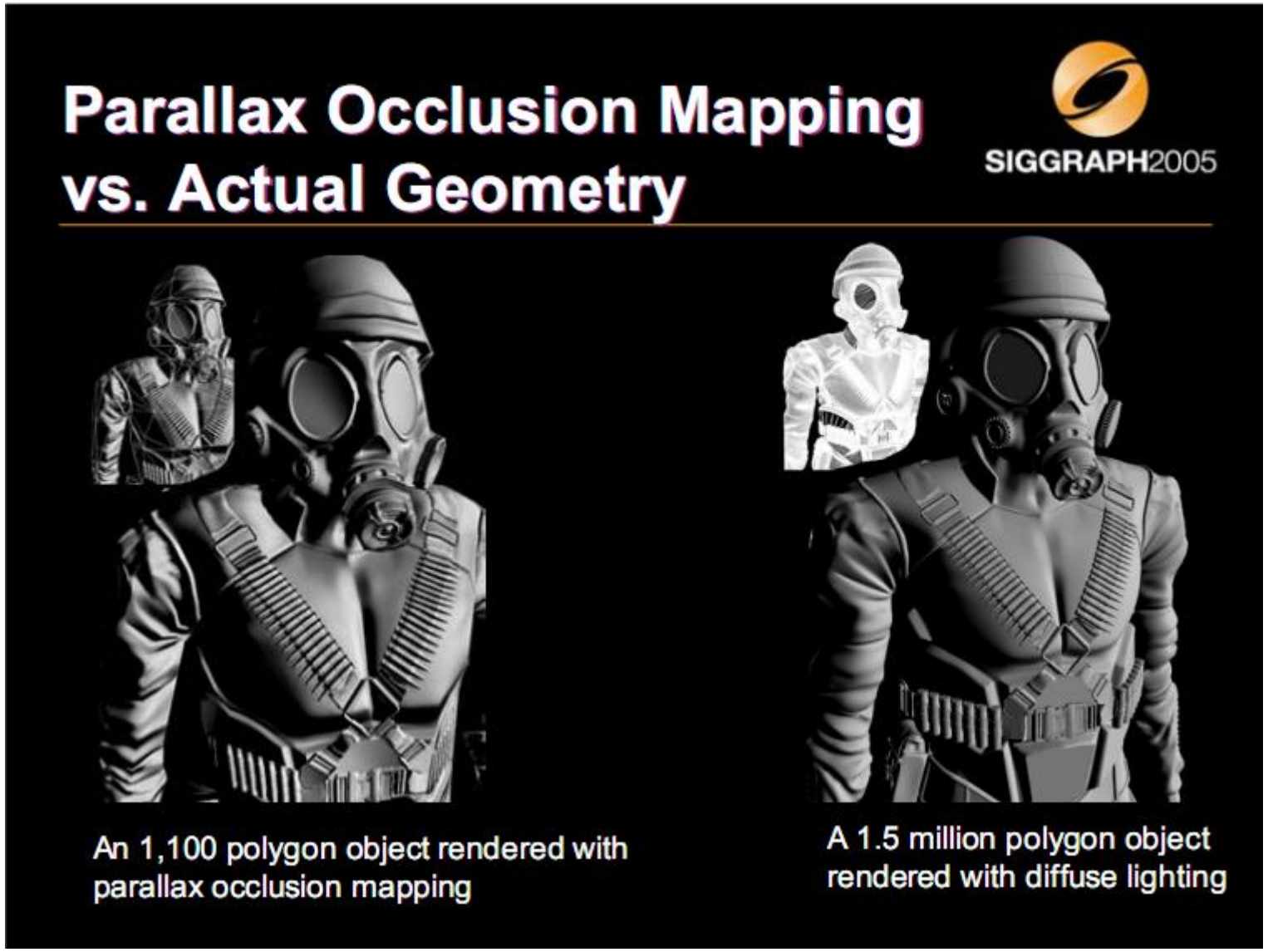
# Немного истории

- Первая работа, посвященная данной методике, появилась в 2004 году на «ShaderX3», её авторами были Зоя Броули (англ. Zoe Brawley) и Наталия Татарчук, затем  на мероприятии SIGGRAPH 2005

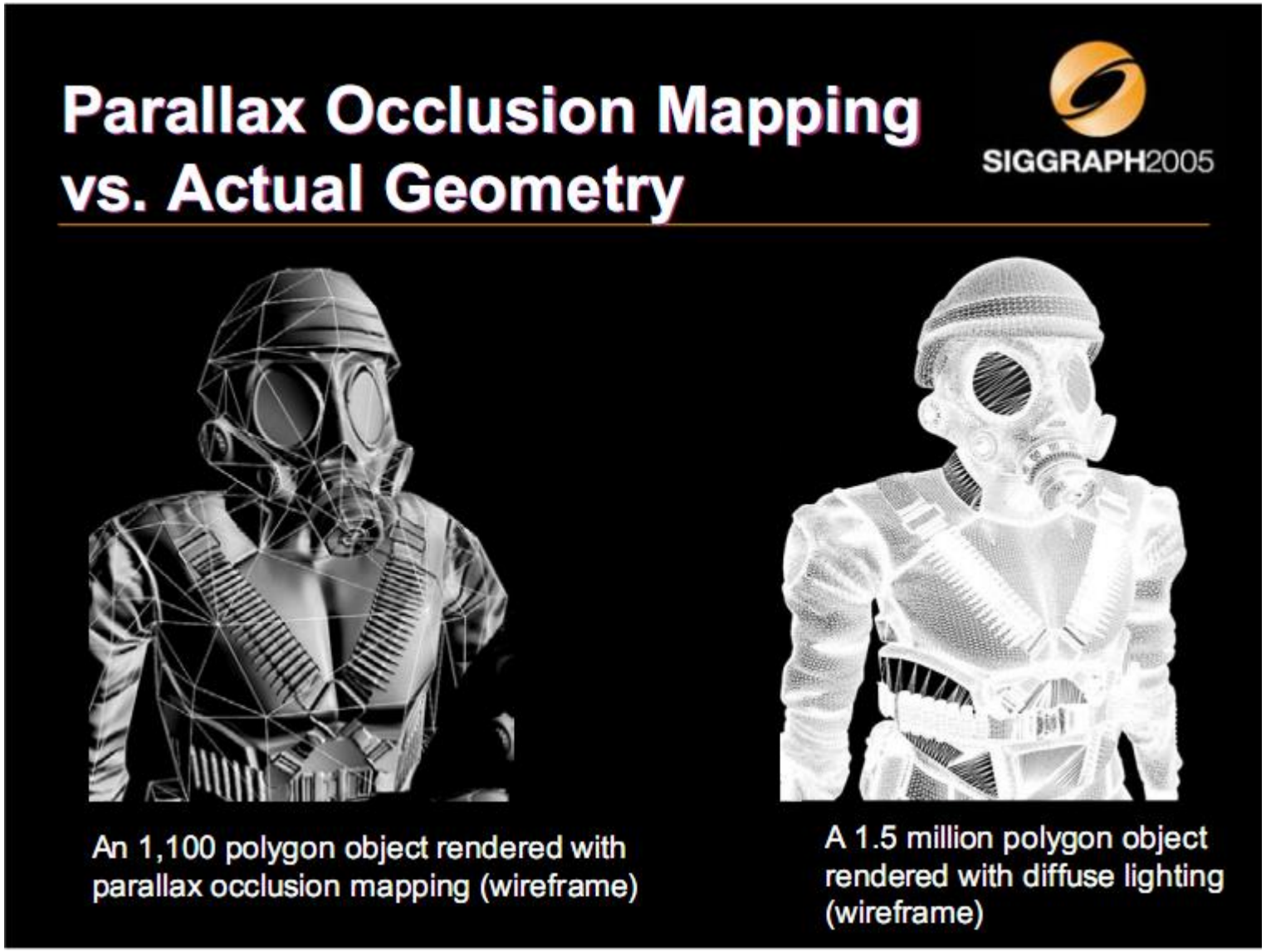
- Техника «Parallax occlusion mapping» использовалась компанией ATI  для презентации возможностей третьей версии шейдерной модели в новейшей на то время видеокарте Radeon X1800

- Первым игровым движком, в котором использовался Parallax occlusion mapping, стал CryEngine 2 от немецкого разработчика Crytek, который впервые использовался в компьютерной ПК-игре Crysis 2007 года выпуска

# Из доклада Наталии Татарчук



# Из доклада Наталии Татарчук



# Из доклада Наталии Татарчук

## Parallax Occlusion Mapping vs. Actual Geometry

SIGGRAPH2005

-1100 polygons with parallax occlusion mapping (8 to 50 samples used)
- **Memory**: 79K vertex buffer
  6K index buffer
  13Mb texture (3Dc)
  (2048 x 2048 maps)

Total: **< 14 Mb**

**Frame Rate:**
- ***255 fps*** on ATI Radeon hardware
- ***235 fps*** with skinning

- 1,500,000 polygons with diffuse lighting
- **Memory**: 31Mb vertex buffer
  14Mb index buffer

Total: **45 Mb**

**Frame Rate:**
- ***32 fps*** on ATI Radeon hardware

# Parallax occlusion mapping

Реализация алгоритма Parallax occlusion mapping в бенчмарке 3DMark Vantage 2008 года. Данное изображение построено при помощи двух полигонов (треугольников), расположенных на одной плоскости

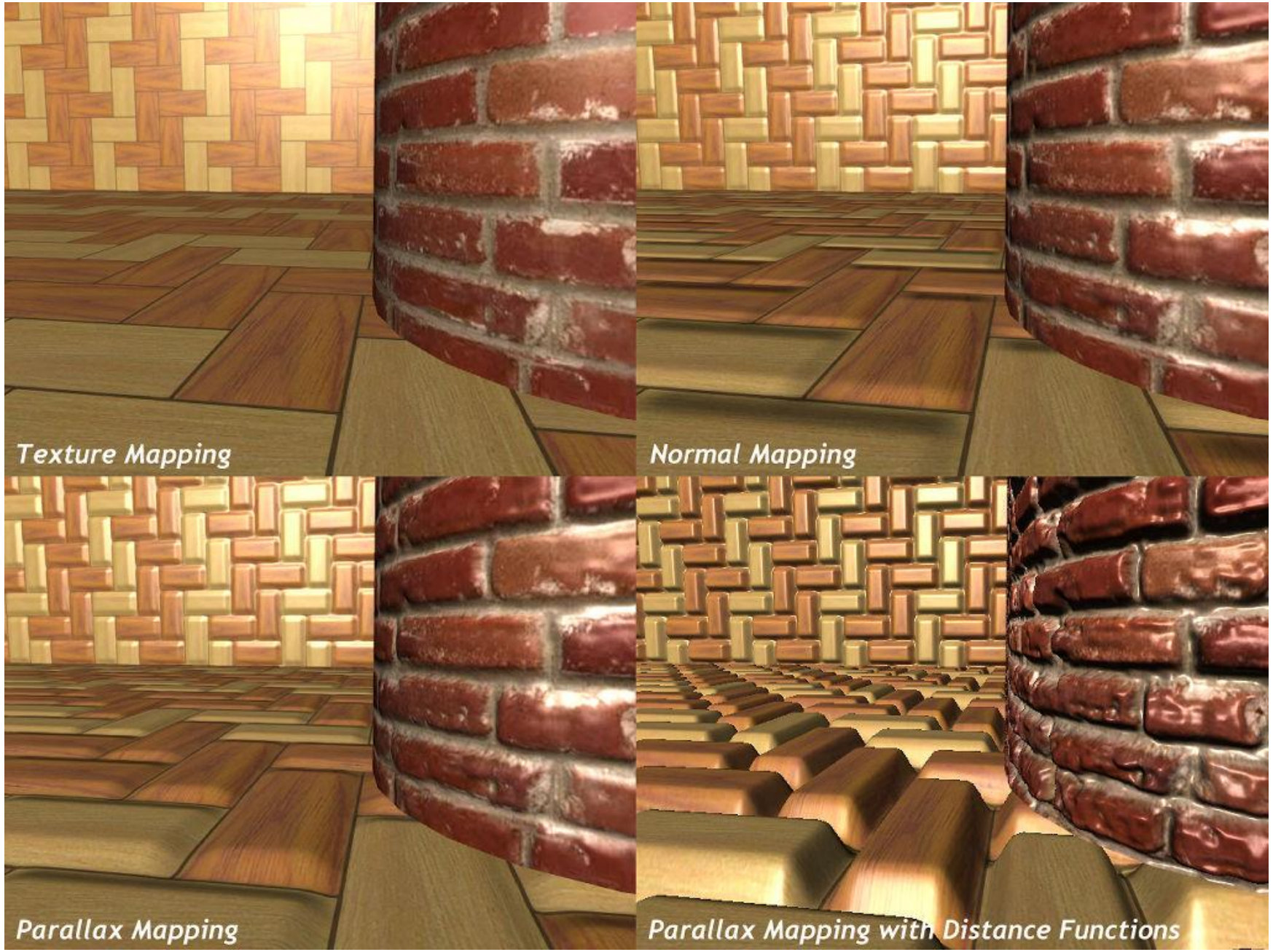
# Parallax occlusion mapping

# Недостатки

Невысокая детализация силуэтов и граней

Видеокарта должна обеспечивать надлежащий уровень скорости исполнения операций ветвления в пиксельном шейдере

Отсутствие геометрически правильных силуэтов (краев объекта)


Texture Mapping
Normal Mapping
Parallax Mapping
Parallax Mapping with Distance Functions