Инженерная и компьютерная графика
6 семестр (диф.зачет)

Лектор:

Таранцев Игорь Геннадьевич

Доцент ФИТ НГУ, ИАиЭ, «СофтЛаб-НСК»

Создатели курса:

Дебелов Виктор Алексеевич

Валеев Тагир Фаридович

Козлов Дмитрий Сергеевич

Лекция №4

Volume Rendering

Volume rendering

Volume rendering (VR) = визуализация объемов. По-русски – визуализация объемных плотностей (ОП), то есть визуализация среды, взаимодействующей со светом. Применение метода VR – графический анализ скалярной функции:

$$f(r),\ \ r \in D \subset R^3$$

Volume rendering

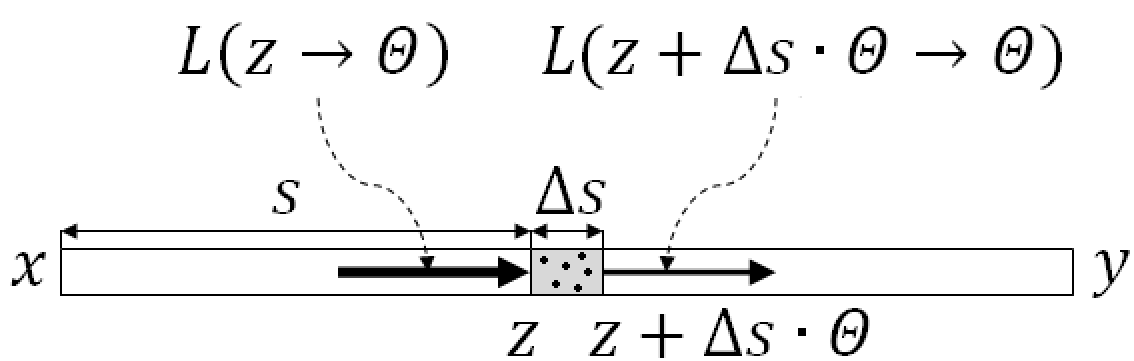
Как бы мы не получали исходные данные о функции f (на прямоугольных или хаотических сетках или по аналитическому выражению), мы считаем, что мы знаем $f(r)$ в любой точке $r \in D \subset R^3$

Основная гипотеза

Полагается, что объемная плотность (**ОП**) – это "участвующая/реагирующая" среда. Она обладает оптическими свойствами – поглощение, эмиссия, рассеивание, которые влияют на прохождение и/или пропускание луча (света, X-лучи, ...). Например, среда состоит из капелек воды или из отдельных молекул.

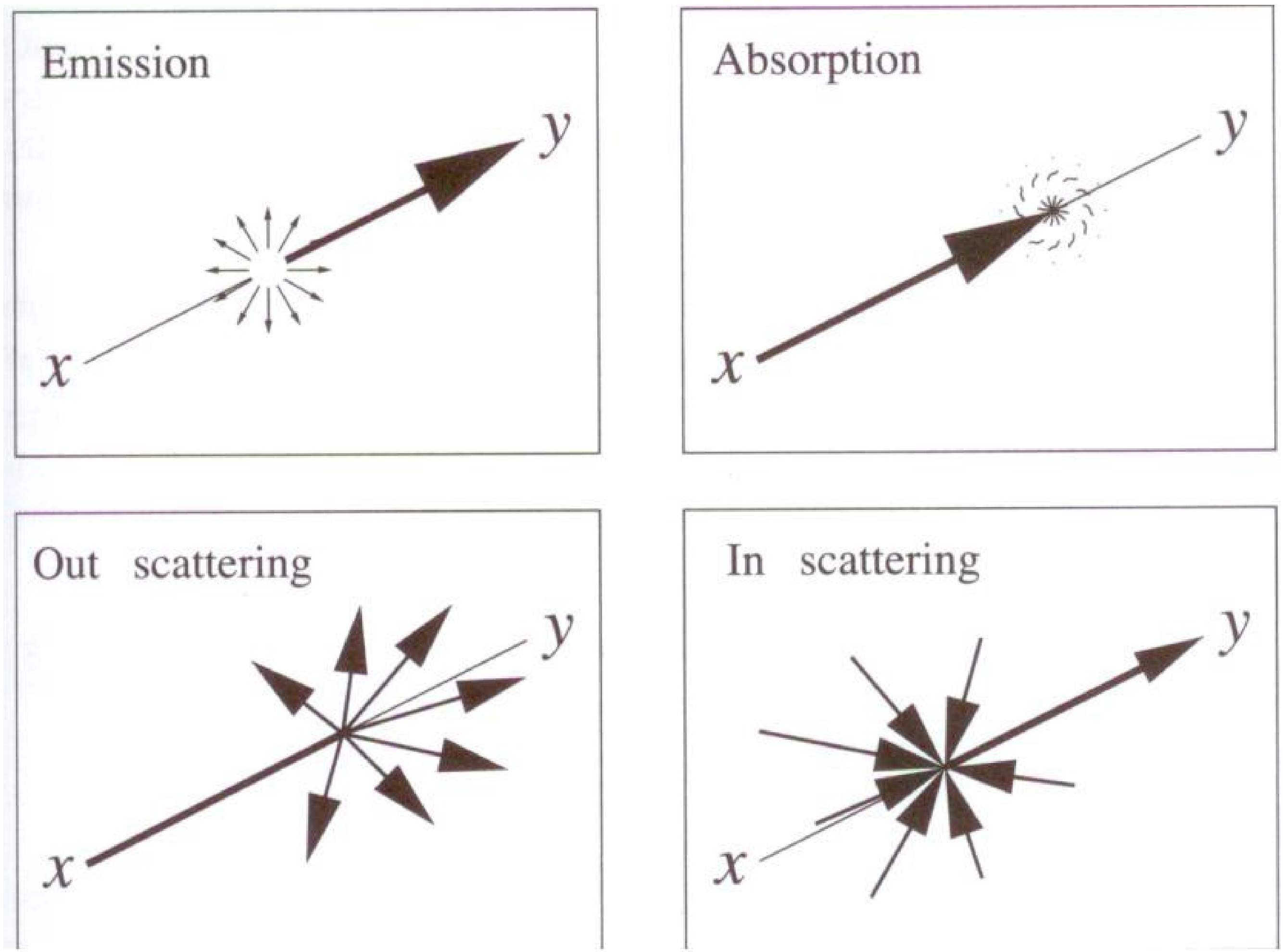
Модель

- Частицы среды бесконечно малы (шарики).
- Все процессы происходят на любом бесконечно малом отрезке светового луча.
- Все приведенные ниже выводы касаются одноцветных лучей (яркость L).



Как правило при помощи цвета исследуется несколько функций одновременно

Оптические свойства среды



Оптические свойства среды

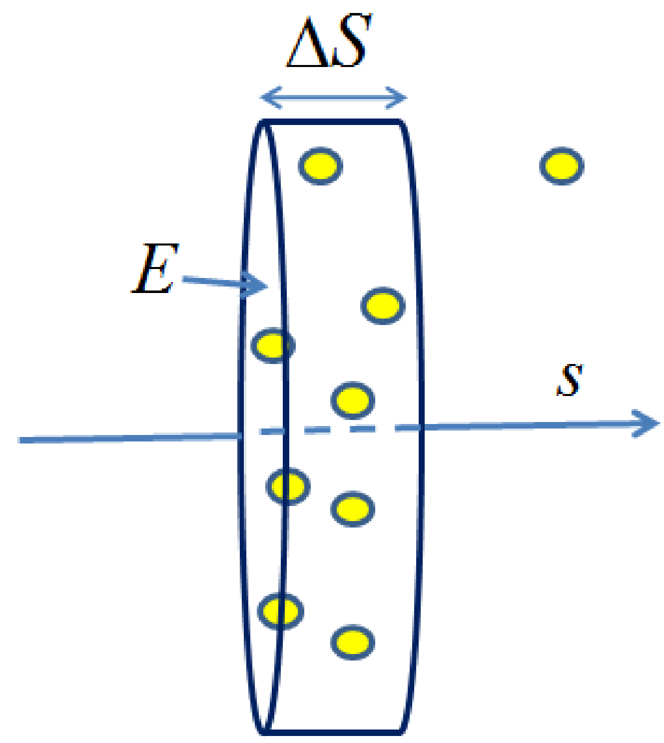
Каждый элементарный объем ОП может:

1. Испускать – *emission*
2. Поглощать – *absorbtion*
3. Рассеивать – *out scattering*
4. Получать рассеянный из соседних объемов – *in scattering*

$$\Delta W_{out} = \Delta W_{in} + \Delta W_{em} - \Delta W_{abs} - \Delta W_{outSc} + \Delta W_{inSc}$$

Только поглощение

- Частица – шарик радиуса r
- Сечение частицы – $A = \pi r^2$
- Плотность частиц – ρ
- Площадь участка – E
- Длина участка – Δs
- Число частиц – $N = \rho E \Delta s$
- Проекционная площадь – $NA = \rho E A \Delta s$



Считаем, что вероятность перекрытия частиц равна 0 (очень малое Δs)

Только поглощение

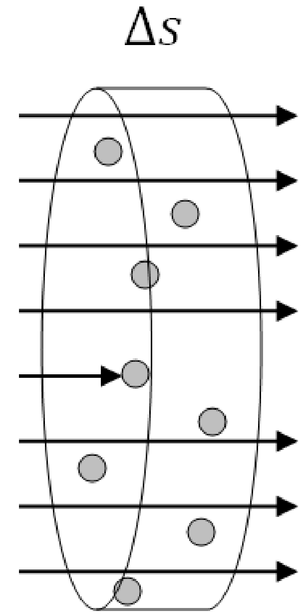
- Часть лучей *I*, которая попадет в частицы и поглотится

$$I \cdot \frac{NA}{E} = I\rho A\Delta s = I(s)\rho(s)A\Delta s$$

- Устремим Δs к нулю:

$$\frac{dI}{ds} = -\rho(s)AI(s) = -\tau(s)I(s)$$

- Коэффициент затухания $\tau(s) = \rho(s)A$



Только поглощение

- Для уравнения $\dfrac{dI}{ds} = -\tau(s)I(s)$
- Решение $I(s) = I_0 e^{-\int_0^s \tau(t)dt}$

  $I_0 = I(0)$
- Величина $T(s) = e^{-\int_0^s \tau(t)dt}$ определяет прозрачность среды между 0 (началом среды) и точкой s на луче

Только поглощение

- Уравнение для поглощения

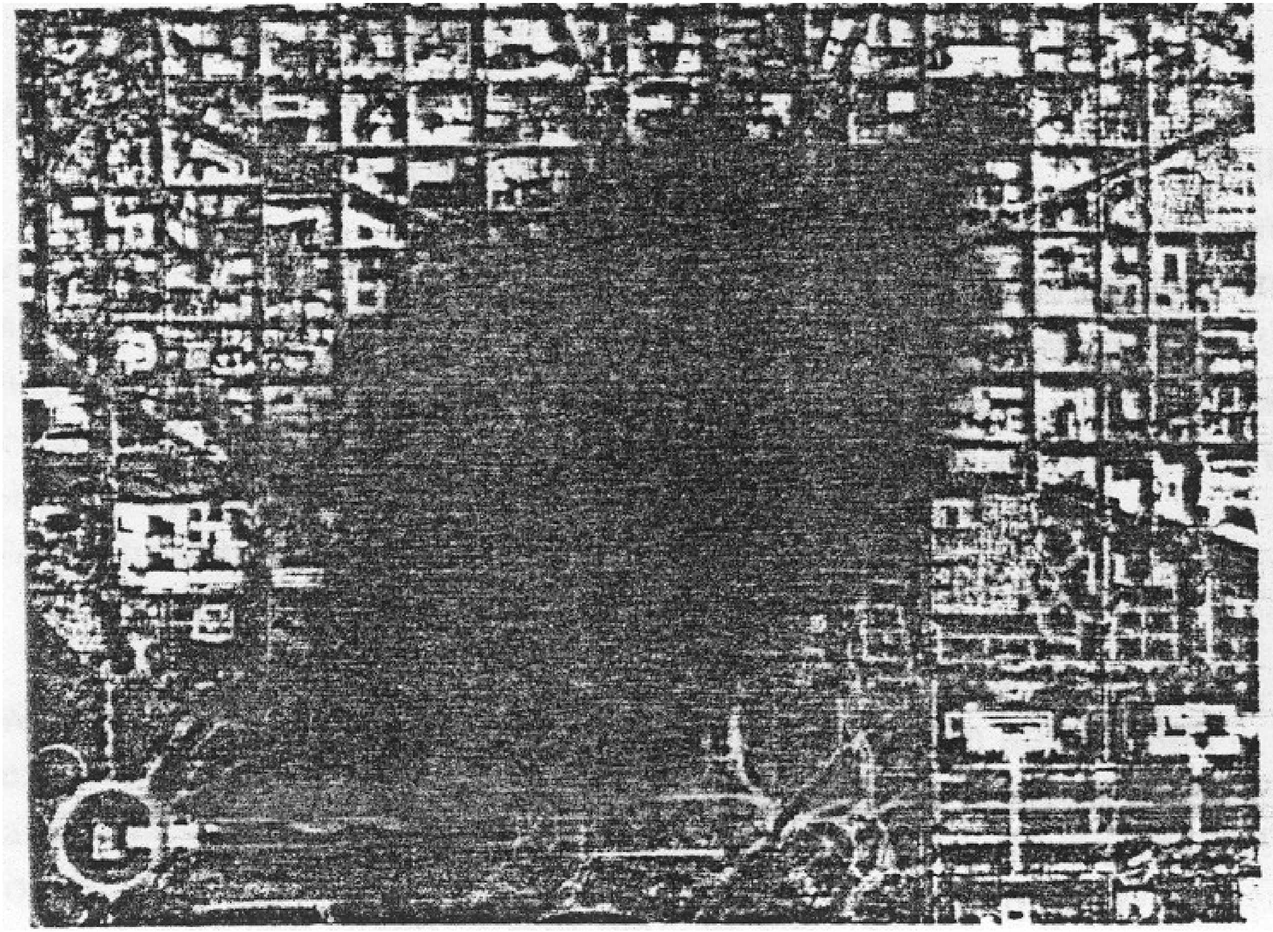
$$I(s) = I_0 \cdot T(s)$$

- Прозрачность среды от 0 до s

$$T(s) = e^{-\int_0^s \tau(t)dt}$$

- Коэффициент затухания $\tau(s)$

Только поглощение (пример)



$f(x,y,z) \sim \tau(x,y,z)$ в роли поглощающего облака (смог над городом)

Только излучение

- Среда испускает лучи

- Пусть частицы прозрачны, но светятся диффузно во все стороны с интенсив-ностью C на единицу проекционной площади

Только излучение

- Площадь $NA = \rho EA\Delta s$

- Дает поток $\rho AC\Delta s$

- и уравнение имеет вид:

$$\frac{dI}{ds} = C(s)\rho(s)A = C(s)\tau(s) = g(s)$$

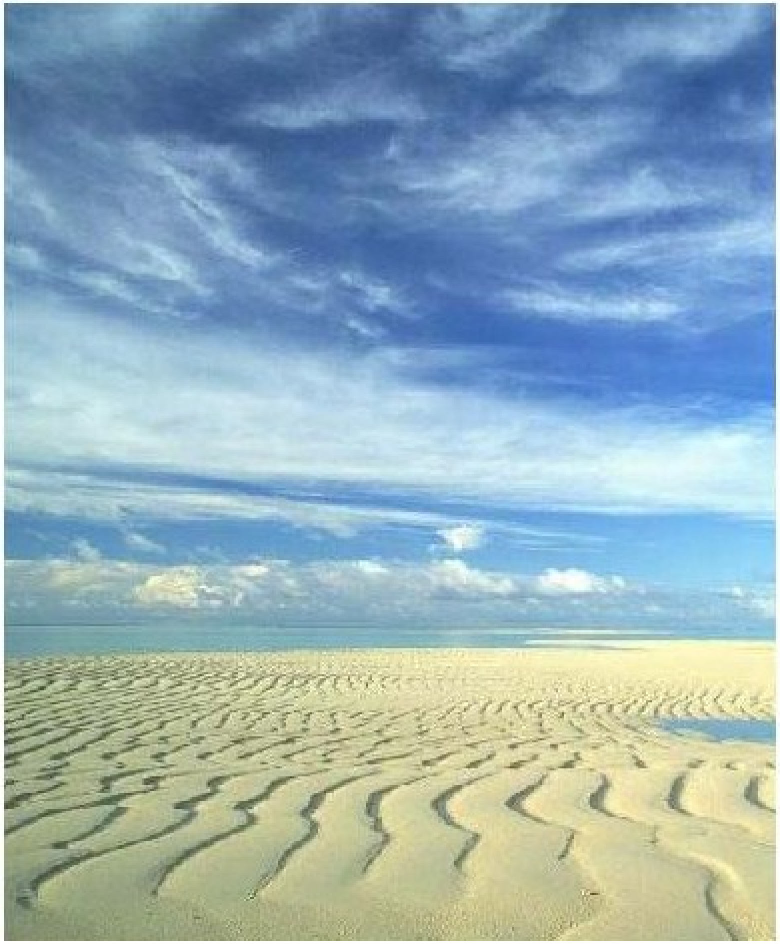
Только излучение

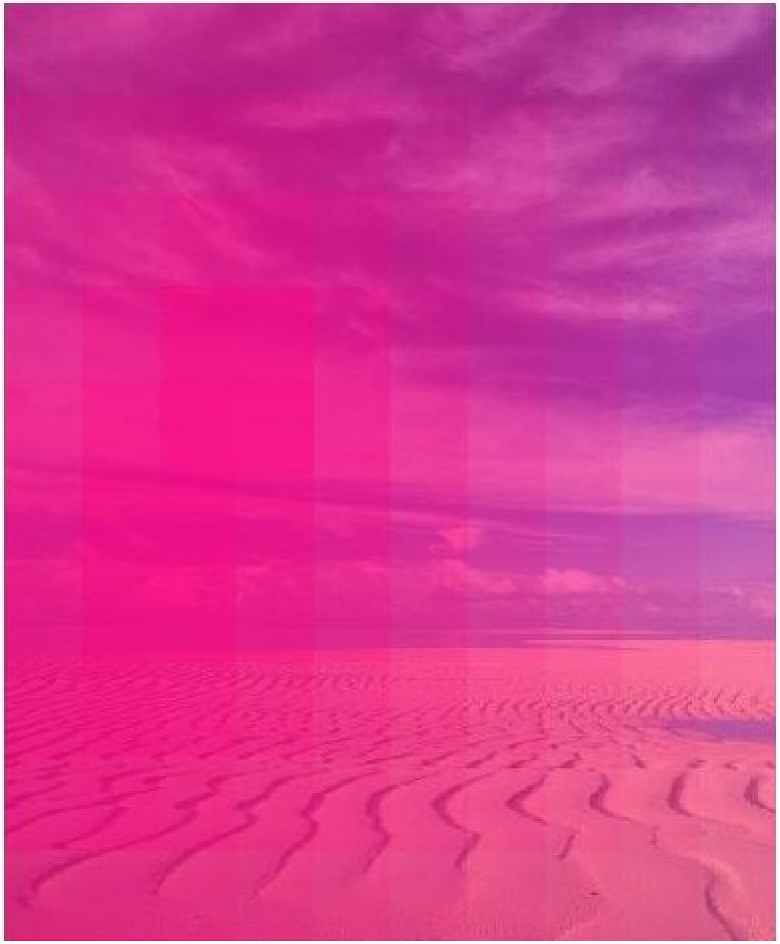
- Для уравнения $$\frac{dI}{ds} = g(s)$$

- Решение $$I(s) = I_0 + \int_0^s g(t)dt$$

$$I_0 = I(0)$$

Только излучение (пример)





Полагая $f(x,y,z) \sim g(x,y,z)$ получим изображение багряного облака

Поглощение + излучение

- Имеем сумму уравнений
$$\frac{dI}{ds} = g(s) - \tau(s) I(s)$$

- Считаем функции $g(s)$ $\tau(s)$ различными – это оказывается удобным во многих случаях, например, в медицине они назначаются различным тканям тела

Поглощение + излучение

- Переносим влево и умножаем на exp()

$$\left[\frac{dI}{ds} + \tau(s) I(s)\right] e^{\int_0^s \tau(t)dt} = g(s) e^{\int_0^s \tau(t)dt}$$

- Преобразуем

$$\frac{d}{ds}\left[I(s) e^{\int_0^s \tau(t)dt} \right] = g(s) e^{\int_0^s \tau(t)dt}$$

- Интегрируем от начала до конца ОП

Поглощение + излучение

- Интегрируем от начала до конца ОП (D - в месте наблюдателя)

$$I(D)e^{\int_0^D \tau(t)dt} - I_0 = \int_0^D g(s)e^{\int_0^s \tau(t)dt} ds$$

- Переносим I_0 вправо и делим на exp()

$$I(D) = I_0 e^{-\int_0^D \tau(t)dt} + \int_0^D g(s)e^{-\int_s^D \tau(t)dt} ds$$

Поглощение + излучение

$$I(D) = I_0 e^{-\int_0^D \tau(t)dt} + \int_0^D g(s) e^{-\int_s^D \tau(t)dt} ds$$

- Первая часть – поглощение на участке $0..D$

$$I(s) = I_0 \cdot T(s)$$

$$T(s) = e^{-\int_0^s \tau(t)dt}$$

Поглощение + излучение

$$I(D) = I_0 e^{-\int_0^D \tau(t)dt} + \int_0^D g(s) e^{-\int_s^D \tau(t)dt} ds$$

- Вторая часть – это интеграл, который подсчитывает вклад источника $g(s)$ для каждого дифференциального участка в точке s на пути луча сквозь облако с его умножением на прозрачность слоя, оставшегося до конца облака.

Поглощение + излучение

$$I(D) = I_0 e^{-\int_0^D \tau(t)dt} + \int_0^D g(s) e^{-\int_s^D \tau(t)dt} ds$$

- Запишем прозрачность слоя от s до D

$$T'(s) = e^{-\int_s^D \tau(t)dt}$$

- Тогда окончательное решение:

$$I(D) = I_0 T(D) + \int_0^D g(s) T'(s) ds$$

Частный случай

Если излучение постоянно $C(s) = C$

Тогда: $g(s) = C \cdot \tau(s)$

$$I(D) = I_0 \cdot T(D) + \int_0^D C \cdot \tau(z) \cdot e^{-\int_z^D \tau(t)dt}\, dz$$

$$I(D) = I_0 \cdot T(D) + C \cdot \int_0^D \frac{d}{dz} e^{-\int_z^D \tau(t)dt}\, dz$$

$$I(D) = I_0 \cdot T(D) + C \cdot \left(e^{-\int_D^D \tau(t)dt} - e^{-\int_0^D \tau(t)dt}\right)$$

$$I(D) = I_0 \cdot T(D) + C \cdot \big(1 - T(D)\big)$$

Это не что иное, как формула тумана, применяющаяся в библиотеках реалистического рендеринга: OpenGL, DirectX. Цвет фона и цвет тумана комбинируются с учетом прозрачности всего слоя тумана.

Поглощение + излучение

Это формула тумана из OpenGL и DirectX. Цвет фона и цвет тумана комбинируются с учетом прозрачности всего слоя тумана.

$$I(D) = I_0 \cdot T(D) + C \cdot \left(1 - T(D)\right)$$

$$T(s) = e^{-\int_0^s \tau(t)dt}$$

Вычисления

$$I(D) = I_0 \cdot T(D) + \int_0^D g(z) \cdot T'(z)\, dz$$

$$T(z) = e^{-\int_0^z \boldsymbol{\tau}(t) dt}$$

$$T'(z) = e^{-\int_z^D \boldsymbol{\tau}(t) dt}$$

$\tau(z)$ – коэффициент ослабления

$g(z)$ – светимость

Вычисления

Переходя к численному решению интеграла для N частей по оси *z* получаем:

$$\int_0^D h(z)\,dz \approx \sum_{i=1}^{N} h(z_i)\cdot\Delta z \qquad \Delta z = \frac{D}{N}$$

$$T(z) = e^{-\int_0^z \tau(t)dt} \approx e^{-\sum_{i=1}^{n_z}\tau(i\cdot\Delta z)\cdot\Delta z} = \prod_{i=1}^{n_z} e^{-\tau(i\cdot\Delta z)\cdot\Delta z}$$

$$e^{\sum_{i=1}^{N} f_i} = \prod_{i=1}^{N} e^{f_i} \qquad x^{a+b+c} = x^a\cdot x^b\cdot x^c$$

Вычисления

Первая часть уравнения:

$$I_0 \cdot T(D) \approx I_0 \cdot \prod_{i=1}^{N} t_i \qquad t_i = e^{-\tau(i\cdot\Delta z)\cdot\Delta z}$$

Вычисления

Вторая часть уравнения:

$$T'(z) = e^{-\int_z^D \boldsymbol{\tau}(t)dt}$$

$$\int_0^D g(z) \cdot T'(z)\, dz \approx \sum_{i=1}^{N} g_i \cdot \prod_{j=i+1}^{N} t_j$$

$$g_i = g(i \cdot \Delta z) \qquad t_i = e^{-\boldsymbol{\tau}(i \cdot \Delta z) \cdot \Delta z}$$

Вычисления

Все уравнение:

$$I(D) = I_0 \cdot \prod_{i=1}^{N} t_i + \sum_{i=1}^{N} g_i \cdot \prod_{j=i+1}^{N} t_j =$$

$$= g_N + t_N \cdot (g_{N-1} + t_{N-1} \cdot (g_{N-2} + t_{N-2} \cdot (\ldots +$$

$$+ (g_2 + t_2 \cdot (g_1 + t_1 \cdot I_0)) \ldots)))$$

Вычисления

Итеративный алгоритм вычисления
от 0 до D выглядит следующим образом:

```
dz = D / N;
I = I0;
for( int j=0; j < N; ++j )
    I = I * exp(-tau[j] * dz) + g[j];
```

где tau[j] – коэффициент ослабления,
    g[j] – светимость.
Оба массива имеют размерность N
и задают значения вдоль оси z от 0 до D.

Вычисления

Два варианта нормировки:

A.
```
for( int j=0; j < N; ++j )
    I = I * exp(-tau[j] * dz) + g[j];
```

B.
```
for( int j=0; j < N; ++j )
    I = I * exp(-tau[j] * dz) + g[j] * dz;
```

Если все значения g[i] равны 1, а все значения tau[i] равны 0, то в (A) результат равен N, а в (B) – равен D. Зависимость от N не удобна для пользователя, а зависимость от D довольно логична (чем больше область, тем сильнее влияние).

Вычисления (версия 2)

Итеративный алгоритм вычисления
в **обратном порядке** (от D до 0):

```
dz = D / N;
I = 0;   T = 1;   j = N;
while ( T > small_threshold  &&  --j >= 0 ) {
   I = I * T + g[j];
   T = T * exp(-tau[j] * dz);
}
I = I0 * T + I;
```

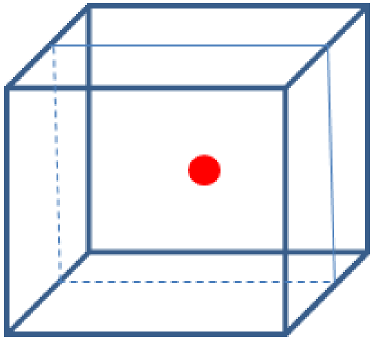
алгоритм позволяет прекратить расчет, как только вклад становится меньше заданного порога

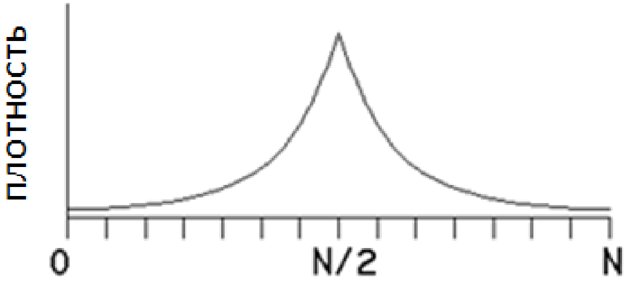
Примеры

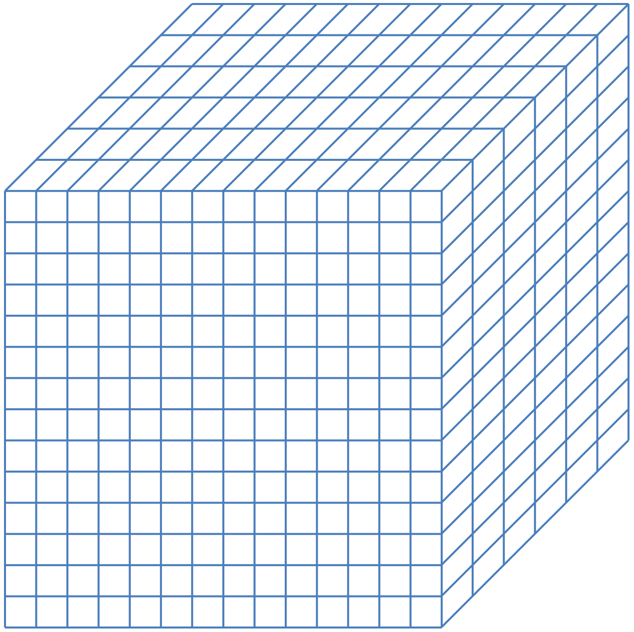
Трехмерный массив:

X,Y – плоскость изображения

Z – глубина

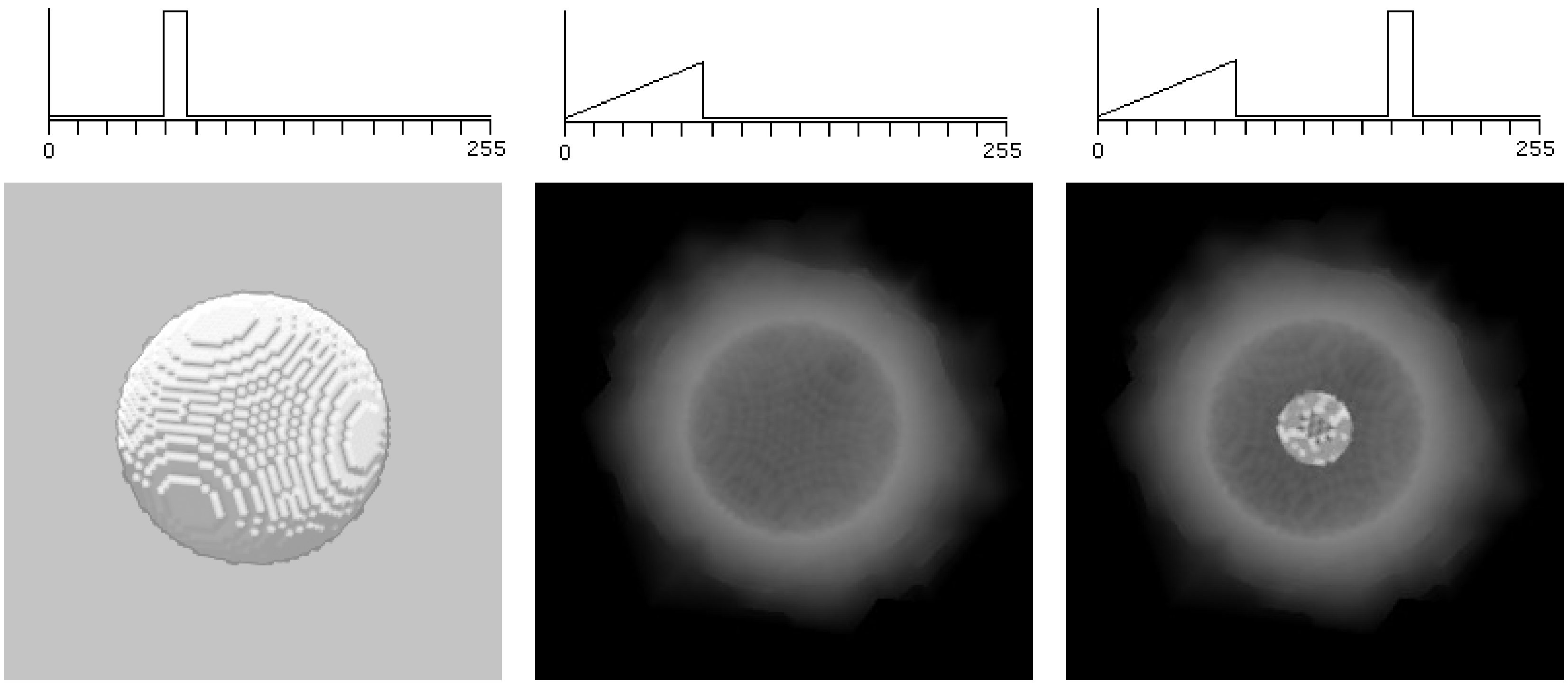






Примеры

Функция передачи на основе непрозрачности – график непрозрачности



Все размеры куба (X,Y,Z) нормированы к 0..255

Примеры

Функция передачи на основе непрозрачности
+ подключаем функцию передачи цветом