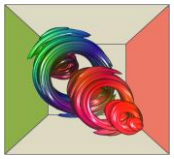
LOBACHEVSKY STATE UNIVERSITY
of NIZHNI NOVGOROD
National Research University

Computing Mathematics and Cybernetics faculty

Software department



CS255. Computer Graphics Introduction Course

Научная визуализация. Стереовизуализация

проф. ННГУ Турлапов В.Е.
vadim.turlapov@cs.vmk.unn.ru

Открытые системы геометрического моделирования, инженерного и научного анализа и визуализации

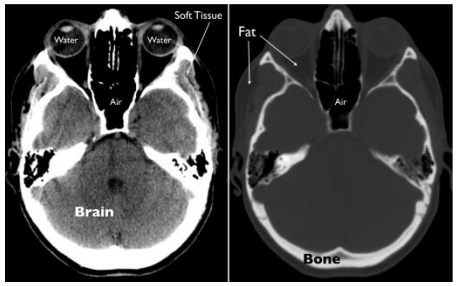
- Открытая система научно-технической визуализации **ParaView** http://www.paraview.org/
- **Visualization Toolkit (VTK)** is an open-source, freely available software system for 3D computer graphics, image processing and visualization www.vtk.org
- National Library of Medicine **Insight Segmentation and Registration Toolkit (ITK)**. http://www.itk.org/

- Система трехмерного моделирования, сеточной декомпозиции и инженерного анализа SALOME (localLink)
- Открытая система и технология для геометрического проектирования и инженерного анализа Open CASCADE Technology (localLink) http://www.opencascade.org/ http://ru.wikipedia.org/wiki/Open_CASCADE_Technology
- **OpenFOAM** (англ. *Open Source Field Operation And Manipulation CFD ToolBox*) — открытая интегрируемая платформа для численного моделирования задач механики сплошных сред.

Ресурсы сообщества компьютерной графики

- SIGGRAPH.org (http://www.siggraph.org/),
- SIGGRAPH 2014 (http://s2014.siggraph.org)
- SIGGRAPH Asia 2014 (http://sa2014.siggraph.org/),
- Open Access SG2014-Courses (http://www.siggraph.org/learn/open-access-s2014-courses).
- Computer Graphics Bibliography Database

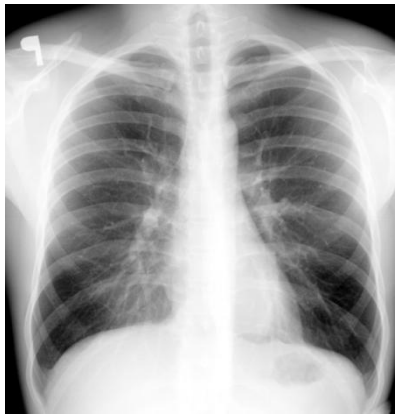
Медицинская визуализация



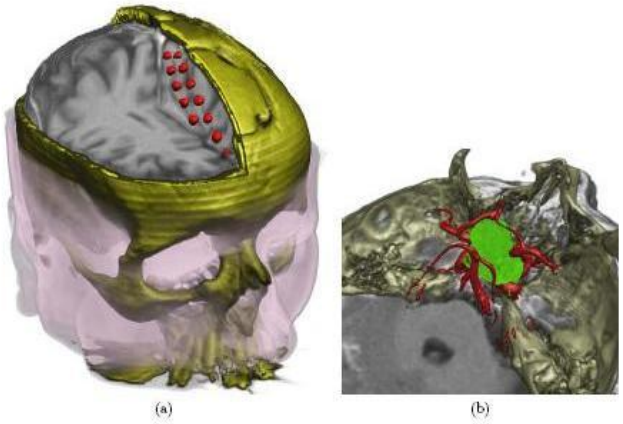
Двумерные чёрно-белые слои

Проекция максимальной интенсивности (MIP)

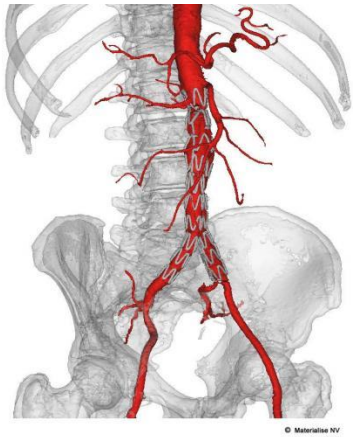
Мультипланарная реконструкция (MPR)



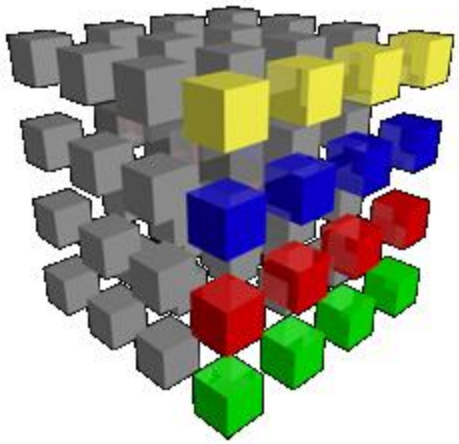
Ренген (интегральная Характеристика плотности)



Мультибъёмный рендеринг (2 и более 3D массивов)



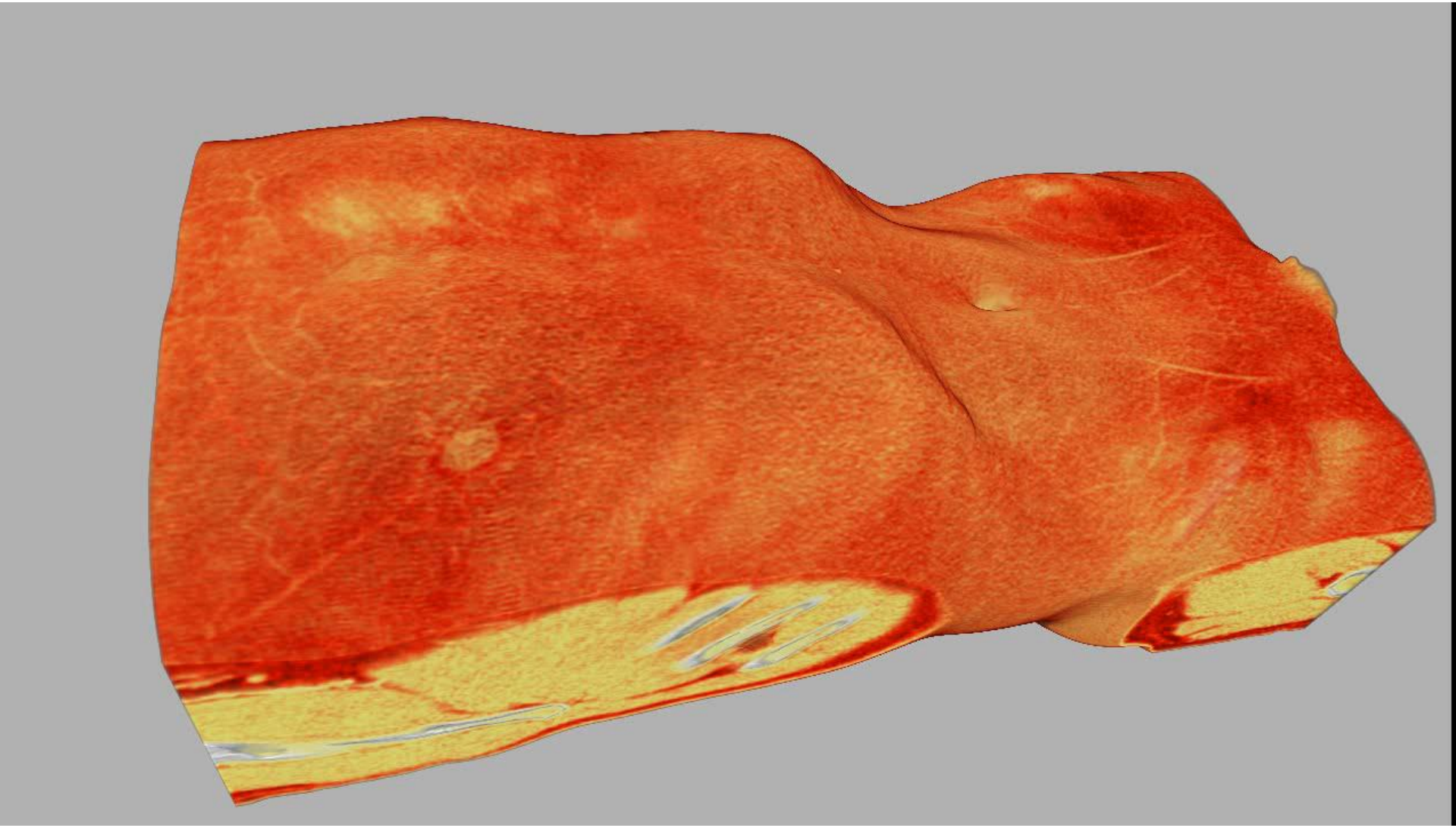
Объёмный рендеринг сегментированных данных



Volume Rendering

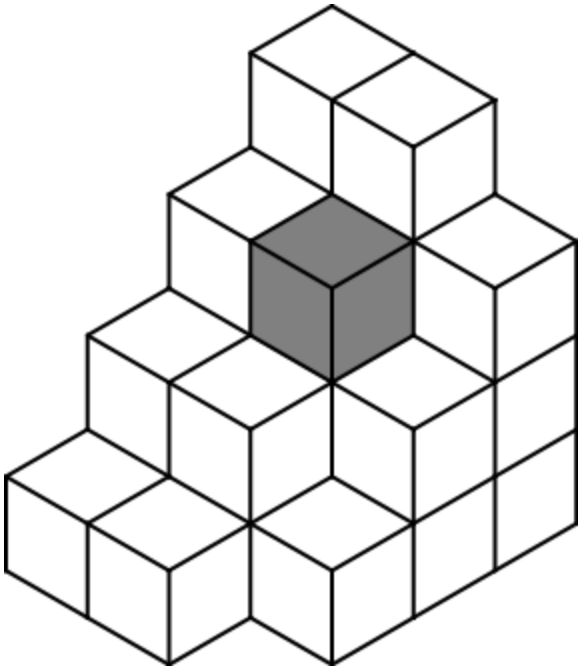
Основные понятия

Пример визуализации



Объёмные данные. Воксели

- Трёхмерная матрица вокселей
  - Воксели лежат в узлах регулярной прямоугольной сетки
- Воксель – элемент объёмных данных
  - Характеризует данные в окрестности своего расположения
  - Может быть не только скаляром (температура, плотность, …), но и вектором, матрицей, и т.д.
- Интерполяция
  - При выборке значения данных из произвольной точки пространства берётся интерполированное значение

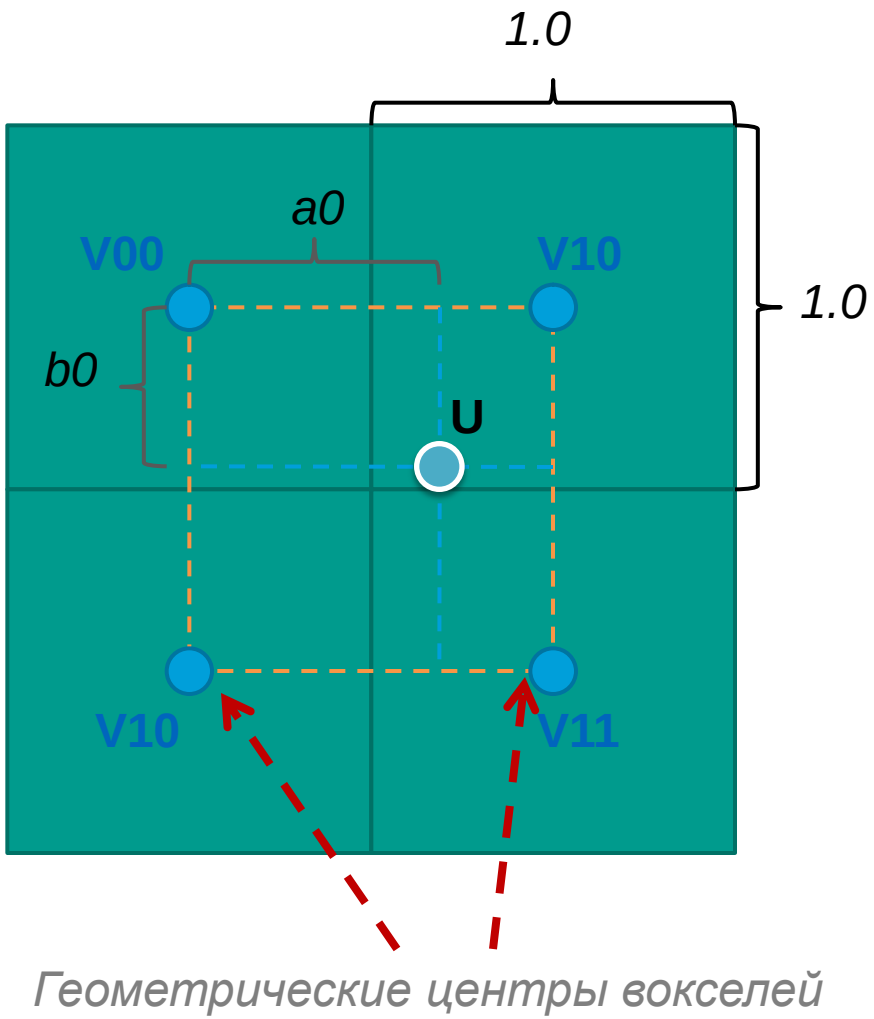


Билинейная и трилинейная интерполяции

U = *a1*b1**V00 +
*a0*b1**V10 +
*a1*b0**V01 +
*a0*b0**V11

a1=1 – a0
b1=1 – b0

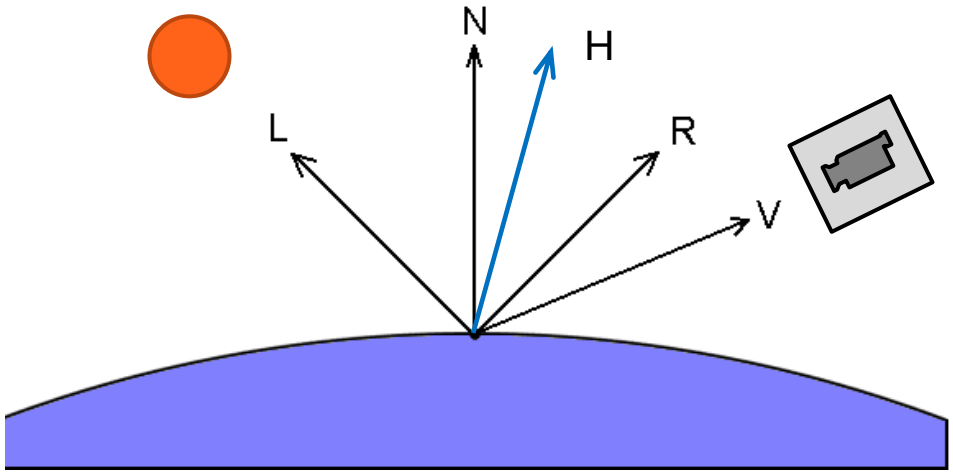
Vij – значения вокселей



Локальное освещение

- Улучшает пространственное восприятие обьектов

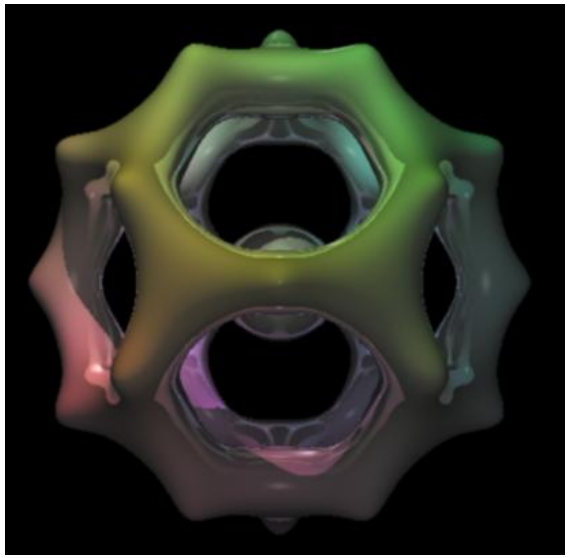
Модель освещения Фонга или Блина-Фонга (с Half-вектором)

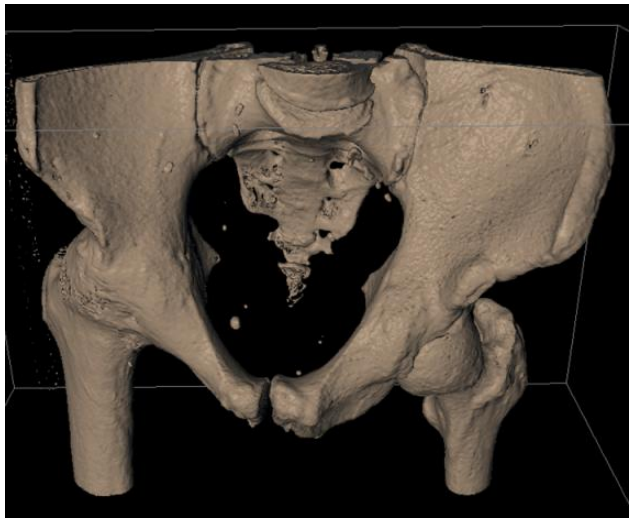


Diffuse = (N,L);
Specular = (V,R); R = L – 2*(N,L)*N;
или Specular = (N,H); H = norm(L + V);
Luminance = Ka + Kd*Color*Diffuse + Ks*Specular^s

Изоповерхности в трехмерном скалярном поле

- Место точек, где данные принимают определённое значение L (изозначение)





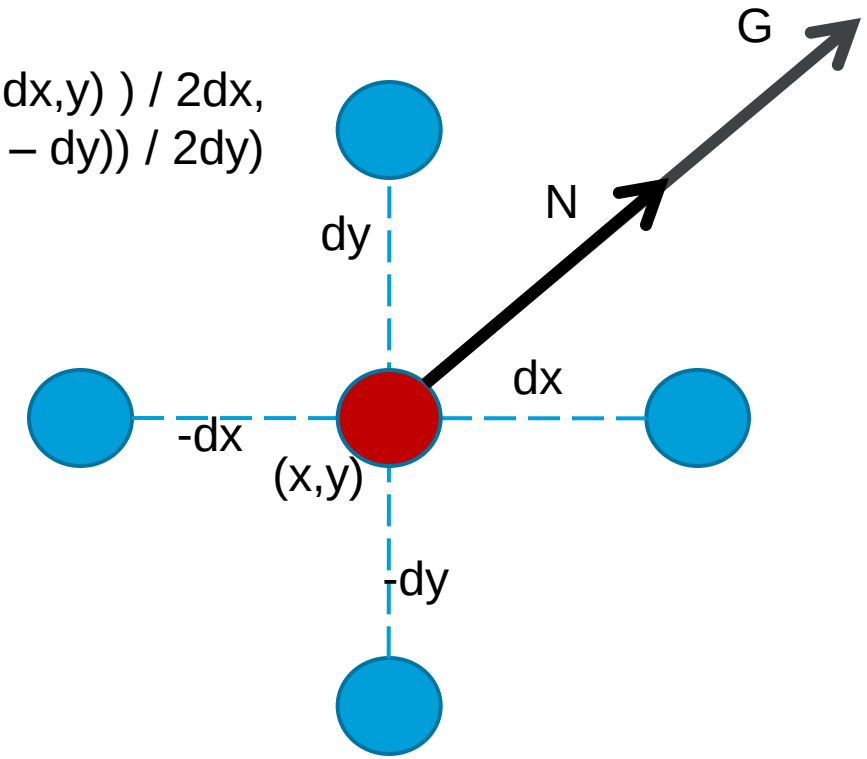
```
F(X,Y,Z) =
2.0 - cos ( X + T * Y ) - cos ( X - T * Y ) -
cos ( Y + T * Z ) - cos ( Y - T * Z ) -
cos ( Z - T * X ) - cos ( Z + T * X ) = 0
```

Изоповерхность для массива CT-данных

Вычисление градиента и нормали

$$G = \mathrm{Grad}\, U(x,y) \approx \left(\frac{U(x + dx,y) - U(x - dx,y)}{2dx}, \frac{U(x,y + dy) - U(x, y - dy)}{2dy} \right)$$

$$N = G / |G|$$



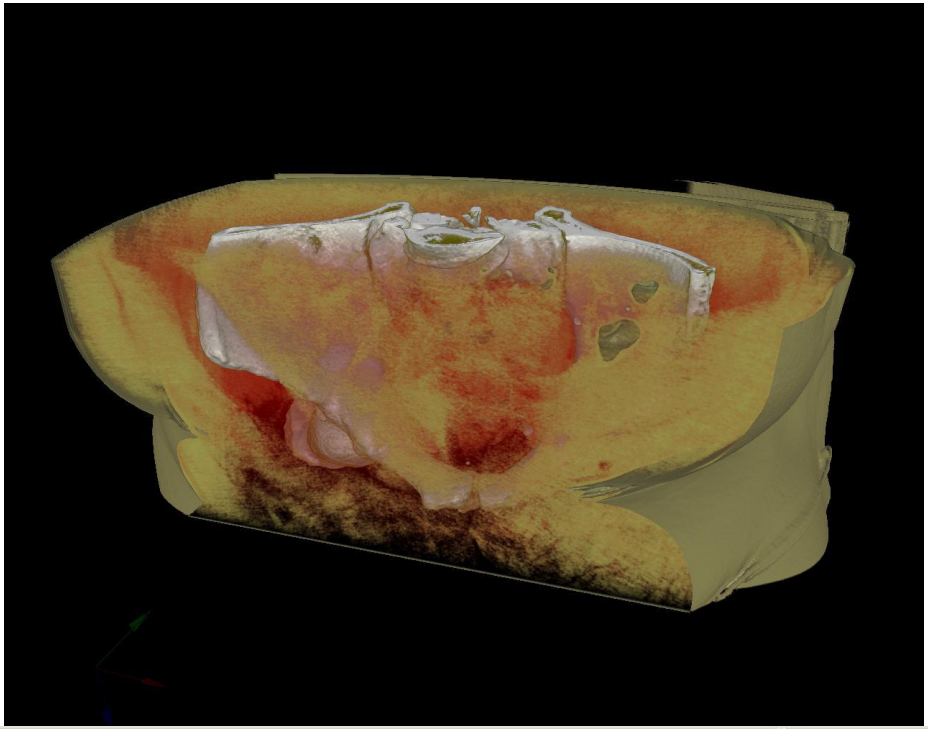
Transfer Function, TF (передаточная функция отображения)

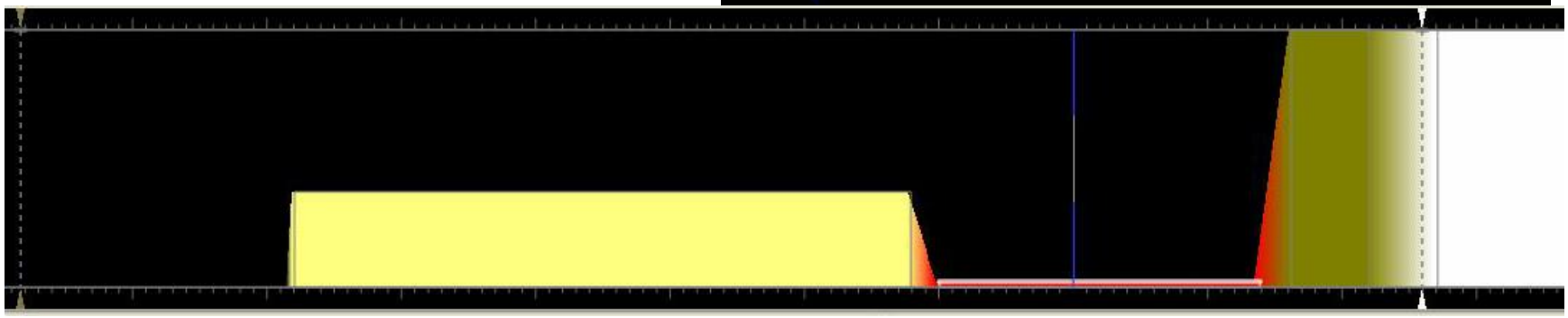
- Любому возможному значению исходных данных ставит в соответствие определённые оптические свойства

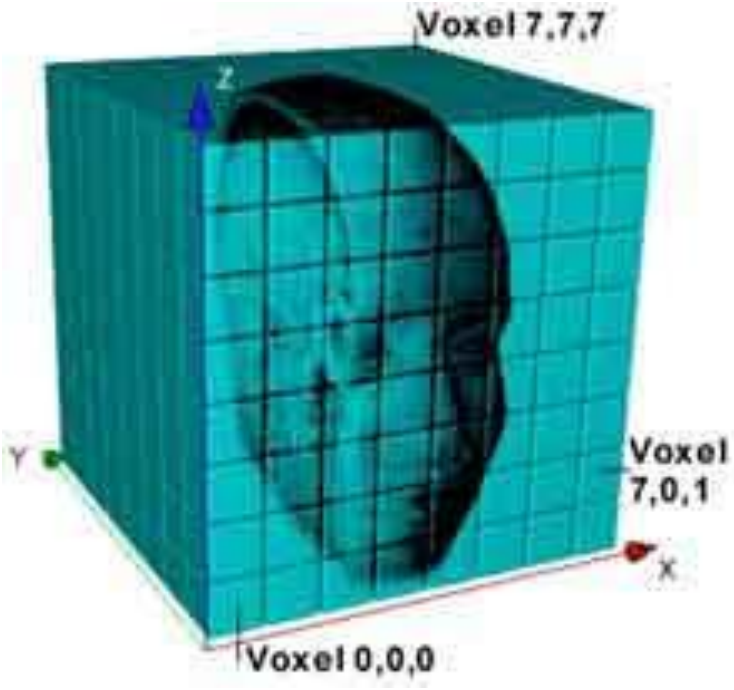
- Можно раскрашивать пространство с данными в соответствии с функцией TF, в результате получается разноцветные затуманенные области

Комбинация изоповерхностей и сред

Изоповерхности для костей и кожи, область жировой ткани закрашена жёлтым цветом



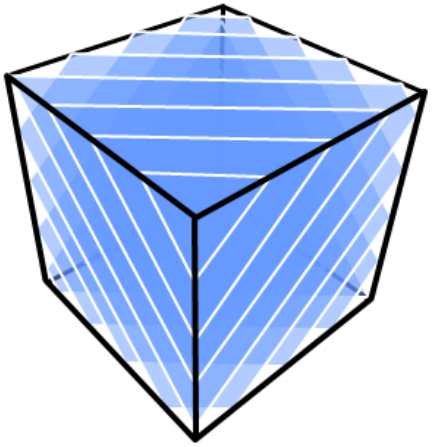




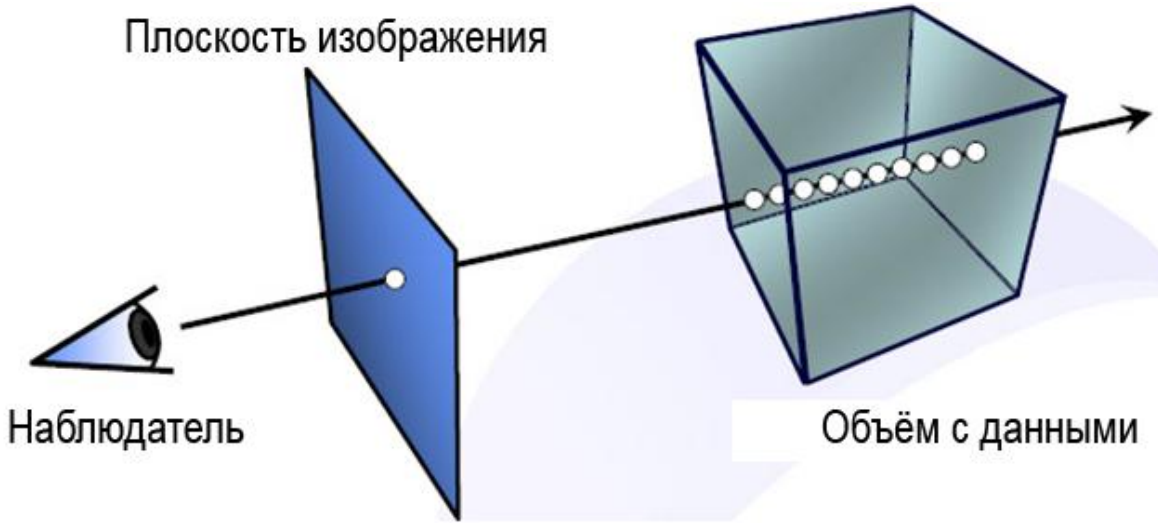
Методы объёмного рендеринга

Методы объёмного рендеринга

- **Метод слоёв** (визуализация сечений объема плокостью от заднего к переднему)
  - Производительность

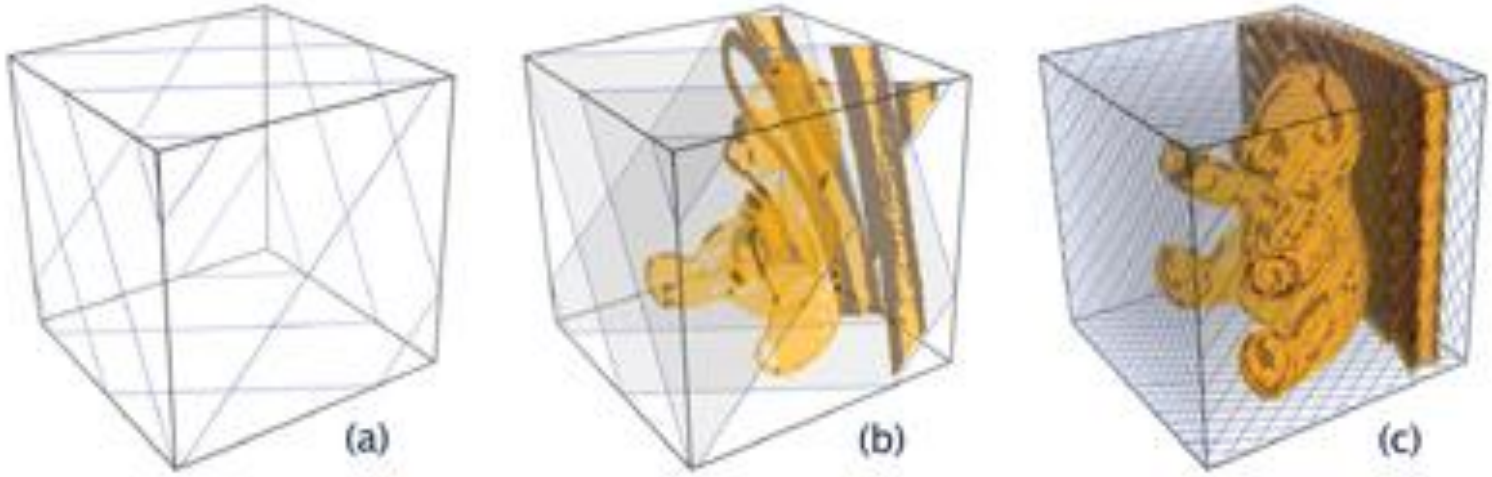


- **Метод обратной трассировки лучей**
  - Качество
  - Гибкость



Метод слоёв: Slice-Based Volume Rendering (SBVR)

- В качестве слоёв (сечений) обычно используются полигоны
- Слои рисуются в обратном для наблюдателя порядке
- Каждый слой закрашивается согласно Transfer Function в соответствии с объёмными данными

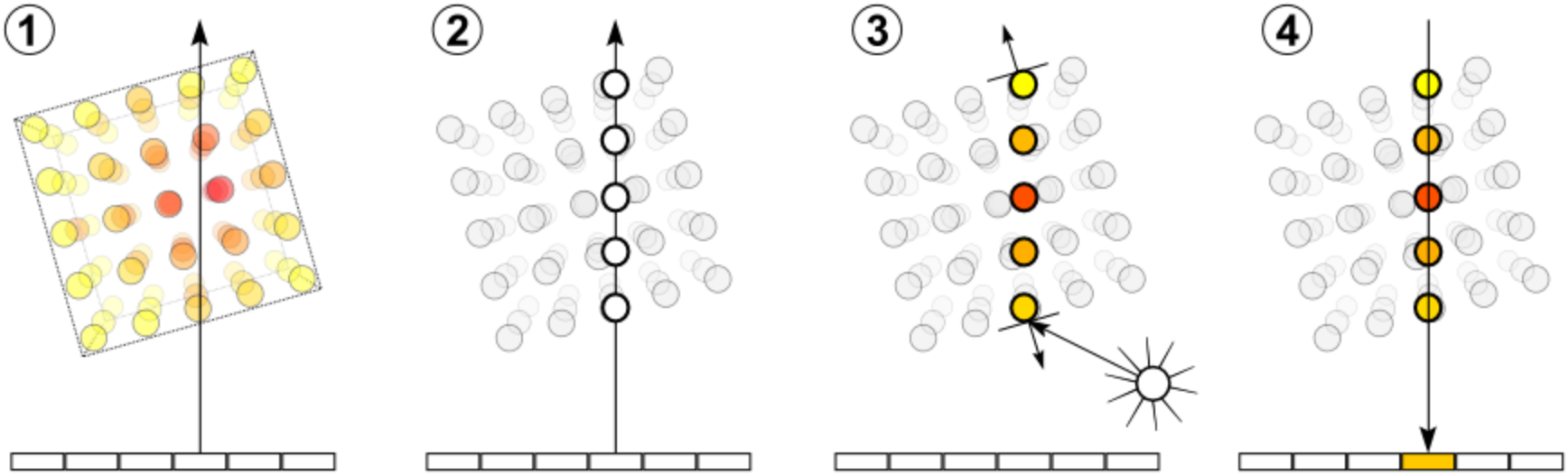


Фрагментный шейдер для метода слоёв

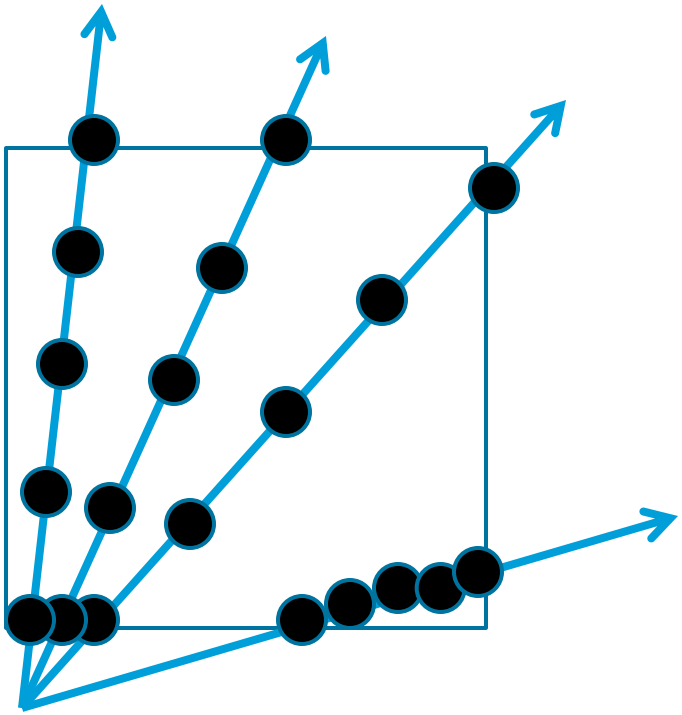
```
void main(    uniform sampler3D dataTex,
              uniform sampler1D tfTex,
              float3 texCoord : TEXCOORD0,
              float4 color : COLOR
         )
{
  float v = tex3d(texCoord, dataTex); // Читаем 3D данные
  color = tex1d(v, tfTex); // transfer function
}
```


Трассировка луча

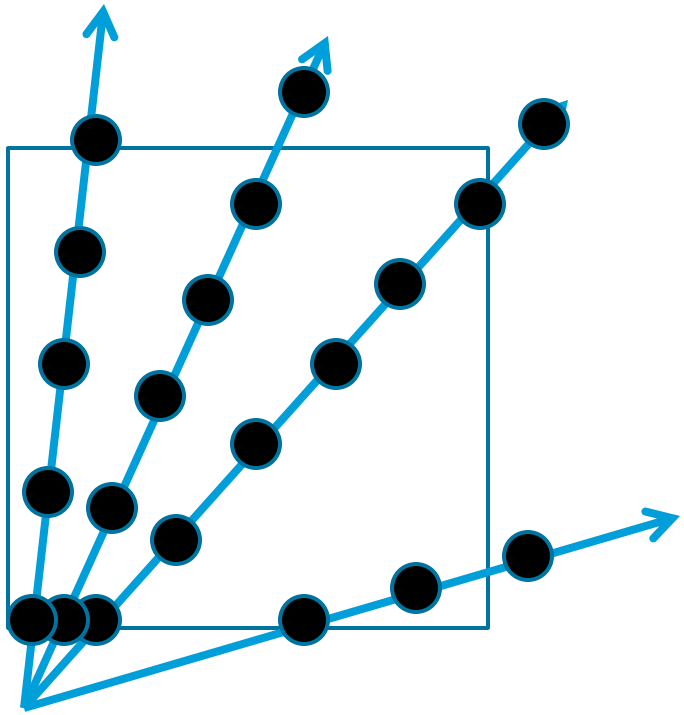
1. Генерация луча для каждого пикселя экрана
2. Конечное число шагов вдоль луча
3. Накопление (интегрирование) цвета
4. Определение итогового цвета



Выбор шага вдоль луча и Размер вокселя



Фиксируем число шагов



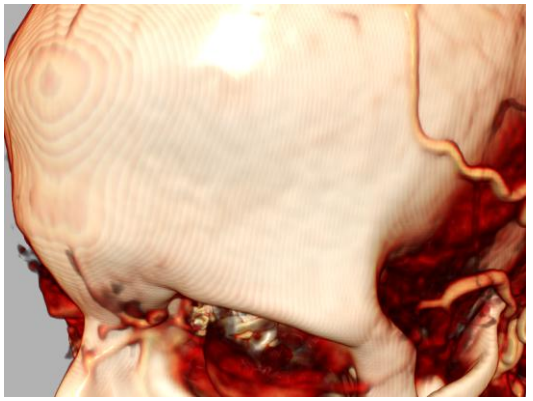
Фиксируем длину шага относительно размеров вокселя

Проблемы.

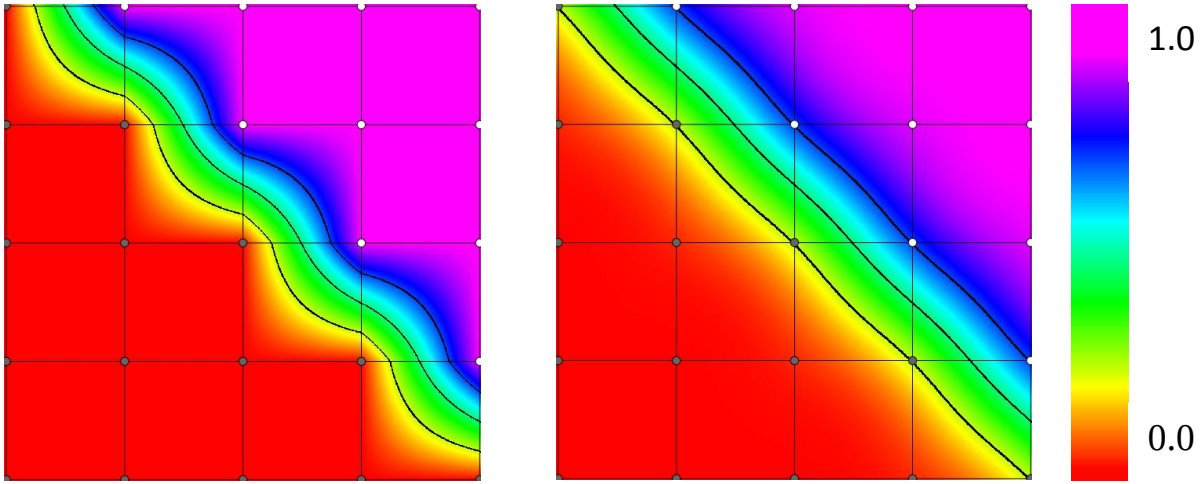
Процесс визуализации объёмных данных всегда включает в себя этапы *интерполяции* и *классификации*

1. Интерполяция

Ступенчатые дефекты визуализации, возникающие на данном этапе, значительно сокращают с помощью трикубической интерполяции

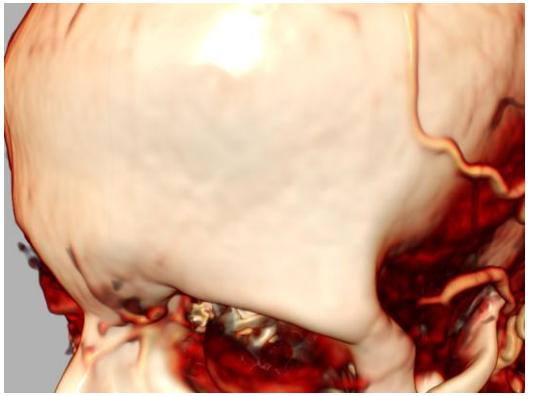


Трилинейная интерполяция



Билинейная интерполяция

Бикубическая интерполяция

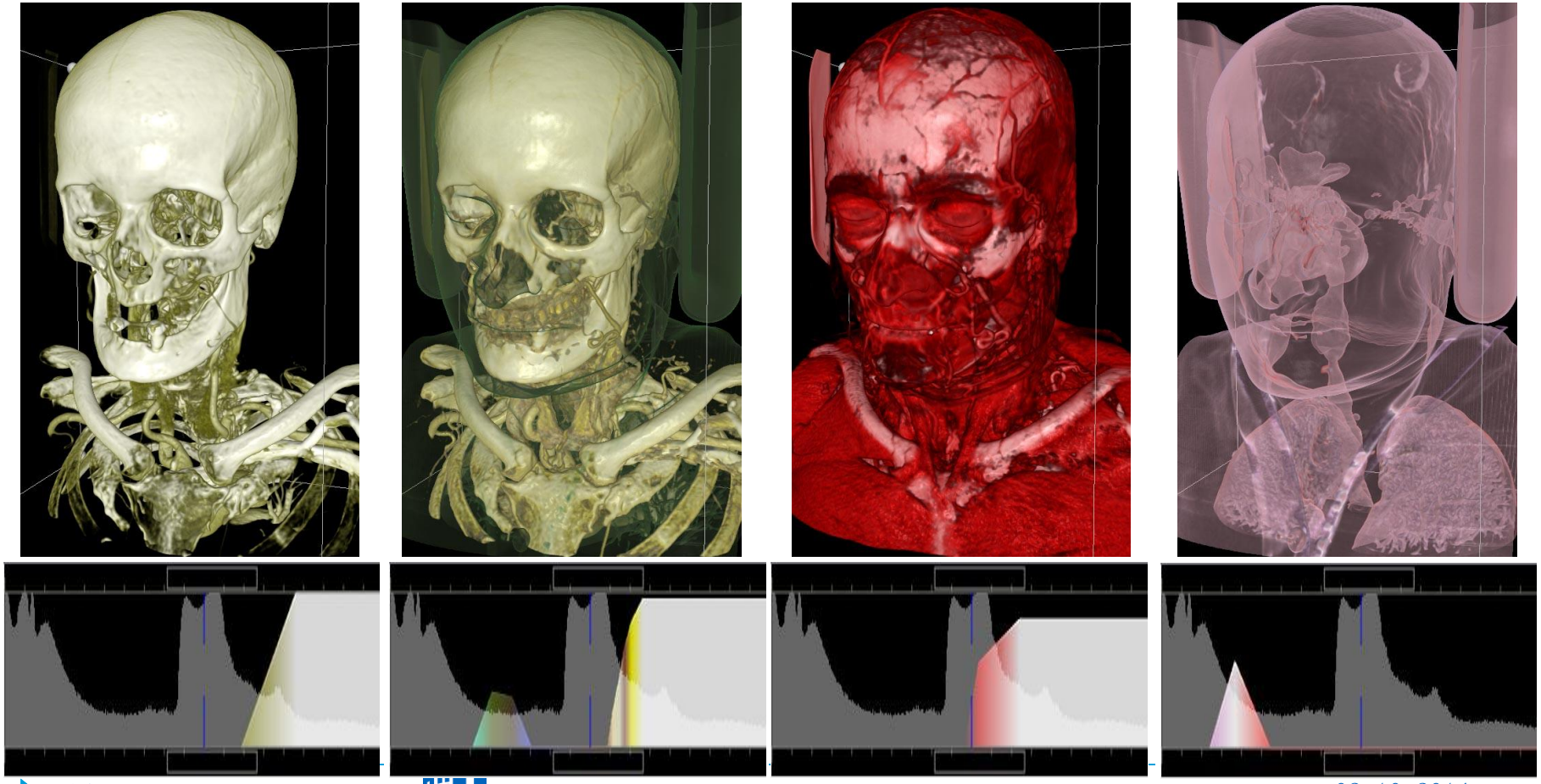


Трикубическая интерполяция

Проблемы.

Процесс визуализации объёмных данных всегда включает в себя этапы *интерполяции* и *классификации*

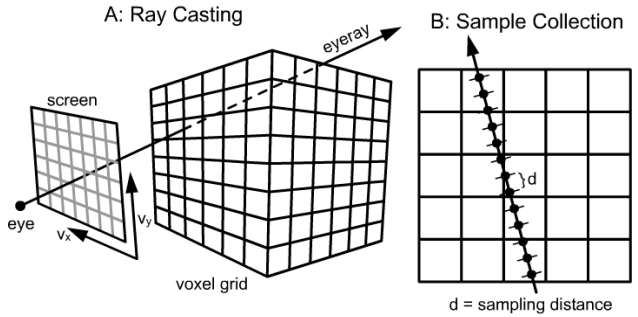
2. Классификация осуществляется через *Transfer Function*

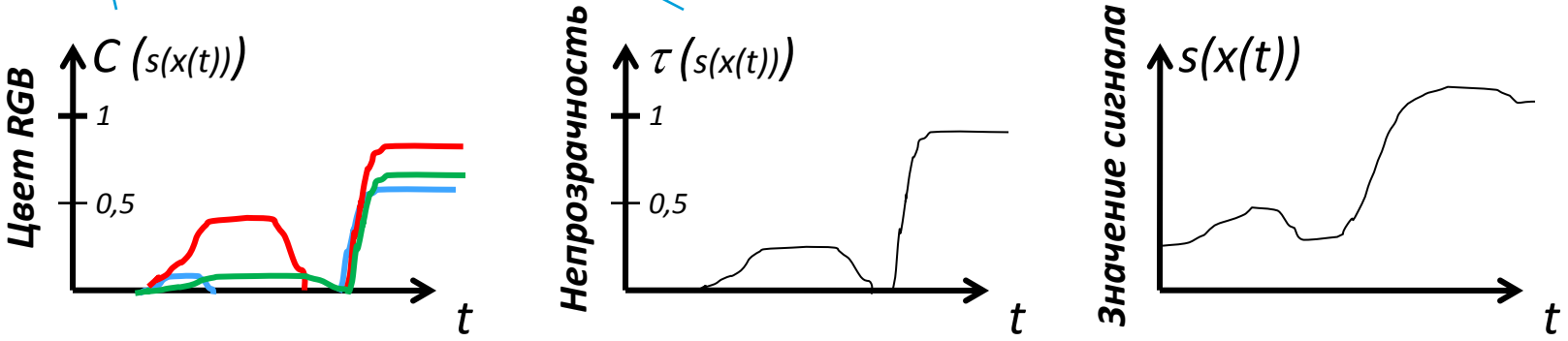


(1) (2) (3) (4)

Математическая модель трассировки луча: интеграл объемного рендеринга

$$I = \int_0^{t_E} C\big(s(\mathbf{x}(t))\big) \times \tau\big(s(\mathbf{x}(t))\big) \times \exp\left(-\int_0^{t} \tau\big(s(\mathbf{x}(t'))\big)\,dt' \right) dt$$





Пример зависимостей различных величин в подынтегральной функции

*C(s(**x**)) – постклассифицированный цвет для значения поля **s** в точке **x***

*τ(s(**x**)) – постклассифицированная непрозрачность для значения поля **s** в точке **x***

Интеграл описывает процесс накопления (интегрирования) цвета вдоль луча **x**(*t*) с учетом его затухания в зависимости от расстояния **t** до точки наблюдения

Процедура численного интегрирования в модели с постклассификацией

Пусть отрезок интегрирования разбивается на n элементарных отрезков равной длины $d = D / n$

$$\exp\left(-\int_0^t \tau(s(\mathbf{x}(t')))\mathrm{d}t'\right) \approx \exp\left(-\sum_{i=0}^{t/d} \tau(s(\mathbf{x}(id)))d\right) \quad (1)$$

$$= \prod_{i=0}^{t/d} \exp(-\tau(s(\mathbf{x}(id)))d) \approx \prod_{i=0}^{t/d} (1-\alpha_i)$$

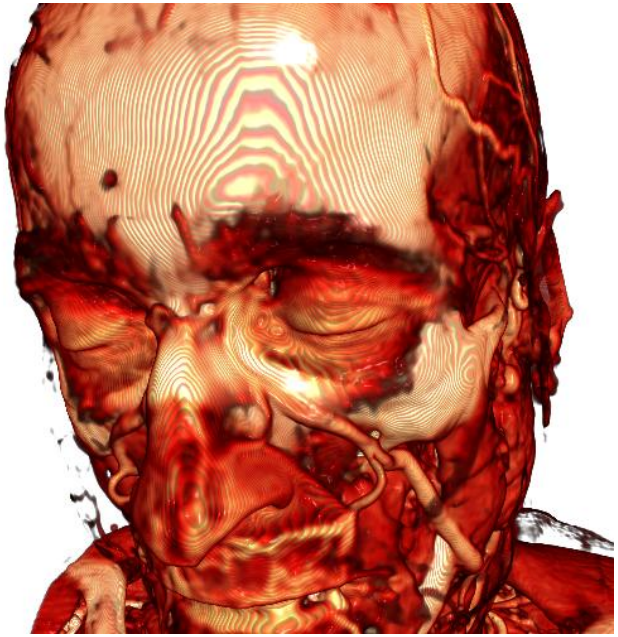
$$\alpha_i = 1 - \exp(-\tau(s(\mathbf{x}(id)))d) \approx \tau(s(\mathbf{x}(id)))d \quad (2)$$

$$I \approx \sum_{i=0}^{n} C_i \cdot \alpha_i \prod_{j=0}^{i-1} (1-\alpha_j) \quad (3)$$

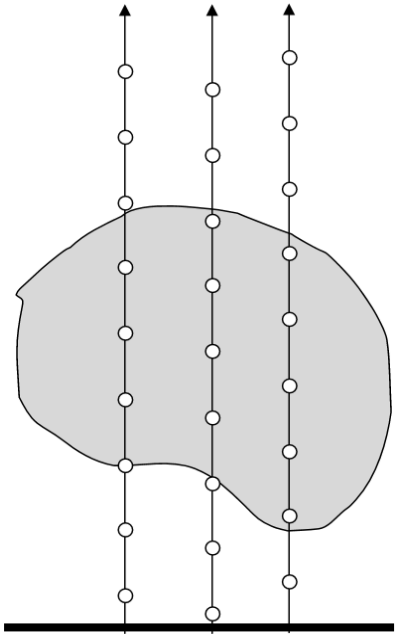
C_i α_i – цвет и непрозрачность i-ого сегмента луча; n – число шагов луча

Случайный сдвиг (*jittering*) стартовых позиций лучей для устранения регулярности дефектов постклассификации

Цвет каждого пикселя изображения рассматривается, как случайная величина.



Старт лучей общей плоскости

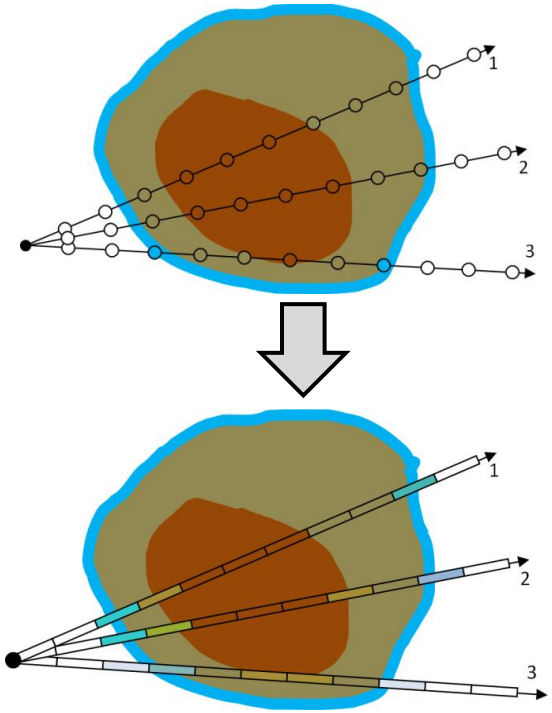


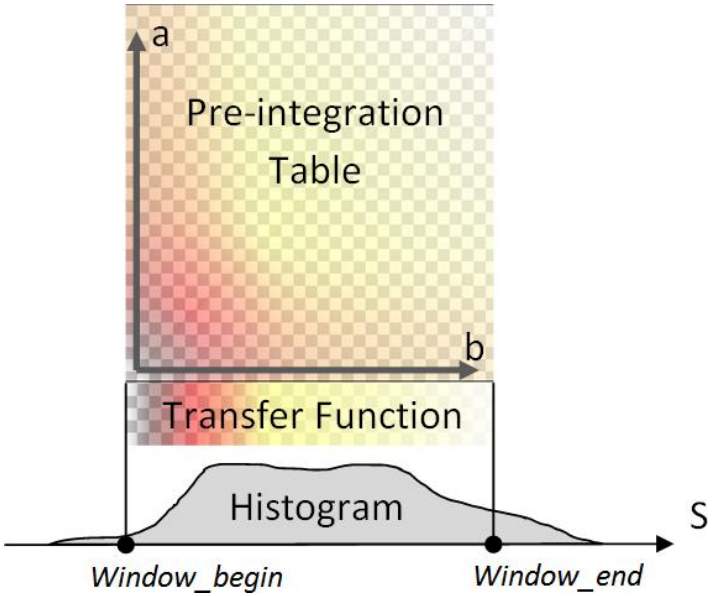
Случайный сдвиг луча (jittering)

Старт со случайным сдвигом

Интегрирование с использованием таблиц, предынтегрированный объёмный рендеринг (PDVR)

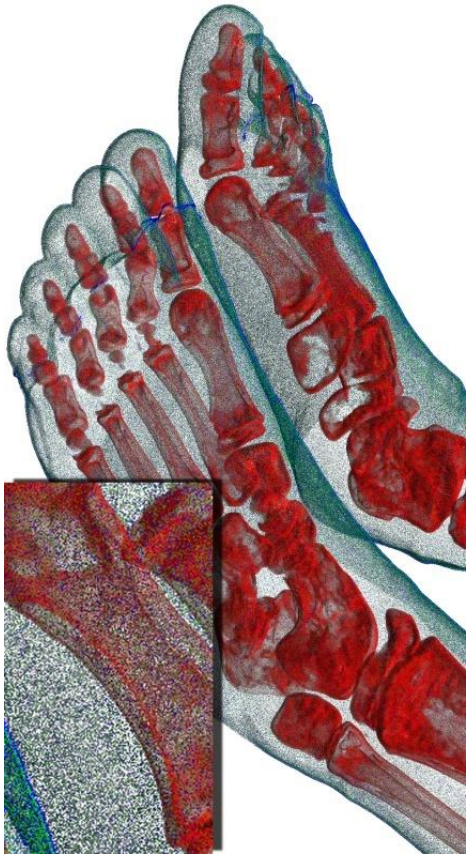
1. При классификации PDVR использует цвета отрезков, а не точек (предынтегрированная классификация)
2. Любой отрезок характеризуется двумя значениями данных – выборки на его концах.
3. Цвета отрезков хранят в предынтегрированной таблице, вычисленной заранее на CPU



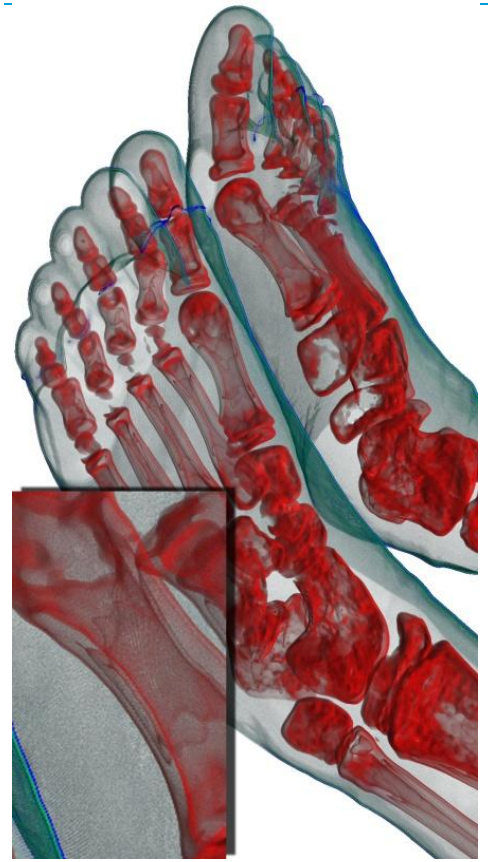


Transfer Function и соответствующая ей предынтегрированная таблица

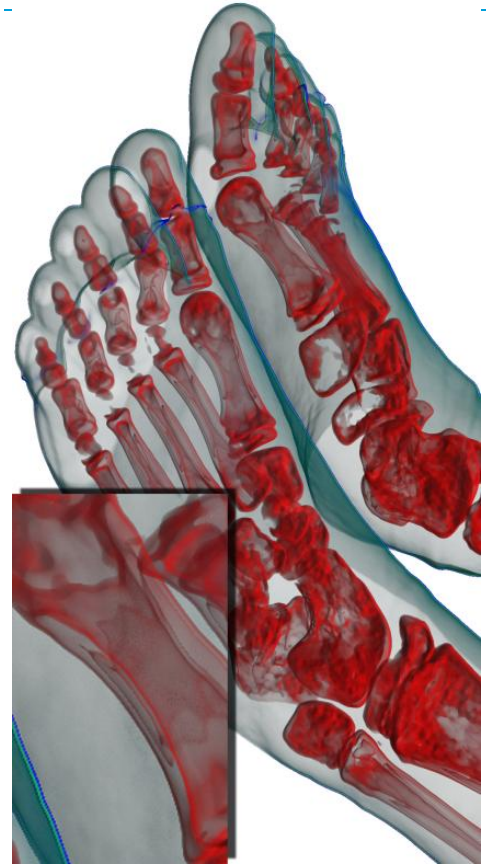
Улучшение качества при использовании PDVR



один кадр обычного рендеринга (UDVR)
PSNR=12dB



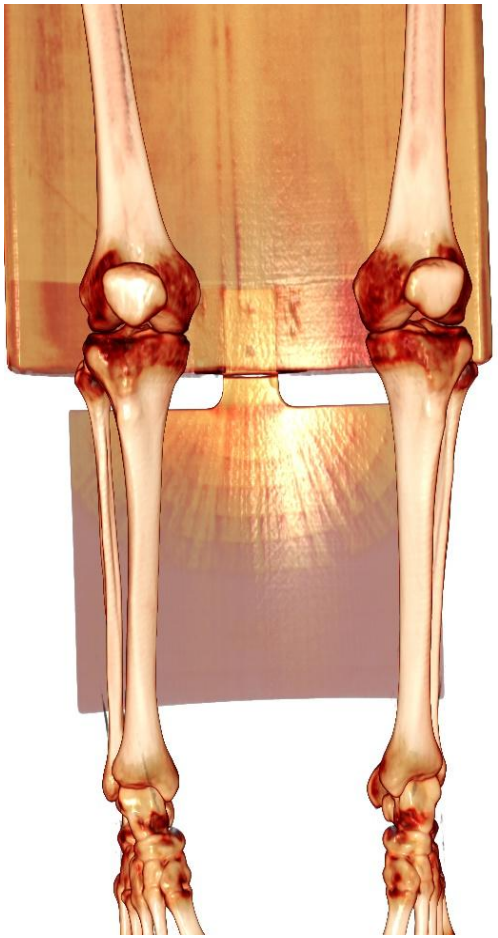
усреднение 60 кадров обычного рендеринга
PSNR=27dB



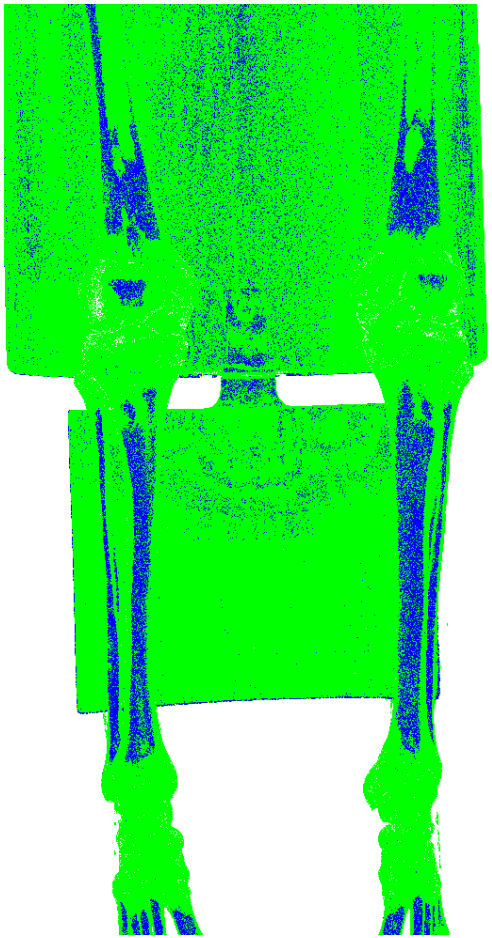
предынтегрированный рендеринг (PDVR)
PSNR=36dB

Частота выборки во всех случаях: 2

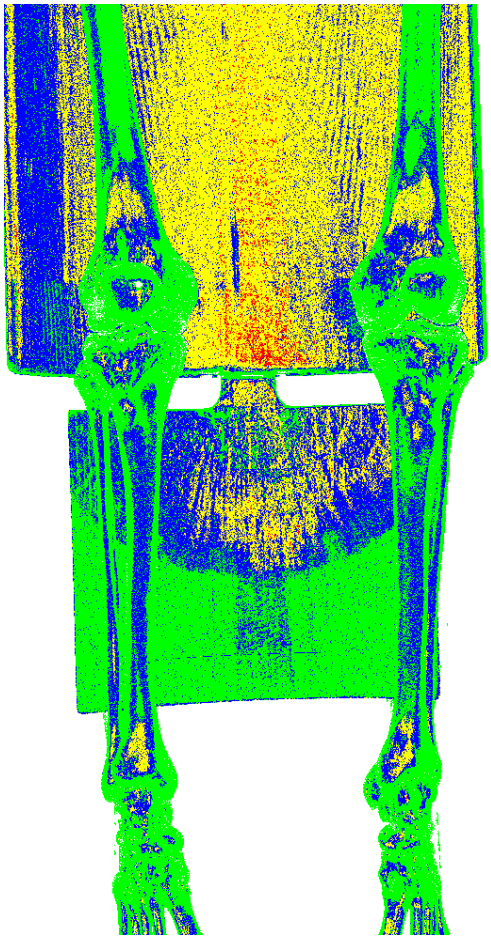
Недостаток предынтегрированого рендеринга: дефекты изображения при использовании локального освещения



Результат рендеринга;

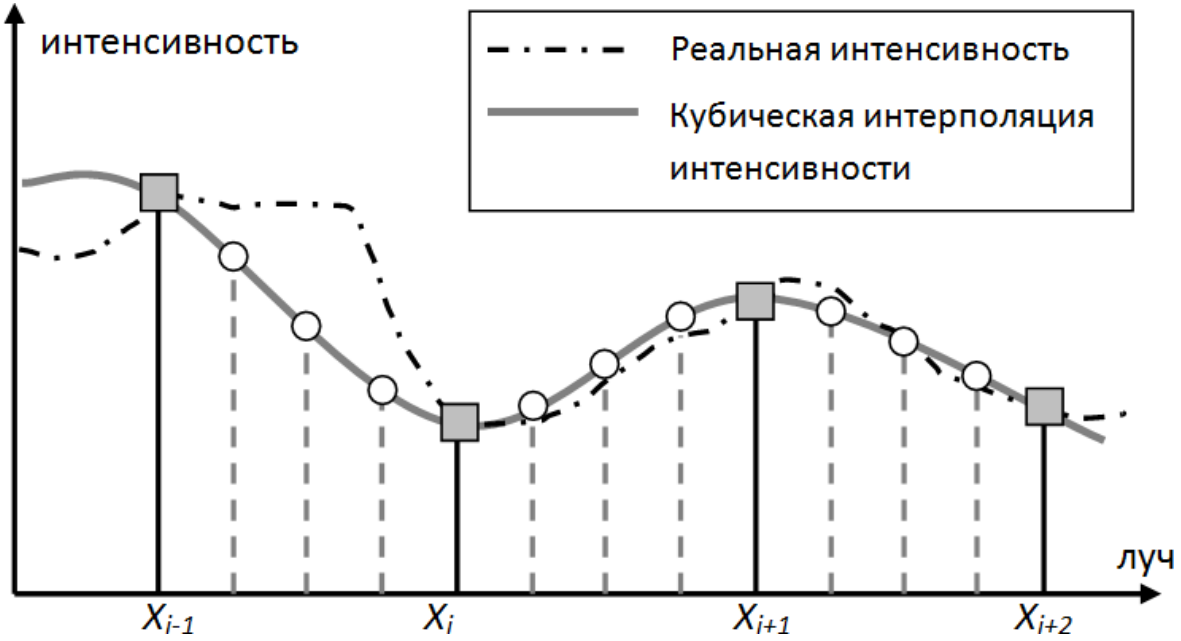


Без освещения
PSNR = 36.4dB



С освещением
PSNR = 22.1dB

Метод виртуальных выборок с предынтегрированием



Известный метод виртуальных выборок доработан заменой постклассификации на предынтегрированную классификацию

Количественная оценка качества визуализации

1) Исследованы оценки качества (дефектов) изображения, принятые в теории сигналов, 2D и 3D сжатии, в оценке качества видео (PSNR, SNR, SSIM)
2) Для задачи количественной оценки дефектов постклассификации в объёмной визуализации предлагается использовать логарифмическую оценку, подобную PSNR

Вывод рендеринга

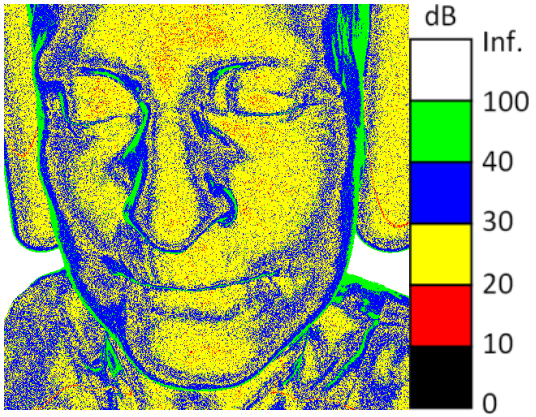
Дисперсия цвета пикселя:

$$D(C(i,j)) = M|C(i,j) - M(C(i,j))|^2 \quad (1)$$

$$MSE = \frac{1}{N}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} D(I(i,j)) \quad (2)$$

$$PSNR = 10log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (3)$$

MSE – средняя квадратичная ошибка пикселя
N – число пикселей, не принадлежащих фону;
m, n – размеры окна вывода;

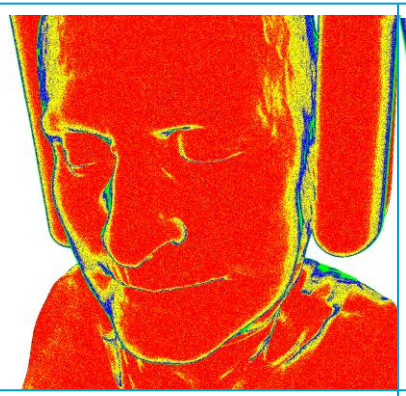
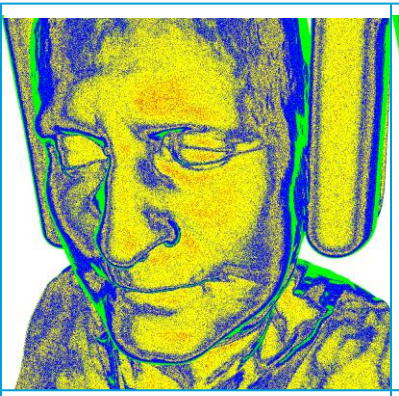
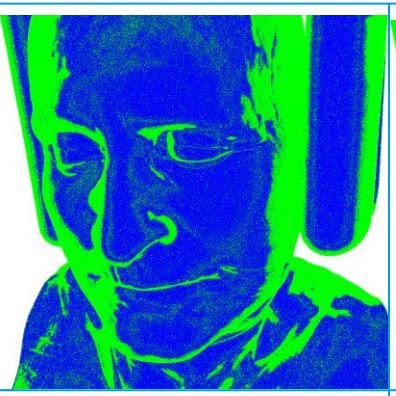
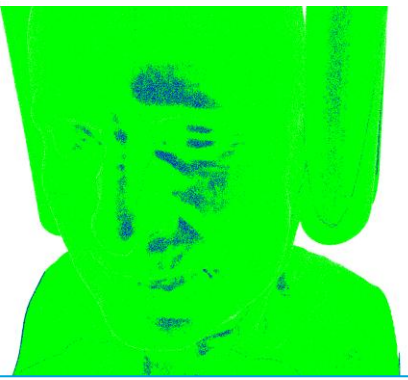


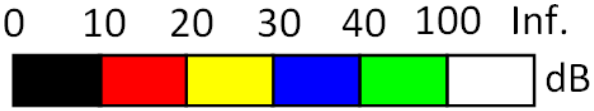
Карта PSNR

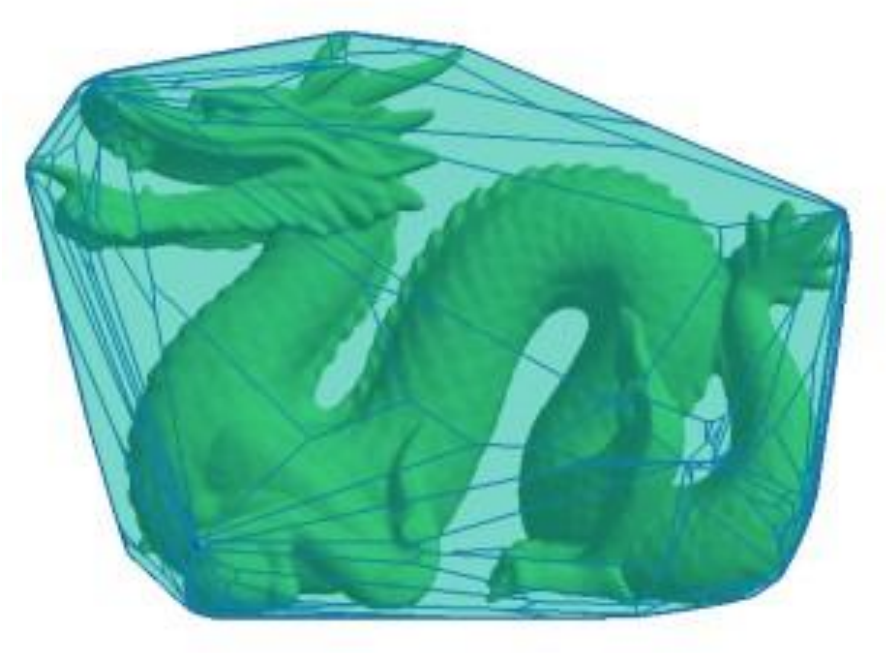
Качество рендеринга при различных частотах выборки

Вне зависимости от метода рендеринга значение PSNR характеризует качество рендеринга:

1) PSNR є [30dB,40dB] – приемлемое качество
2) PSNR > 40dB – улучшение качества практически незаметно
3) PSNR < 30dB – шум сильно заметен

Карта PSNR				
Частота выборки	1.2	2.2	4.3	6.9
PSNR	10dB	20dB	30dB	40dB

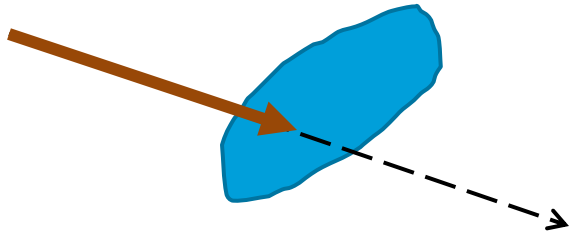




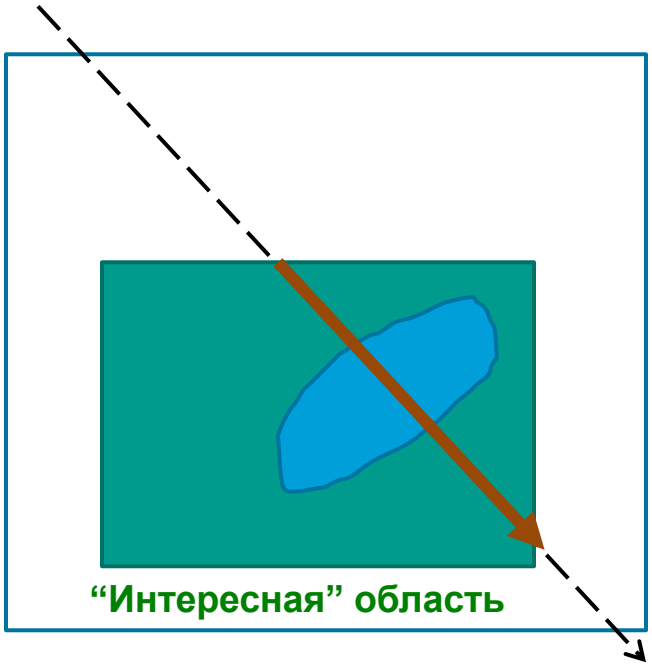
Оптимизационные стратегии

Возможные оптимизации

- Раннее завершение луча
  - Сравнение накопленной непрозрачности с порогом



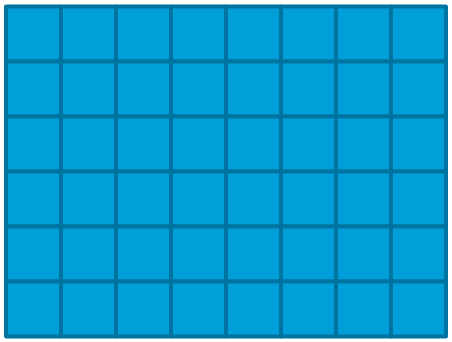
- Игнорирование пустых областей
  - Хранить диапазоны значений блоков ячеек
  - Хранить расстояние до ближайшего непустого вокселя
  - Использовать список примитивов (параллелепипедов, сфер, полупространств), ограничивающих видимые данные



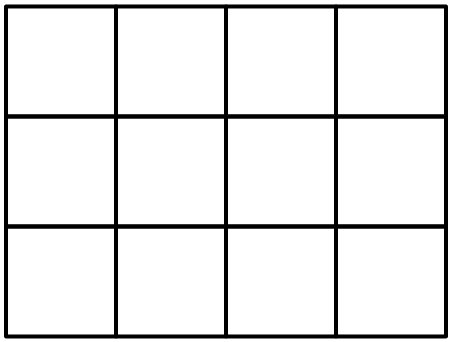
Ускоряющая структура

- Использование в несколько раз меньшей по размеру матрицы
- Элементы этой матрицы – общая оценка исходных данных, покрываемых макроячейкой (максимальное и минимальное значение)

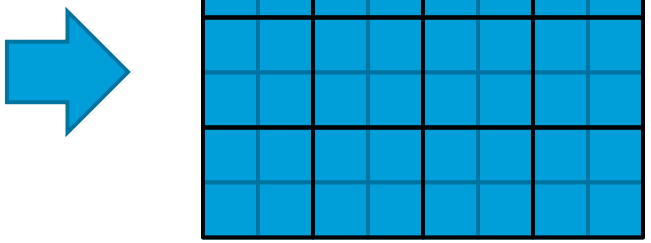
Исходные данные



Ускоряющая структура

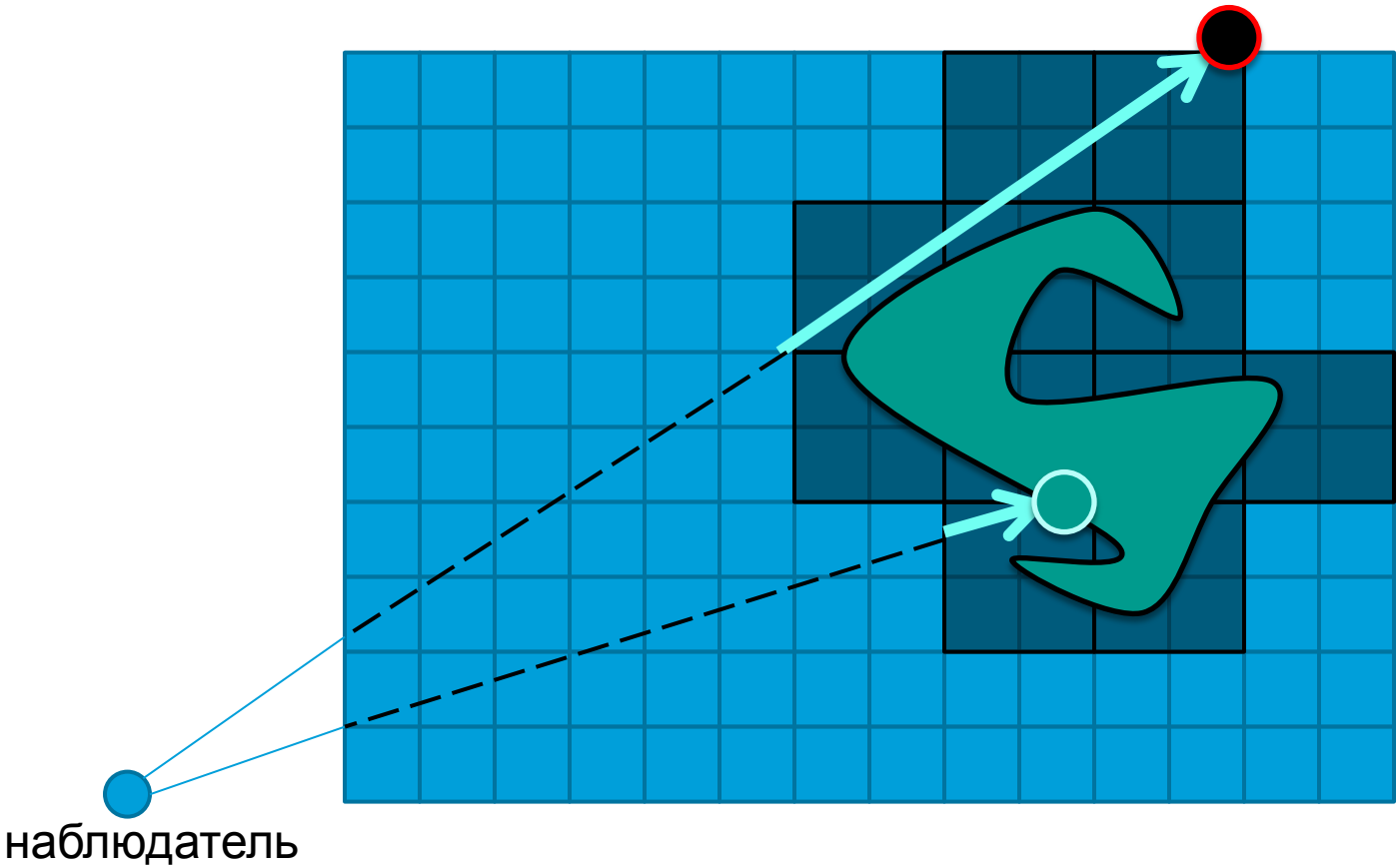


Иерархическая структура



Ускоряющая структура

- Луч пропускает “неинтересные” области



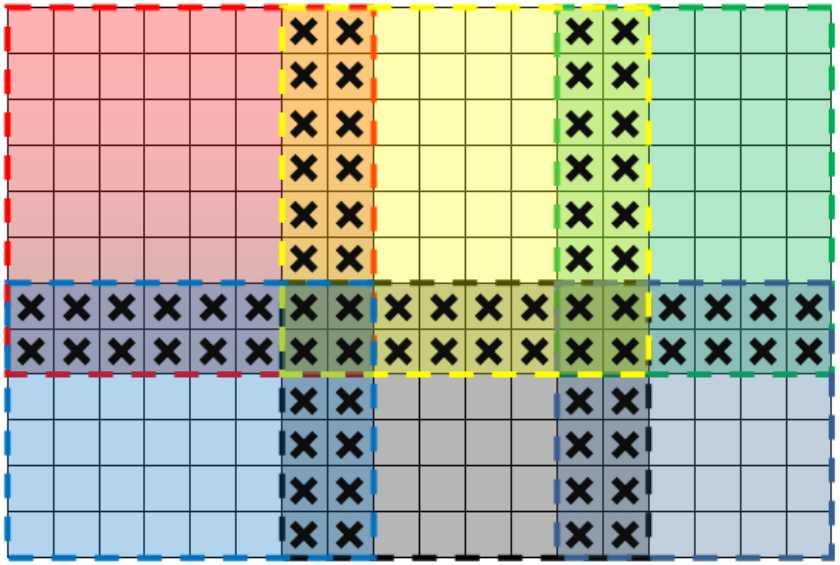
Блочная декомпозиция пространственных данных

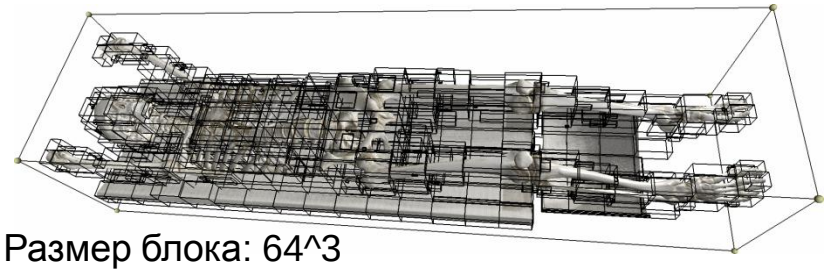
Модификация метода блочной декомпозиции данных в алгоритме Ray Casting:

1) оптимальная последовательность обхода блоков;

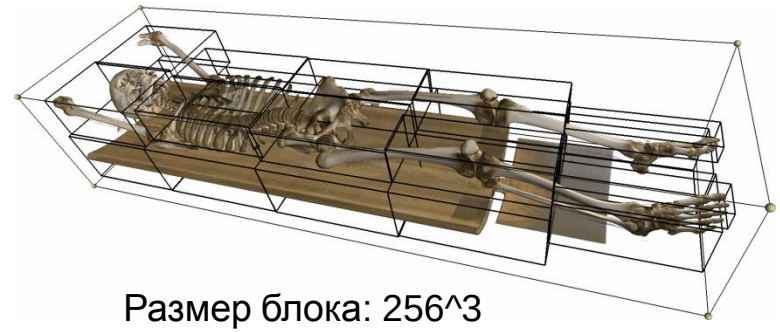
2) использованы оптимизированные по объему вспомогательные структуры метода пропуска пустых областей;

3) обеспечение возможности построения локального освещения и теней



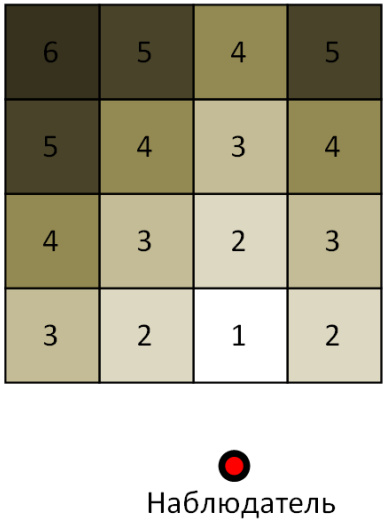


Размер блока: 64^3

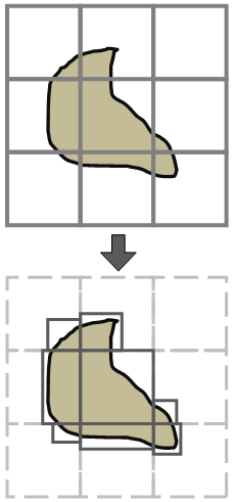


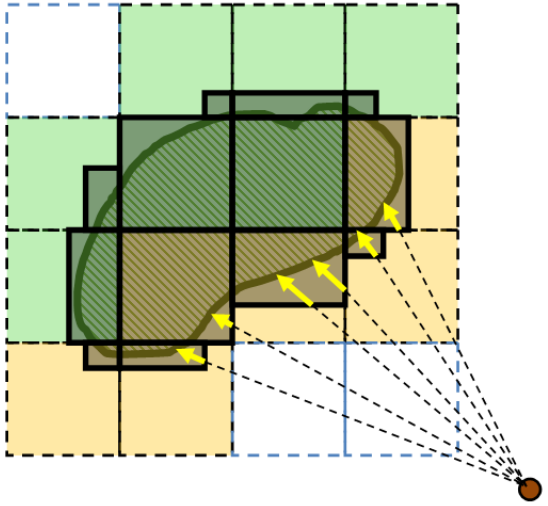
Размер блока: 256^3

Сокращение трассируемого объёма подгонкой ограничивающих блоки боксов



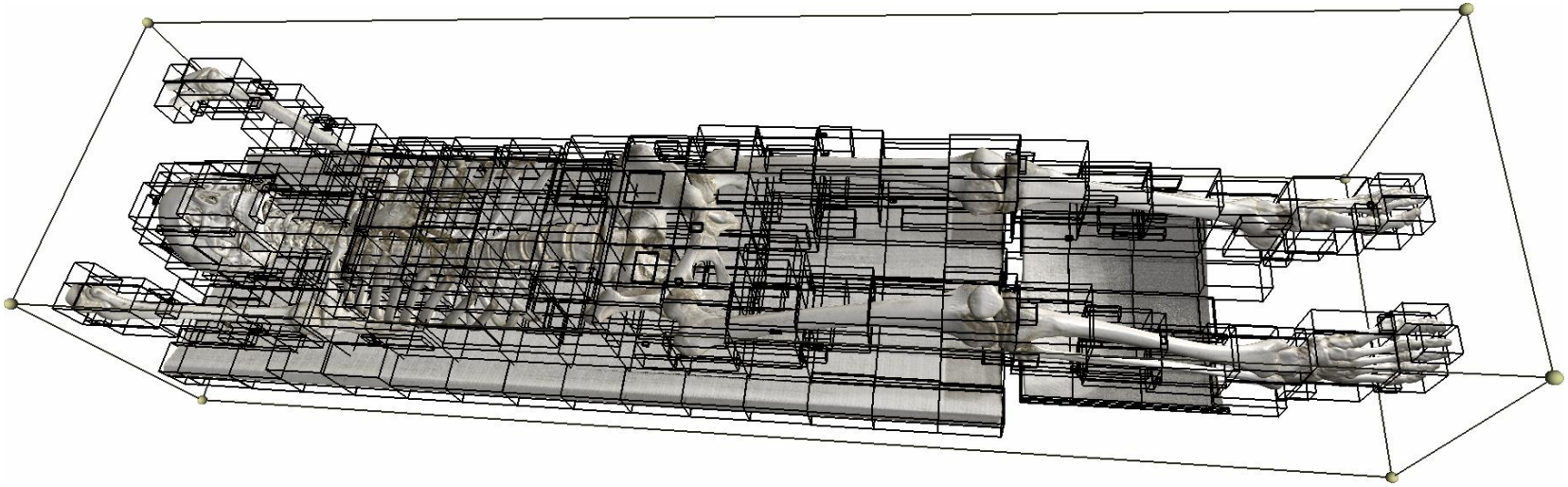
Оптимальная последовательность обхода блоков во время рендеринга



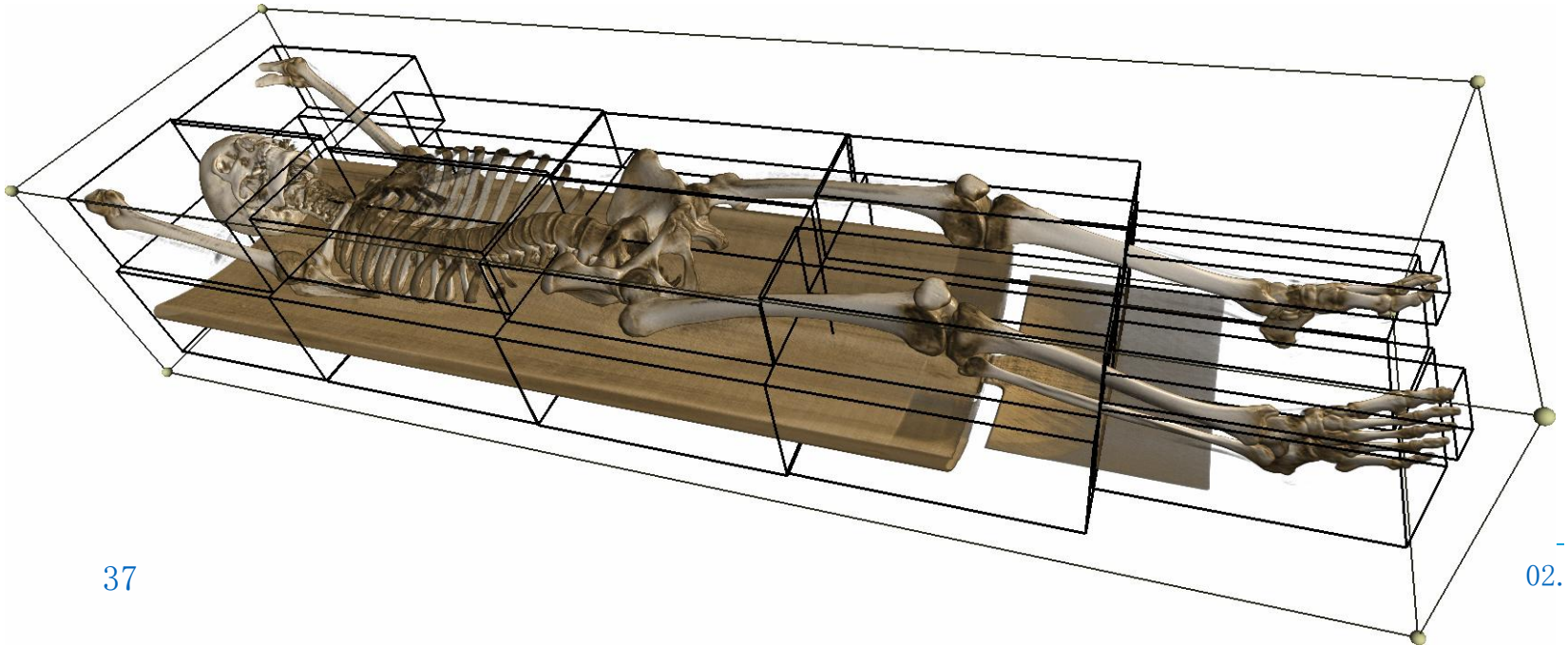


Для каждого блока данных заводится ограничивающий параллелепипед, который автоматически подгоняется под визуализируемые данные

Подгонка ограничивающих боксов



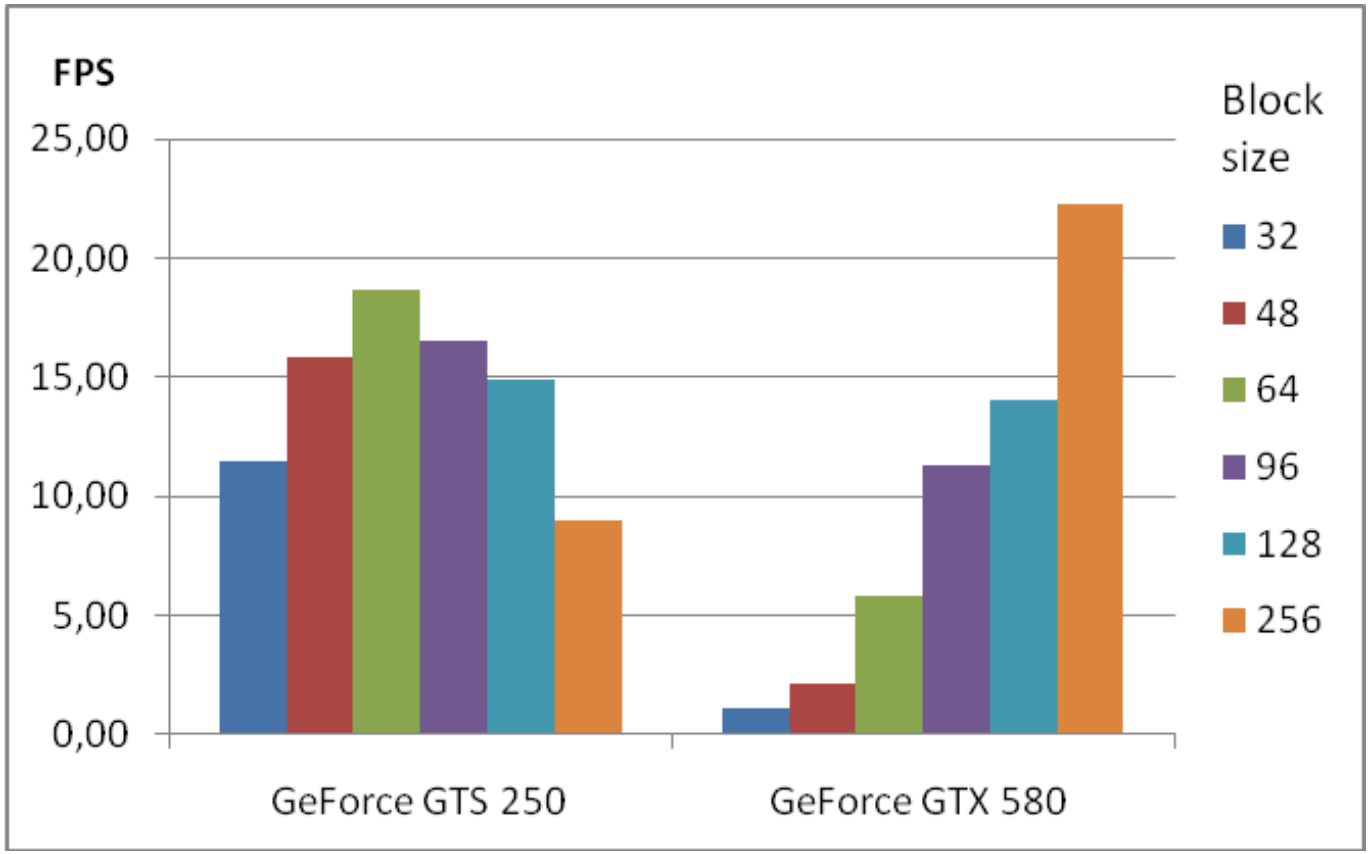
64^3



256^3

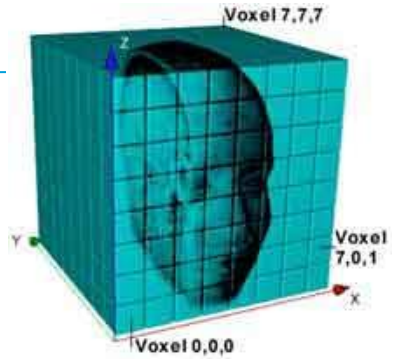
Производительность рендеринга при различных размерах блоков в зависимости от GPU

Для GPU разных классов оптимальны разные размеры блока разбиения



Вопросы реализации объёмного рендеринга на GPU

Инициализация данных



Объёмные данные (в 3D текстуре)

Программа трассировки луча (в виде шейдерной программы)

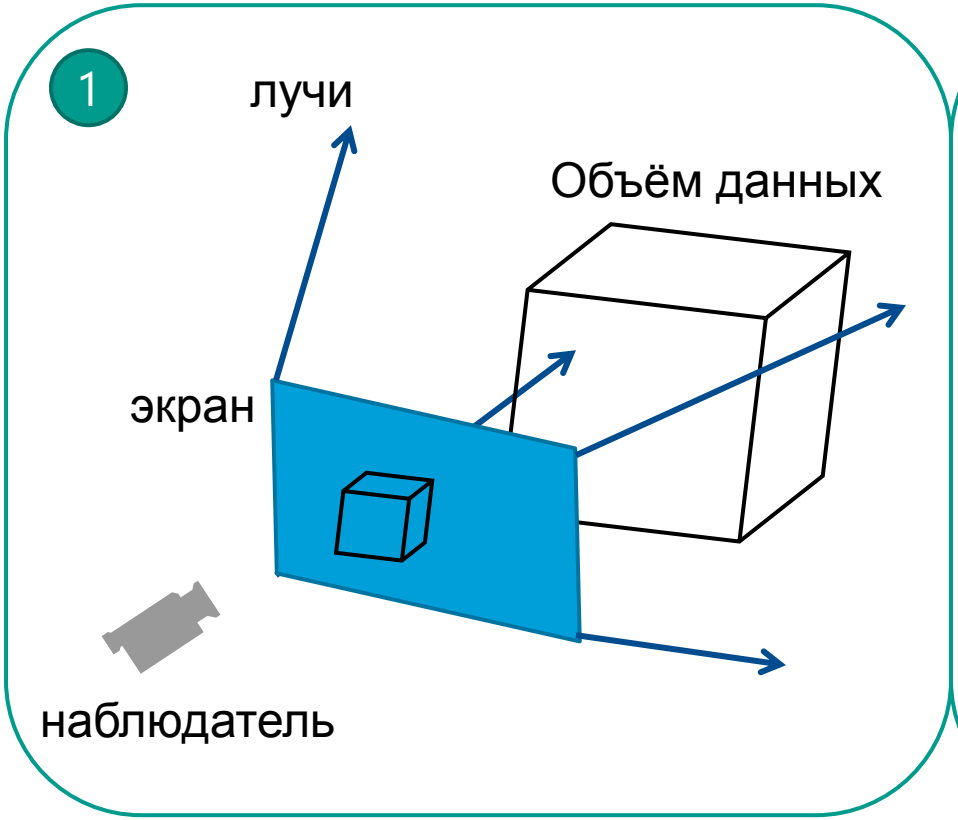
При смене параметров

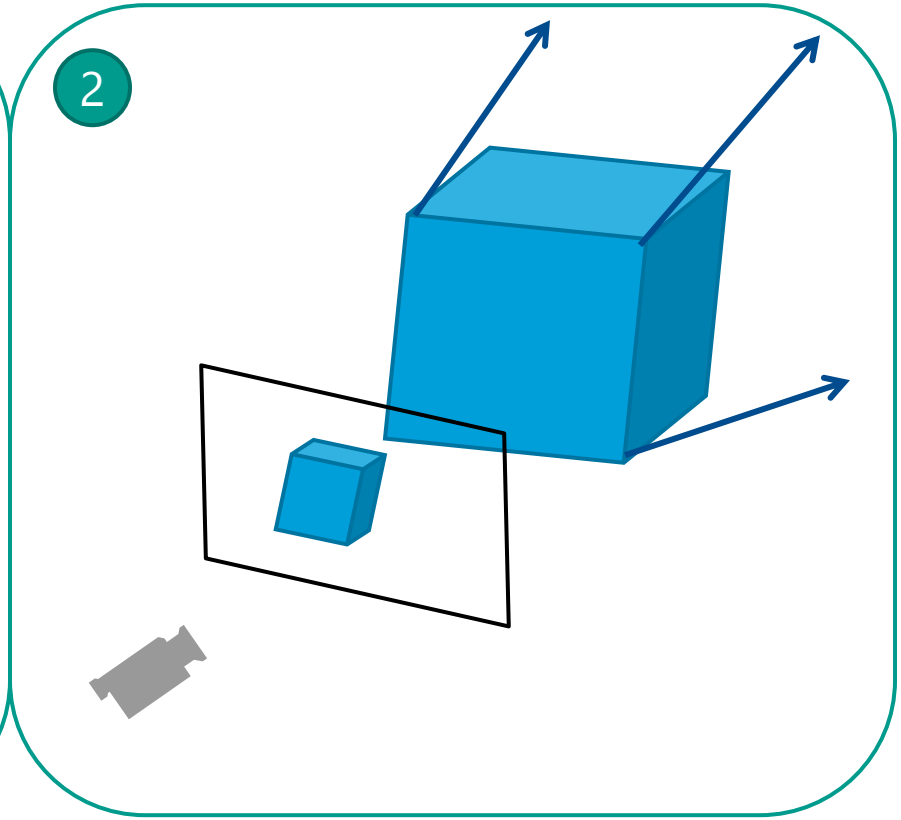
Параметры просмотра (в виде uniform-переменных для шейдерной программы)

GPU

Генерация лучей

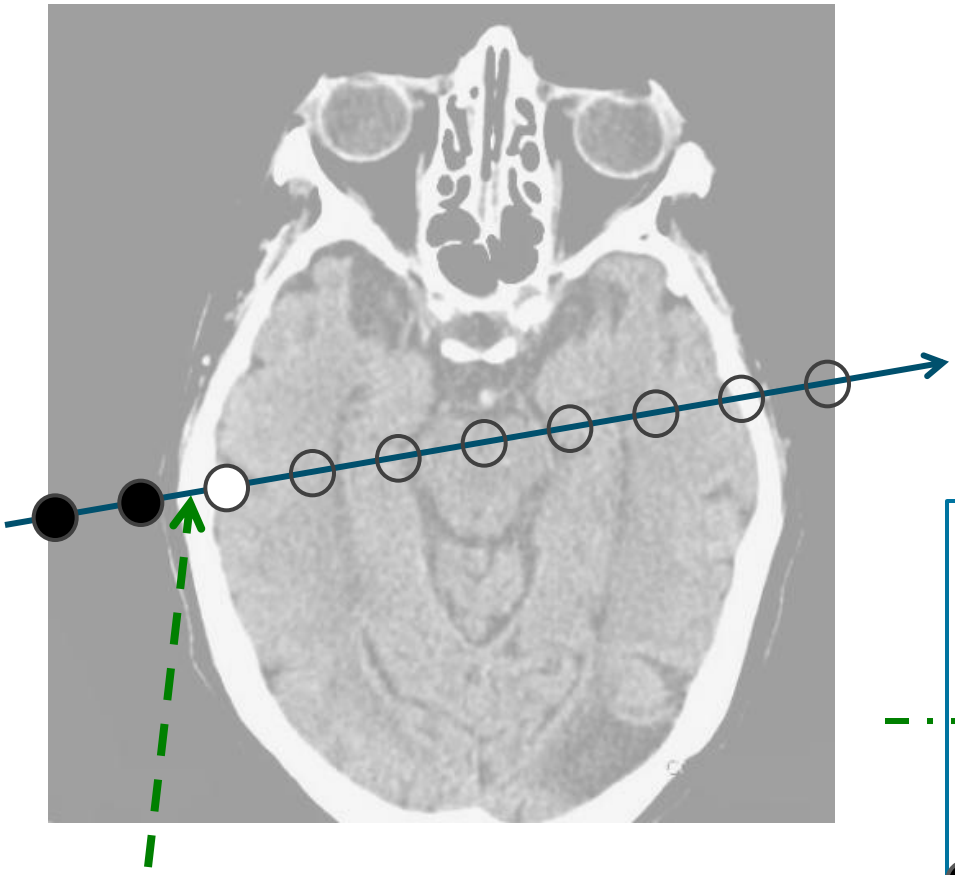
- Шейдерную программу трассировки луча делаем текущей
- В OpenGL рисуем прямоугольное окно для генерации лучей с шагом по x,y определяемым из формата окна в пикселах.



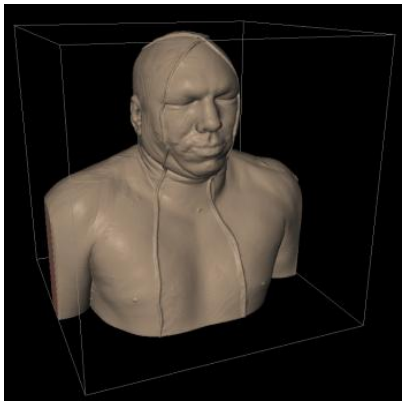


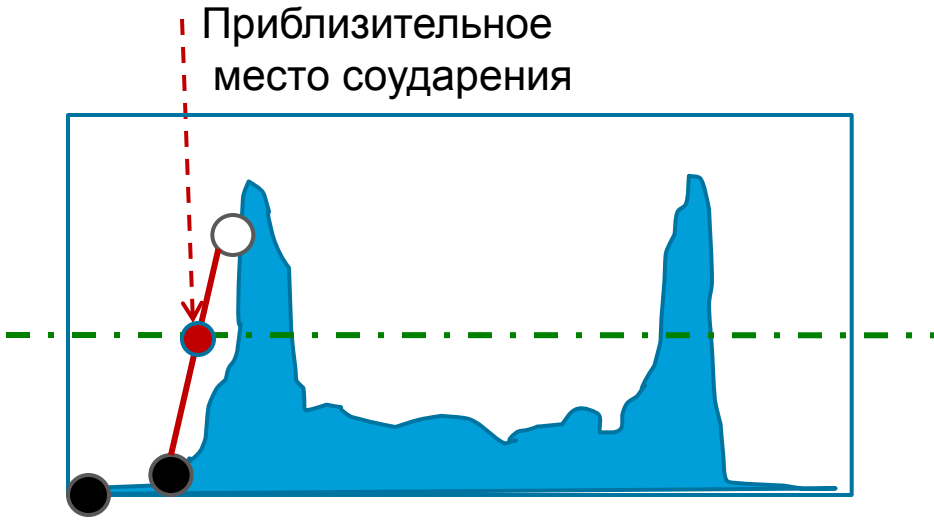
Различные алгоритмы накопления цвета вдоль луча

Изоповерхности



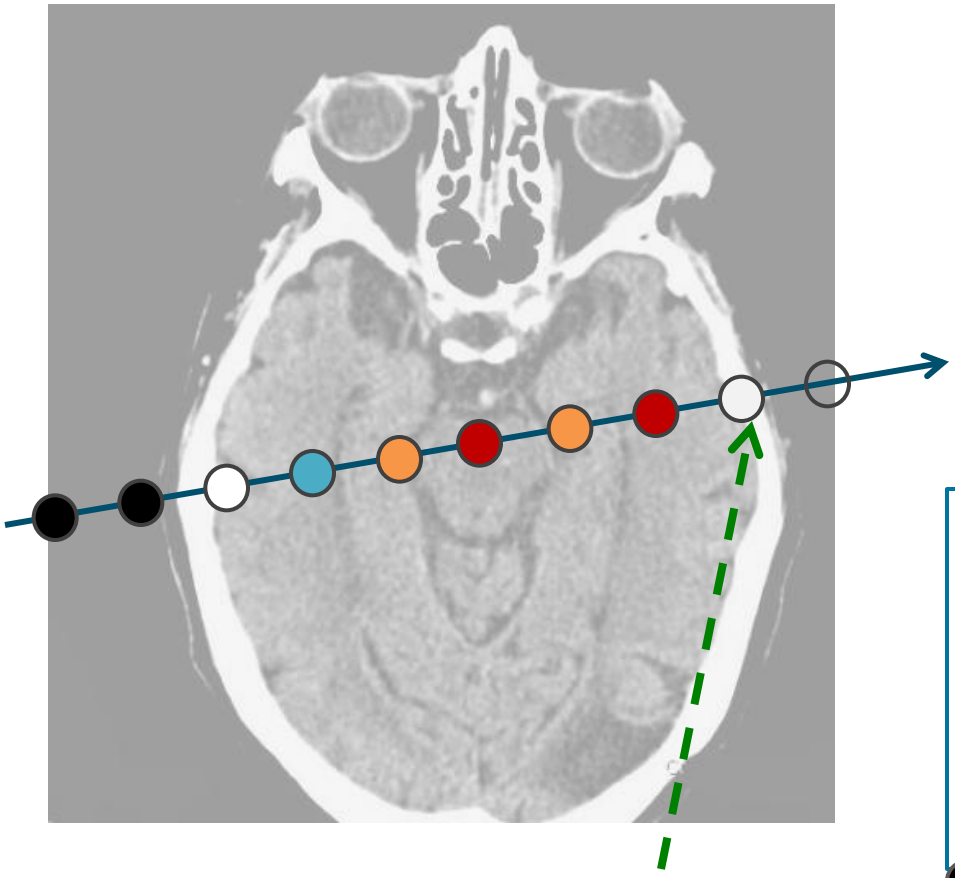
Отрезок, пересекающий изоповерхность



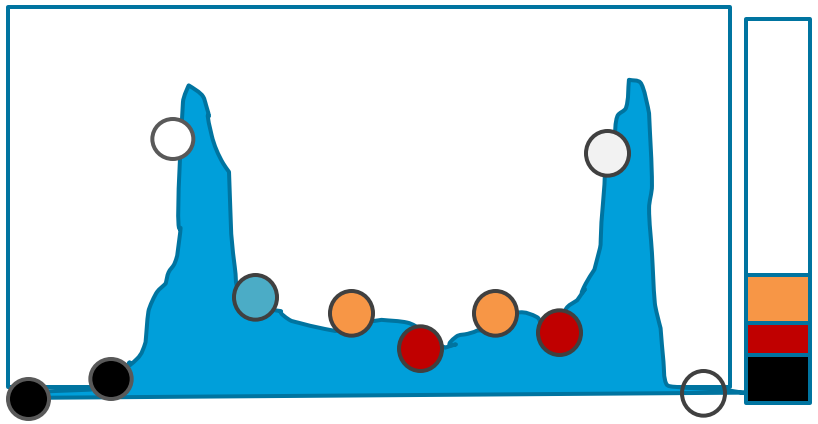


Значения вдоль луча

Полупрозрачные среды

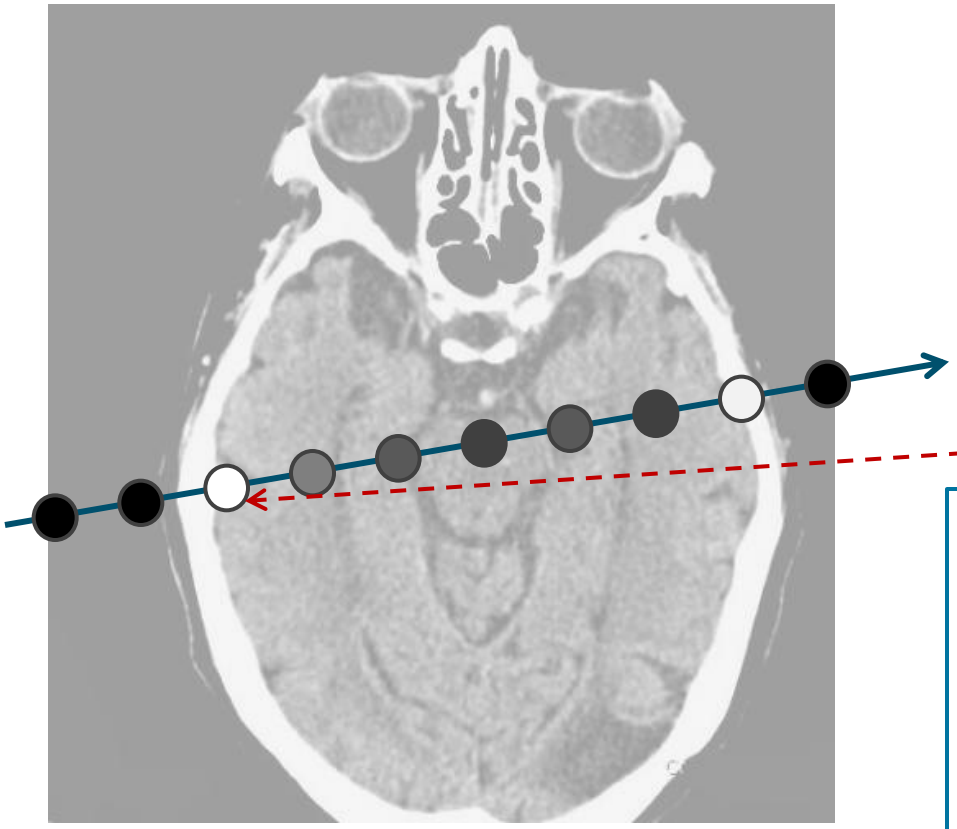


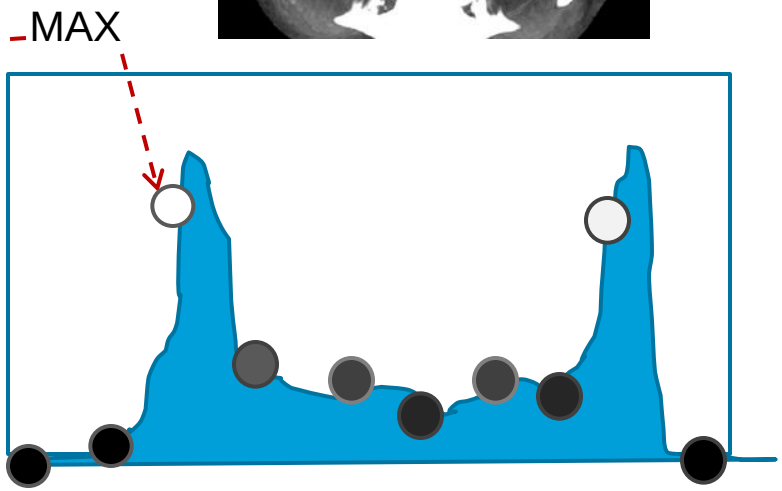
Накопление достаточной непрозрачности



Значения вдоль луча

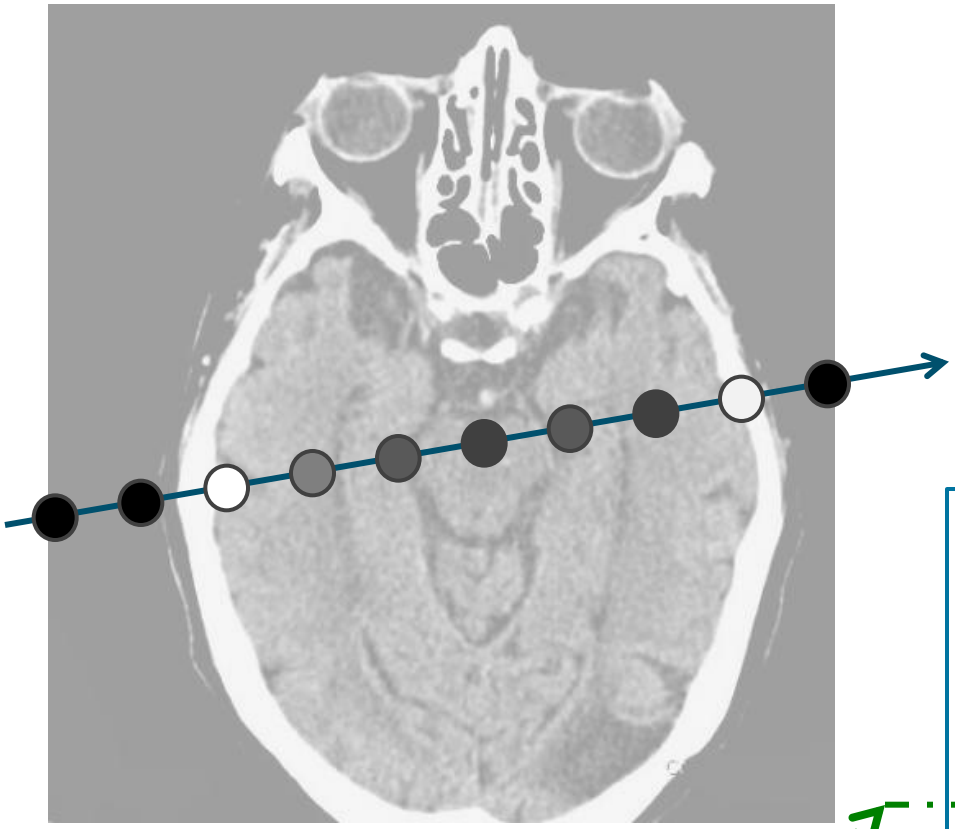
MIP -проекция максимальной интенсивности

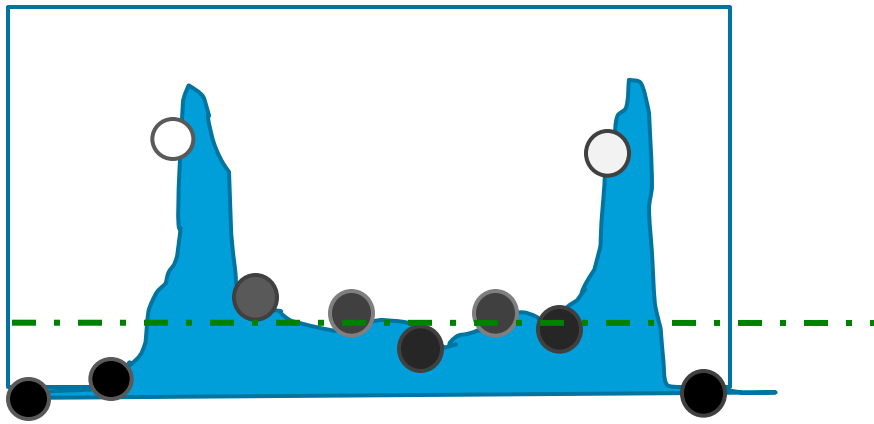




Значения вдоль луча

X-Ray style (среднее значение, квазирентген)

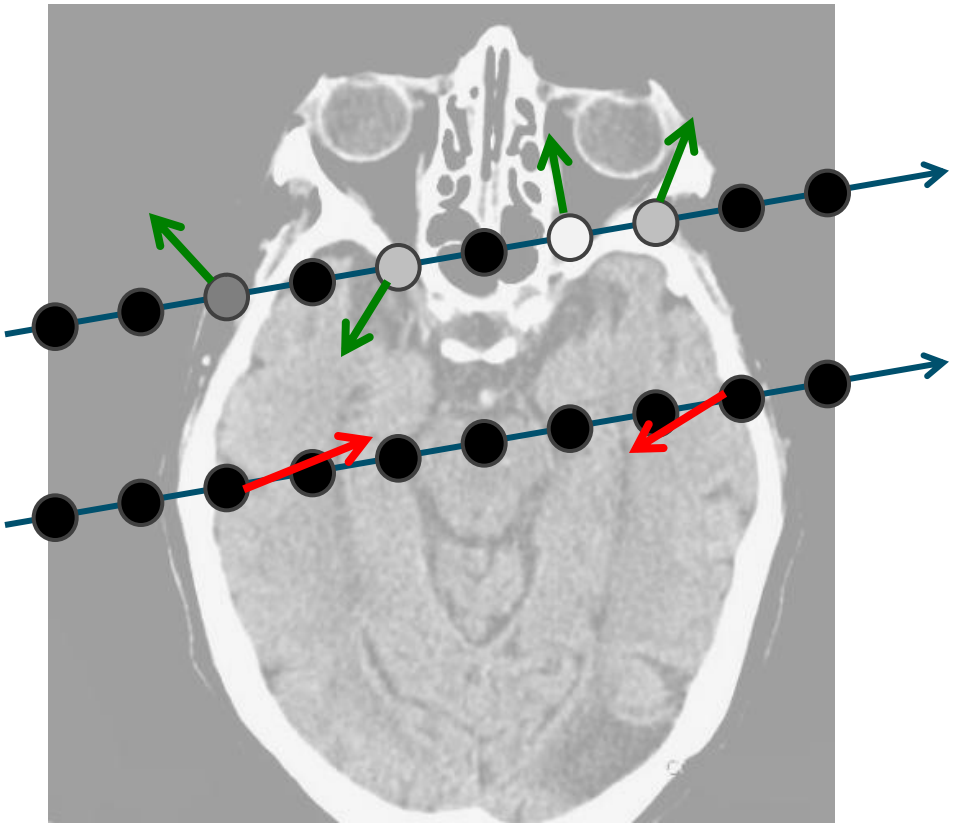


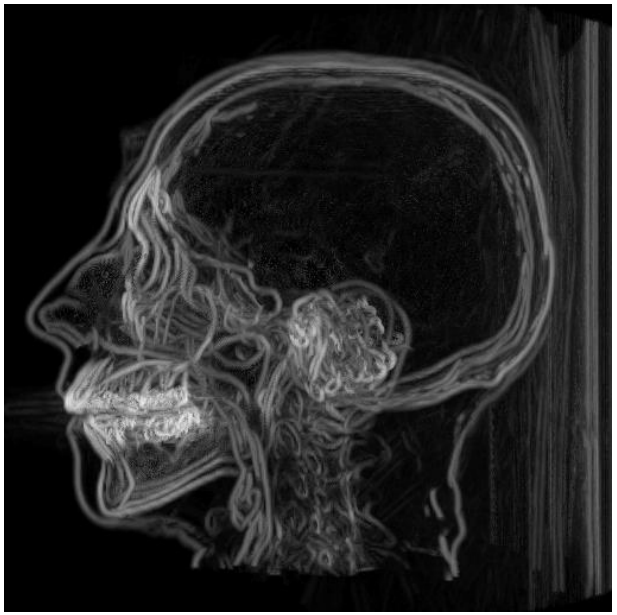


среднее

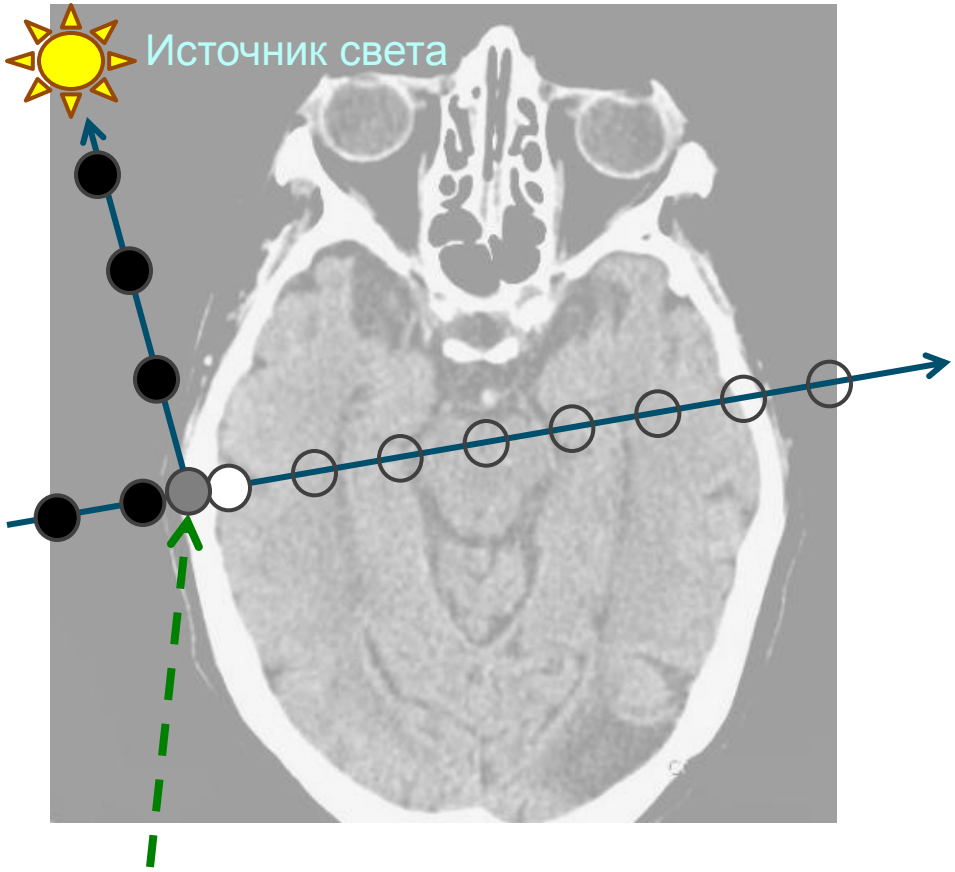
Значения вдоль луча

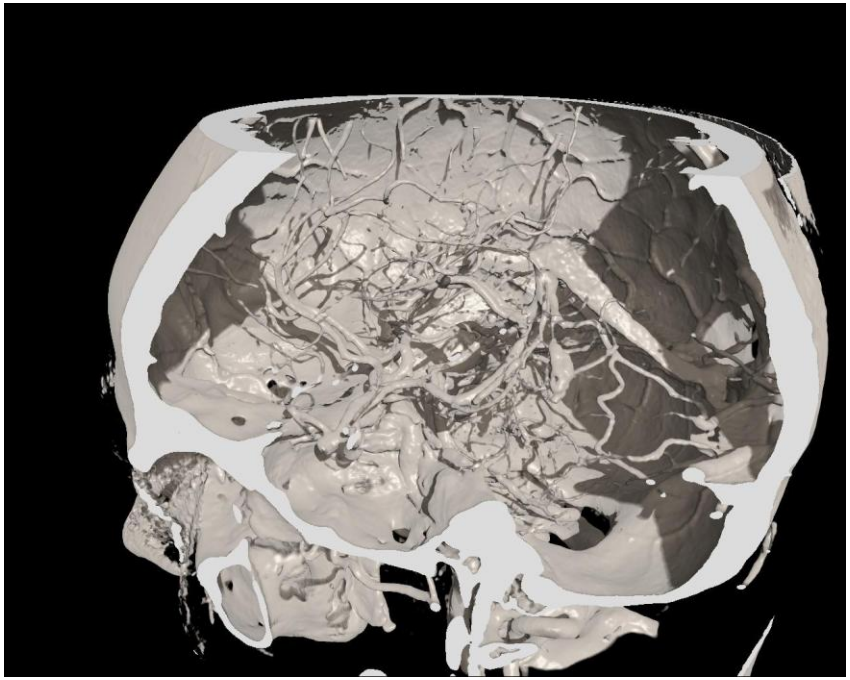
Контуры



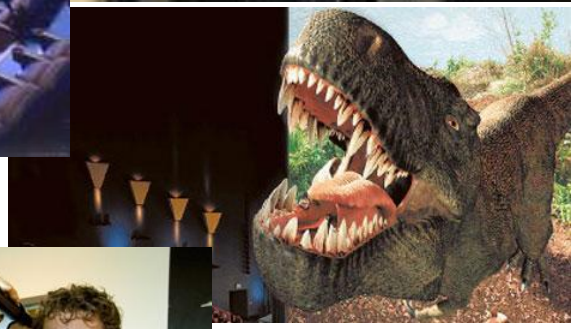


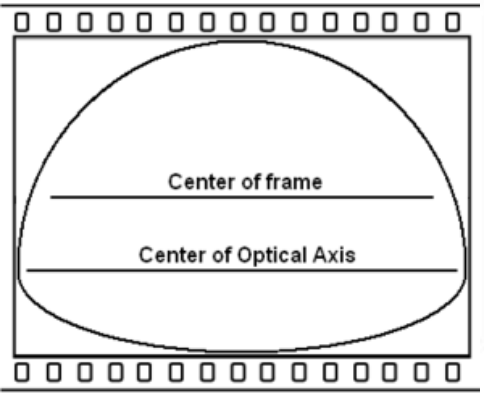
Изоповерхность с тенью



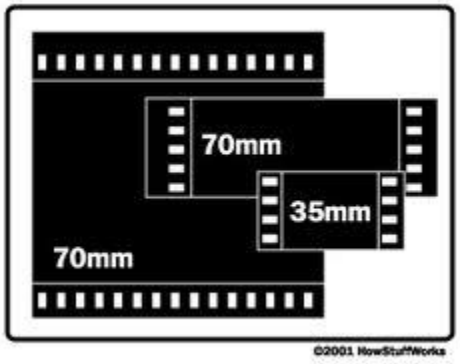


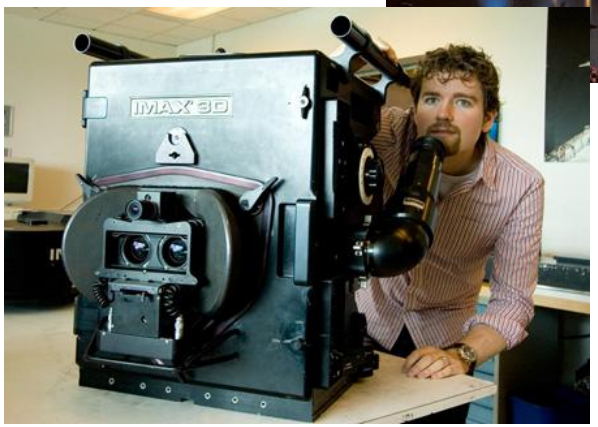
Новая виртуальная реальность





230%

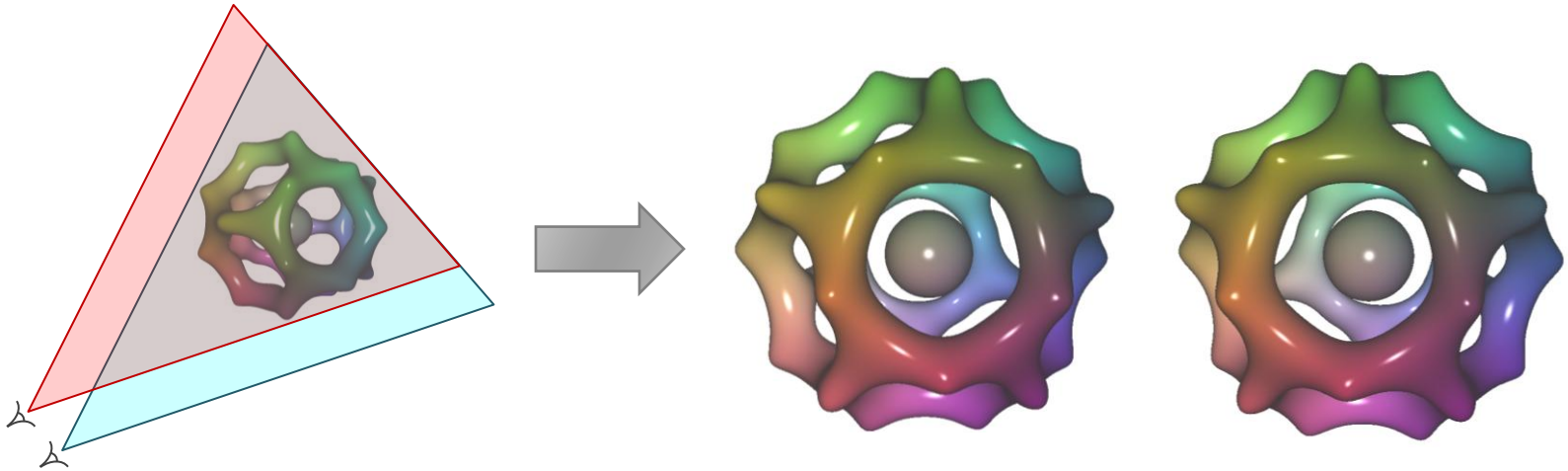




10000x7000px
22[м]x16[м]

Стереоизображение

- *Стереоизображение* – картина или видеоряд, использующий *два* отдельных изображения, позволяющих достичь *стереоэффекта*



- *Стереоэффект* (зрительный) – ощущение протяжённости пространства и рельефности, возникающие при рассматривании реальных объектов, *стереопар*, *стереофотографий*, *стереоизображений* и *голограмм*

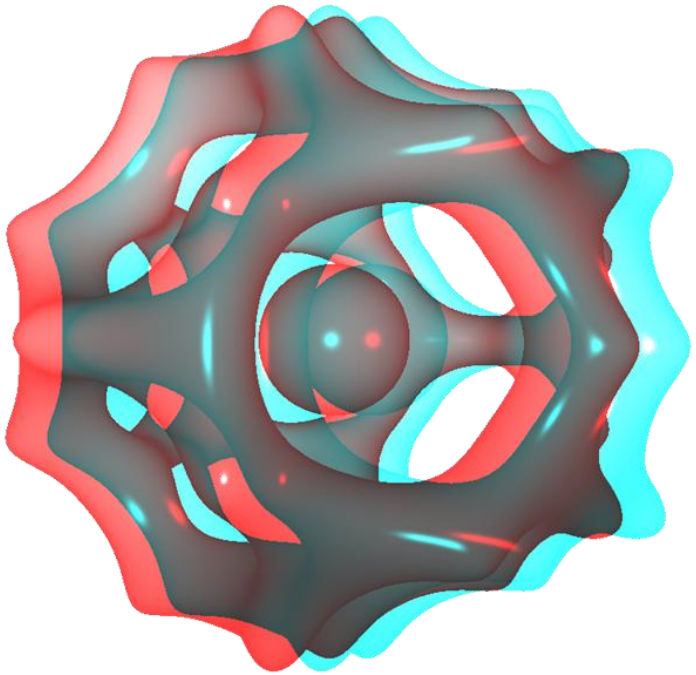
Анаглифные очки…

- *Анаглифные очки* – это очки с цветными стеклами, вместо линз у которых вставлены *светофильтры* противоположных на цветовом круге цветов

- Дешёвый но достаточно эффективный метод, физически он *не обеспечивает правильную передачу цветного стереоизображения*, однако нервная система довольно хорошо интерпретирует его (наилучший эффект достигается для изображений в оттенках серого)
- Время адаптации около 30 секунд, после длительного использования на пропорциональный период нарушается цветовосприятие

Сравнение различных анаглифов...

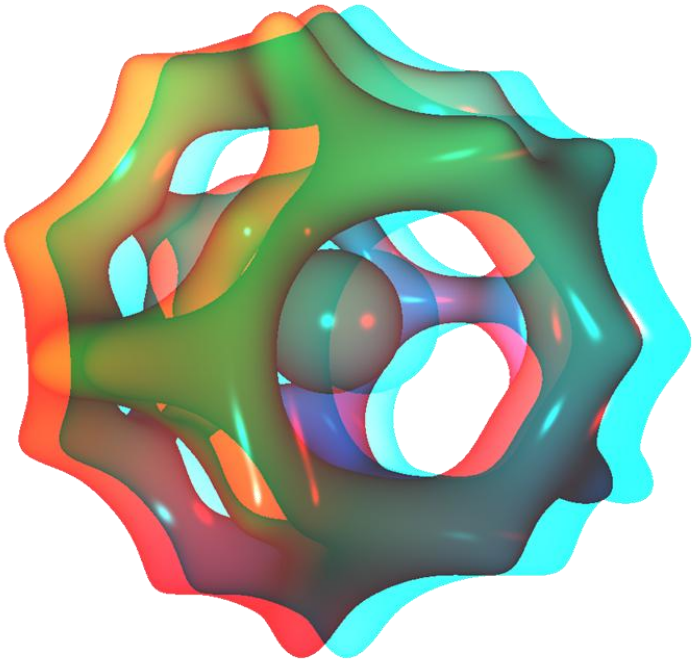
- *Gray Anaglyph*



$$\begin{bmatrix} r_a \\ g_a \\ b_a \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} r_1 \\ g_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0.299 & 0.587 & 0.114 \\ 0.299 & 0.587 & 0.114 \end{bmatrix} \times \begin{bmatrix} r_2 \\ g_2 \\ b_2 \end{bmatrix}$$

Сравнение различных анаглифов...

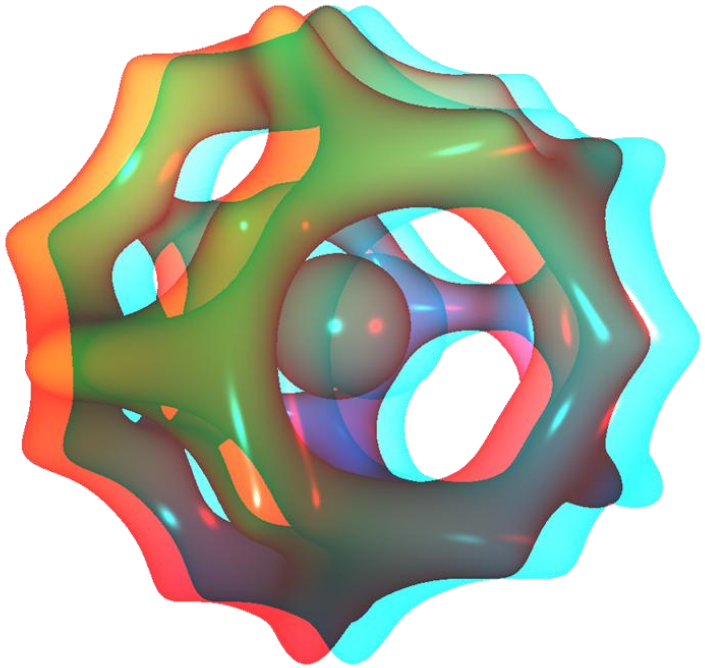
- *Color Anaglyph*



$$\begin{bmatrix} r_a \\ g_a \\ b_a \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} r_1 \\ g_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_2 \\ g_2 \\ b_2 \end{bmatrix}$$

Сравнение различных анаглифов...

- *Half Color Anaglyph*



$$\begin{bmatrix} r_a \\ g_a \\ b_a \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} r_1 \\ g_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_2 \\ g_2 \\ b_2 \end{bmatrix}$$

Использование поляризационных очков...

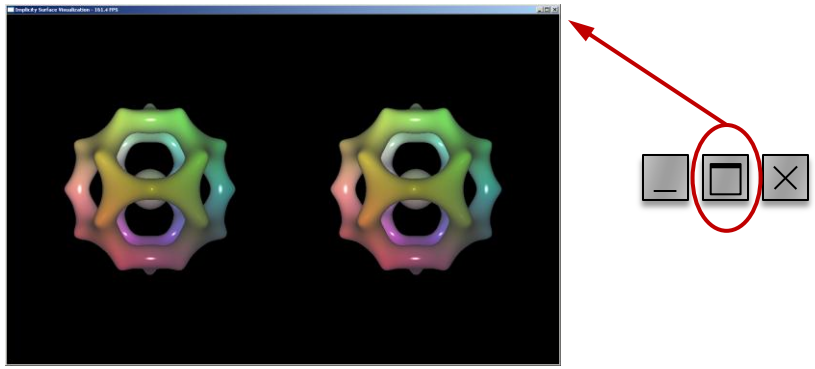
- Простейший способ использования двух проекторов (с поляризацией световых пучков и ***металлизированным*** экраном):
  - Разделить окно визуализации на две равные части
  - Произвести рендеринг сцены с различных ракурсов, и вывести результат в соответствующие части
  - Масштабировать окно на объединенный рабочий стол
- Для масштабирования окна на два разных проектора (монитора):
  - Воспользоваться специальными настройками драйвера
  - Воспользоваться универсальной утилитой *Virtual Screen Maximizer* [*http://www.ghacks.net/2008/12/07/virtual-screen-maximizer*]

↳ *Каждый проектор (или экран) получит уникальное изображение*

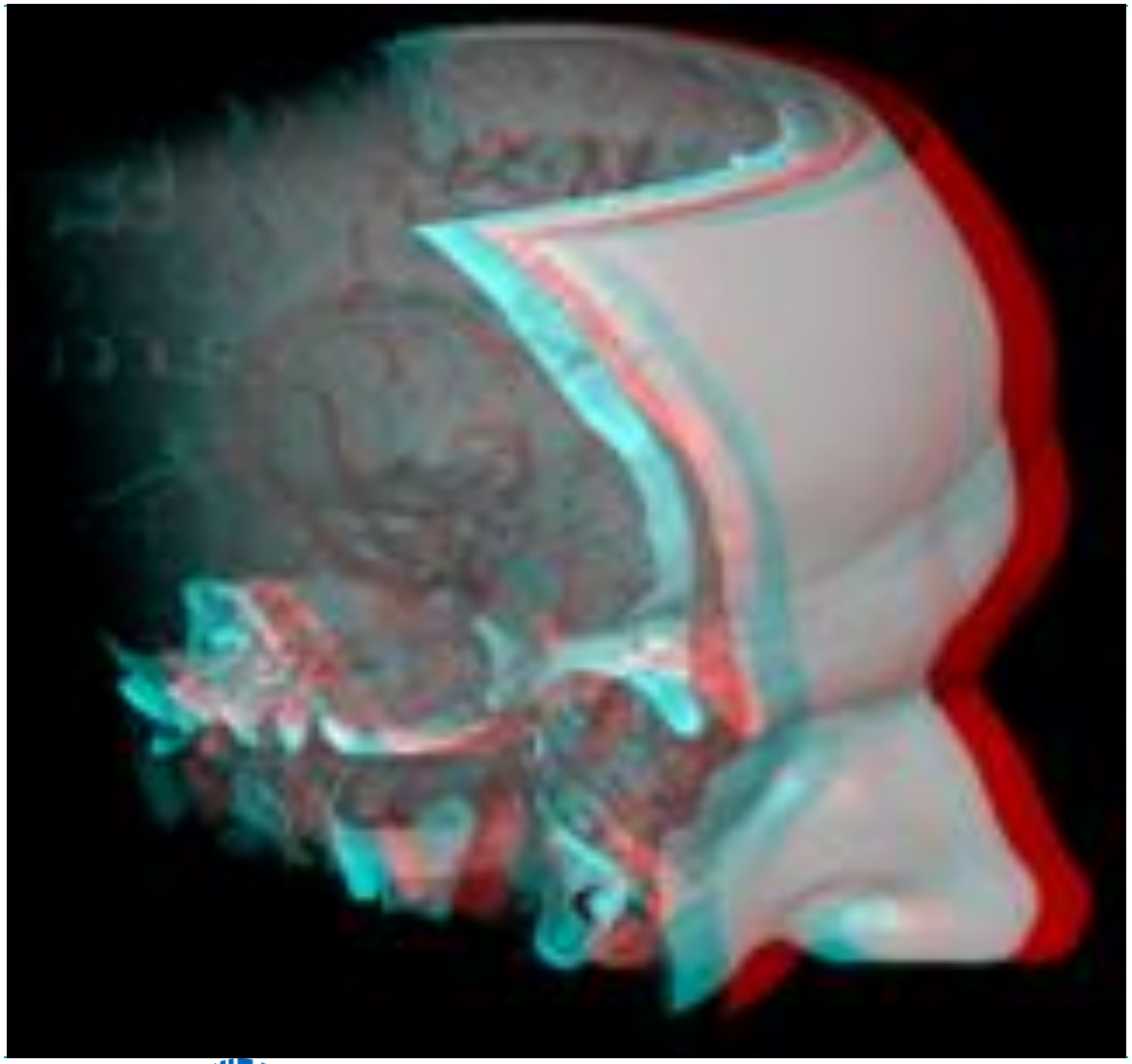
Использование поляризационных очков...

```
glViewport ( 0, 0, W/2, H )
// Moving Camera to Left Eye
// Rendering Scene

glViewport ( W/2, 0, W, H )
// Moving Camera to Right Eye
// Rendering Scene
```



Стереовизуализация в медицине…



Анаглифные очки…