Компьютерная графика и визуализация в реальном времени

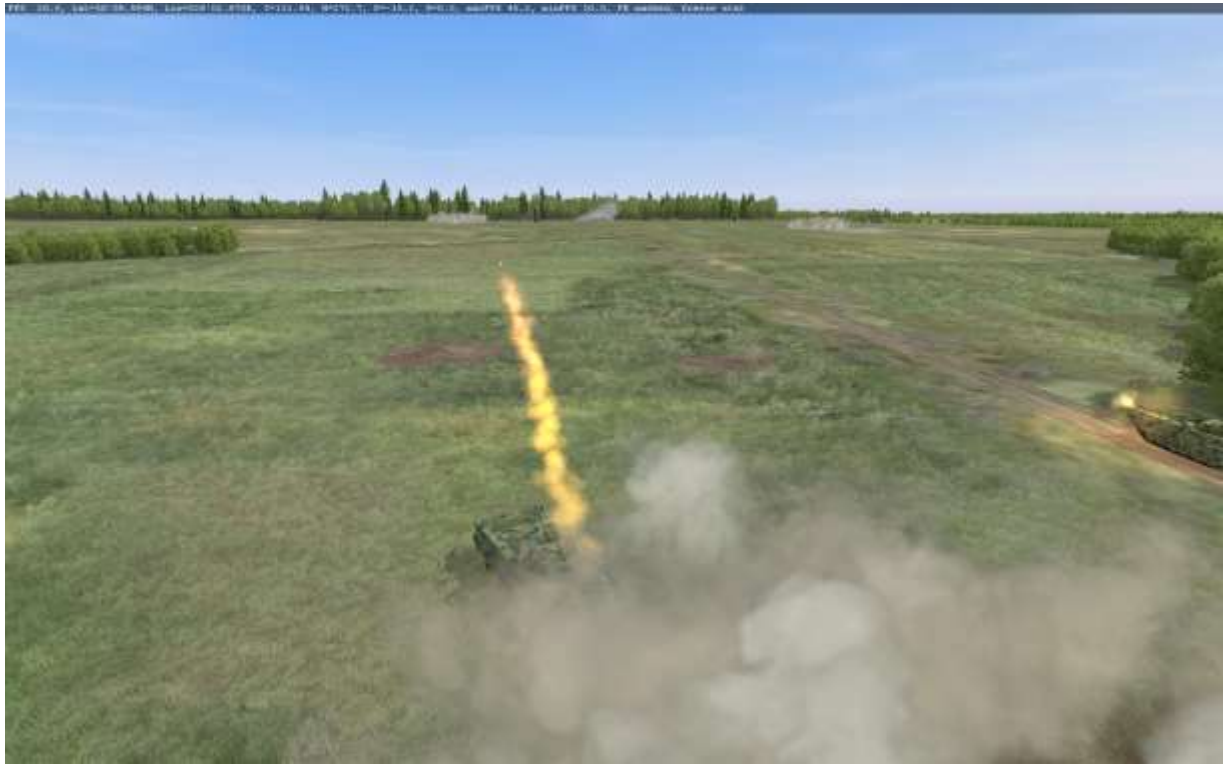
Эффекты на основе систем частиц, смешивание цветов

Алексей Романов

Эффекты на основе систем частиц

В основном используются для представления эффектов, хорошо разделяемых на составные части с одинаковым поведением

- Огонь
- Дым
- Пыль
- …



Эффекты на основе систем частиц

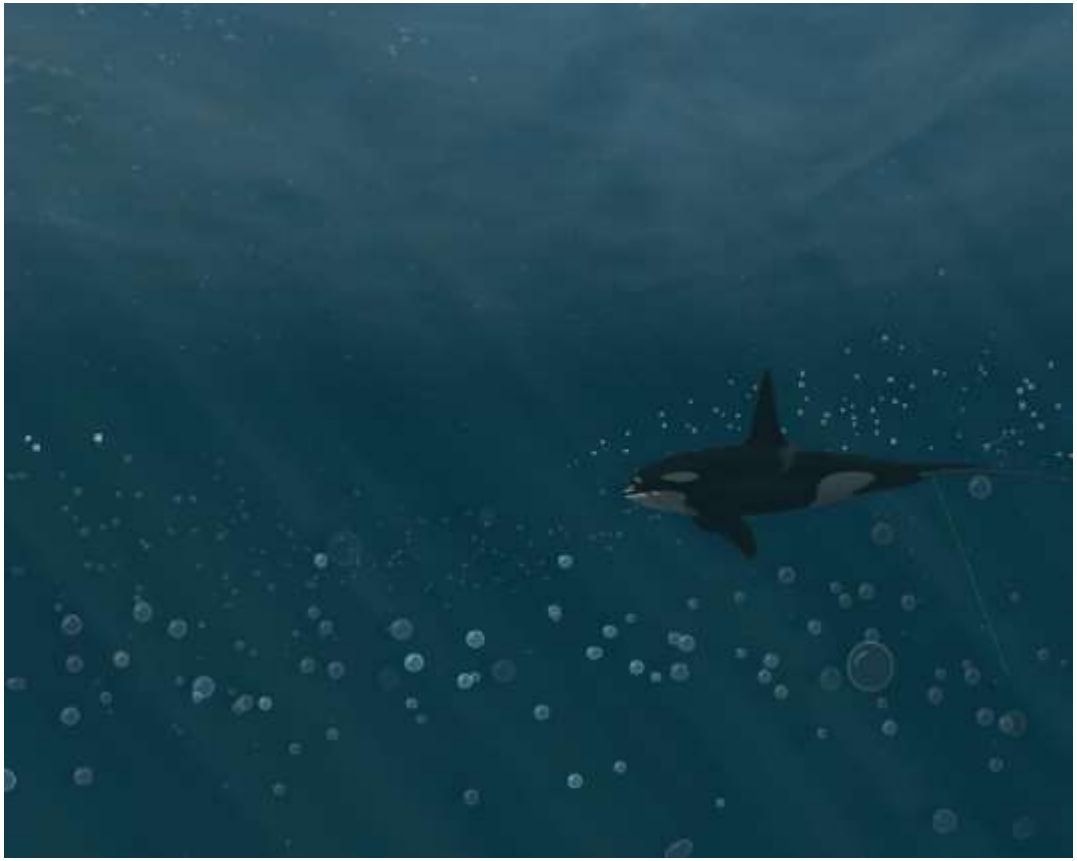
В основном используются для представления эффектов, хорошо разделяемых на составные части с одинаковым поведением

- Огонь
- Дым
- Пыль
- …

Эффекты на основе систем частиц

В основном используются для представления эффектов, хорошо разделяемых на составные части с одинаковым поведением

- Огонь
- Дым
- Пыль
- …



Эффекты на основе систем частиц

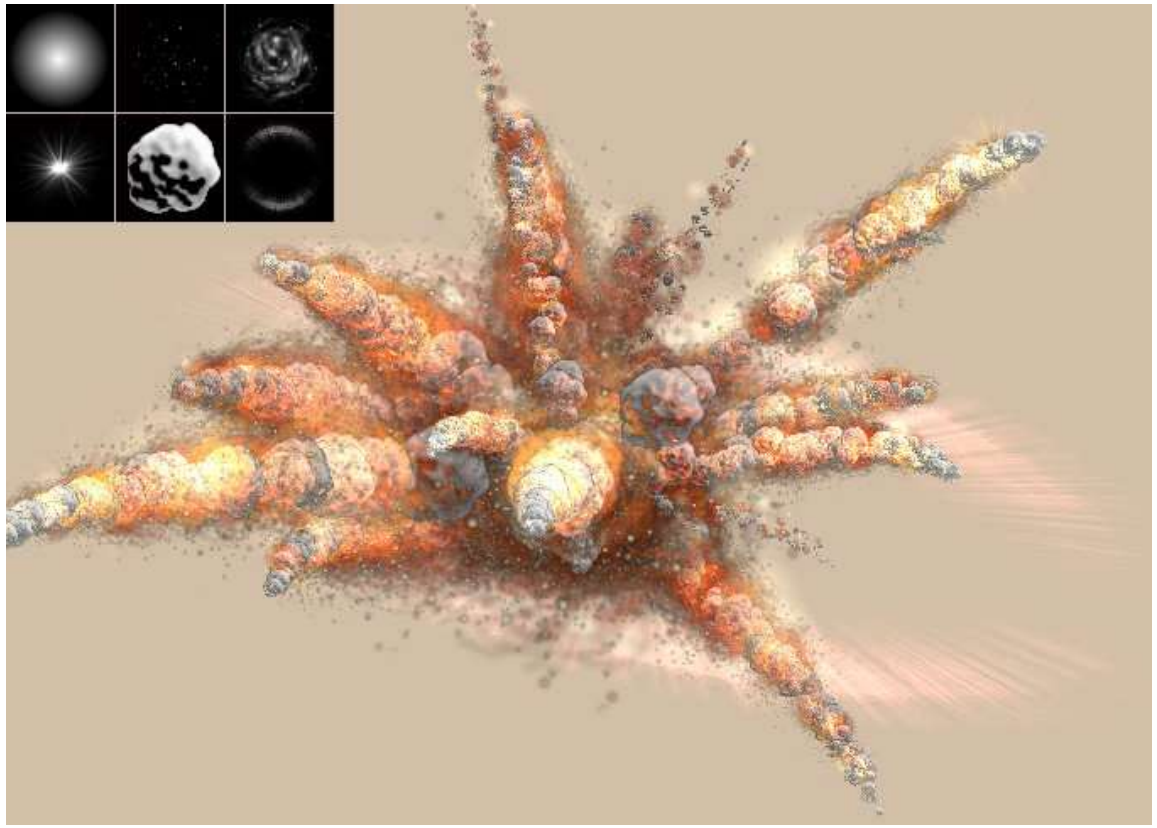
В основном используются для представления эффектов, хорошо разделяемых на составные части с одинаковым поведением

- Огонь
- Дым
- Пыль
- …

Эффекты на основе систем частиц

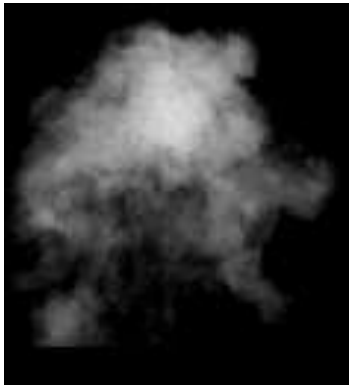
В основном используются для представления эффектов, хорошо разделяемых на составные части с одинаковым поведением

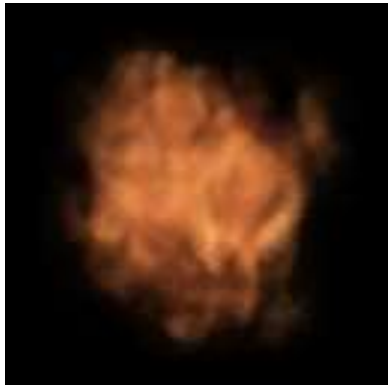
- Огонь
- Дым
- Пыль
- …

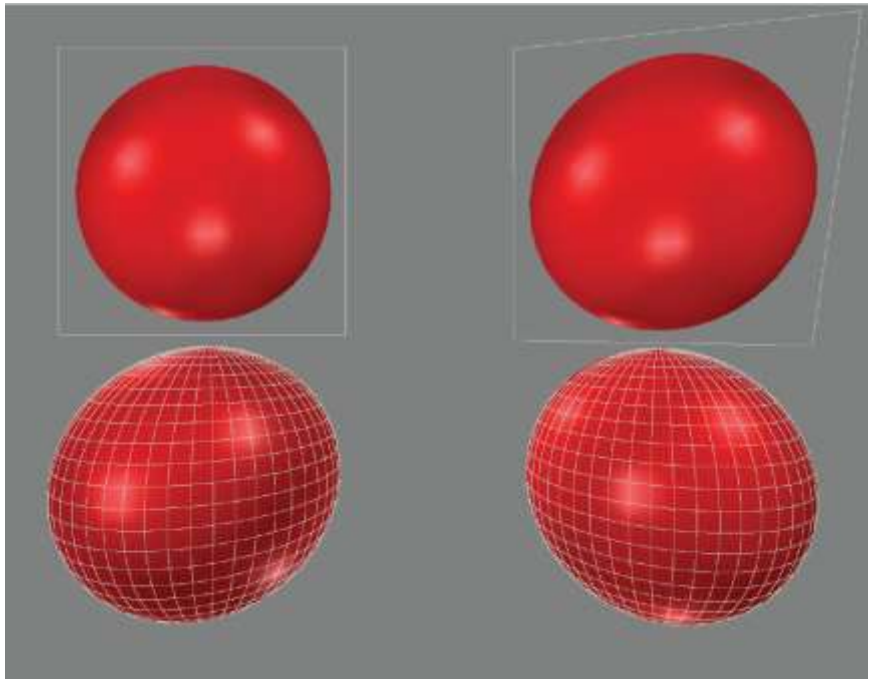


Частицы

- Моделирующие геометрию
  - Низкодетализированная геометрия
  - Плашка/*Billboard*, имитирующая освещение объекта
- *Billboard*







Базовые понятия

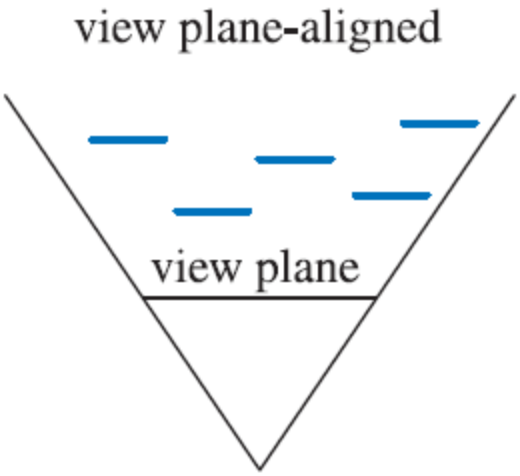
- Частица
  - Набор параметров: $\bar{p}, \bar{v}, s, \ldots$
  - Функция обновления параметров $f(t)$
    - Интегральная $\mathbf{p}(t) = f(t)$
    - Итеративная $\mathbf{p}(t+\delta) = f(t, \delta, \boldsymbol{p}(t))$
- Эмиттер
  - Определяет условия порождения, функцию обновления и начальные параметры частиц
  - В к**о**мплексных эффектах может быть частицей

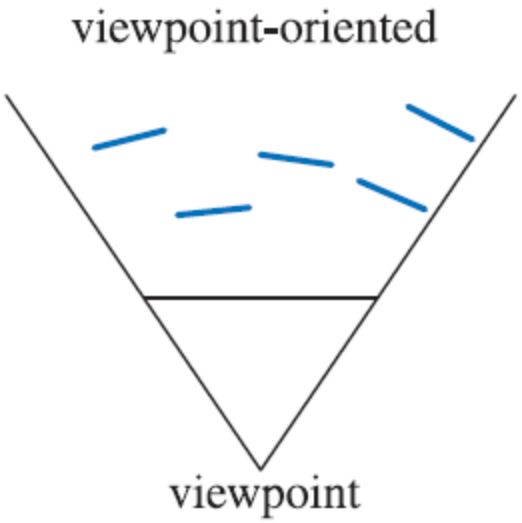
Billboard

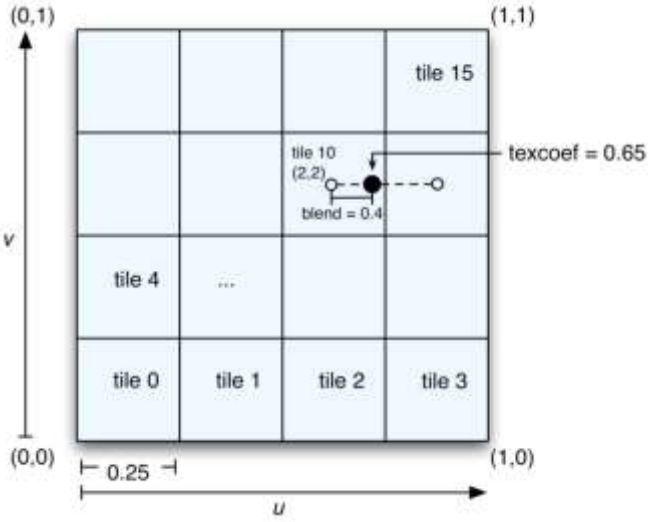
- Экранный Billboard
- Ориентированный на точку Billboard

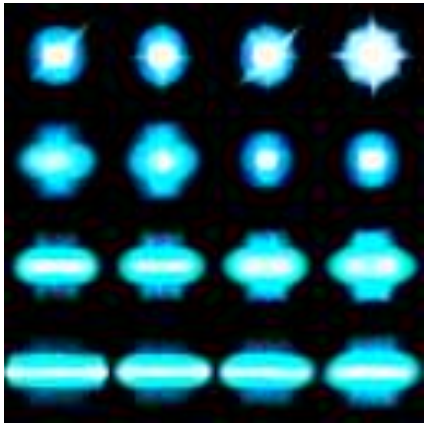
  Используются в мультиэкранной визуализации
- Volume lines

  Используются для визуализации вытянутых линий









Смешивание цветов, *blend mode*

- Непрограммируемая часть графического конвейера
- Выполняется после вычисления цвета на фрагментном шейдере для (r, g, b) и a компонент

Destination, c_d, a_d

Source , c_s, a_s

$$c_r = \phi_c(c_s f_1, c_d f_2)$$
$$a_r = \phi_a(a_d f_3, a_s f_4)$$

Result c_r, a_r

Смешивание цветов

Общий вид

$$c_r = \phi_c(c_s f_1, c_d f_2)$$
$$a_r = \phi_a(a_s f_3, a_d f_4)$$

r — результат
s — растеризуемый фрагмент
d — целевой буфер
c — цвет
a — прозрачность
f_i — множитель аргумента
ϕ — функция смешивания

Типы функции смешивания ϕ

$$\phi(a_s, a_d) = a_s + a_d$$
$$\phi(a_s, a_d) = a_s - a_d$$
$$\phi(a_s, a_d) = a_d - a_s$$
$$\phi(a_s, a_d) = max(a_s, a_d)$$
$$\phi(a_s, a_d) = min(a_s, a_d)$$

Типы множителей

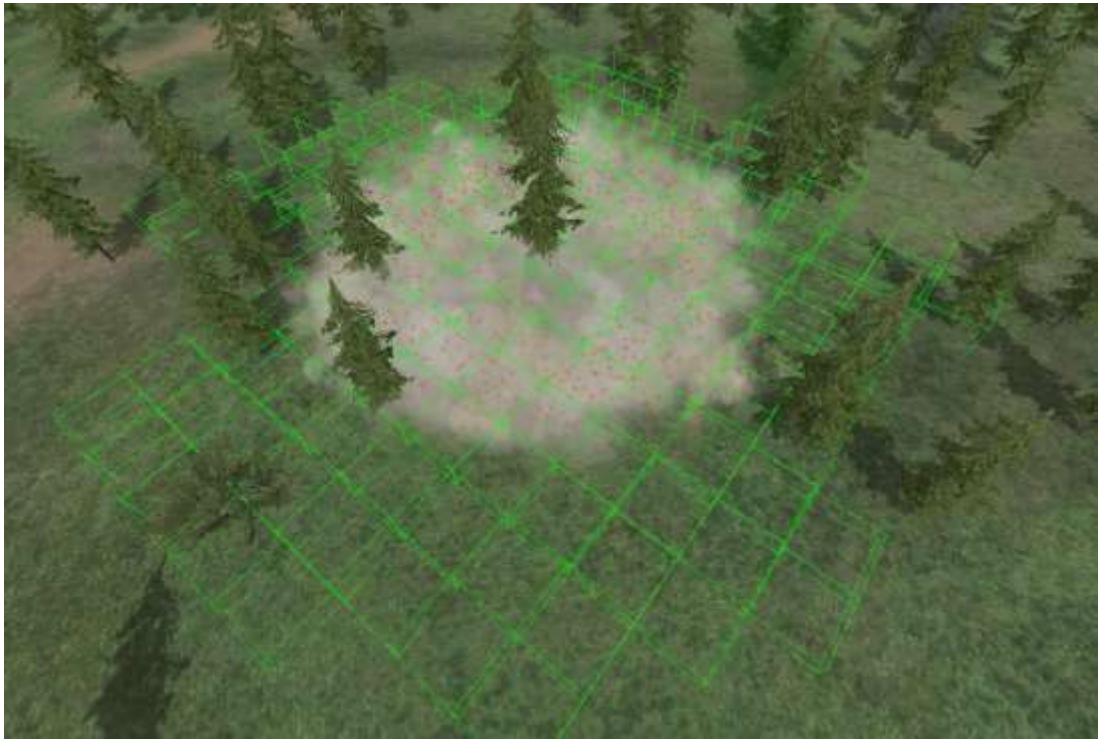
$$f = 1 \qquad f = 0$$
$$f = 1 - s_{[a|c]} \qquad f = 1 - d_{[a|c]}$$
$$f = s_{[a|c]} \qquad f = d_{[a|c]}$$

Прозрачность

- Без сортировки
- Сортировка всех частиц
  - $O(nlogn)$ на CPU
  - $O(n(logn)^2)$ на GPU (bitonic sort)
- Поячеечная сортировка частиц
- Списки фрагментов
  - Большая нагрузка на фрагментный процессор
  - Применимо только для локальных скоплений частиц
- Weight sum

Поячеечная сортировка частиц

- Сортировка по центрам ячеек
- Дублирование частиц на границах ячеек (обеспечивает плавный переход)

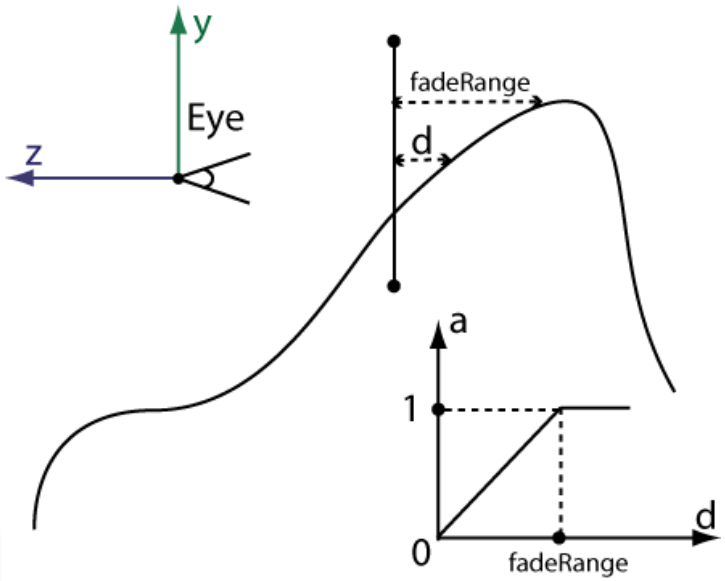


Мягкий «Z»

Прозрачность частиц представляет собой функцию от разницы значений глубин геометрии частиц и буфера глубины, уводимую в 0 при уменьшении разницы

Обычно используется линейная зависимость

$$\alpha(d) = satuate(\frac{d}{fadeRange}), fadeRange > 0$$



Модификация для работы в отрицательной области, требует выключения теста глубины (*GL_DEPTH_TEST*)

$$\alpha(d) = saturate(\frac{d - fadeRange}{|fadeRange|}), fadeRange < 0$$

Мягкий «Z», примеры

Мягкий «Z», примеры

Мягкий «Z», примеры

Мягкий «Z», примеры

Мягкий «Z», примеры

Мягкий «Z», примеры

Схемы обновления данных

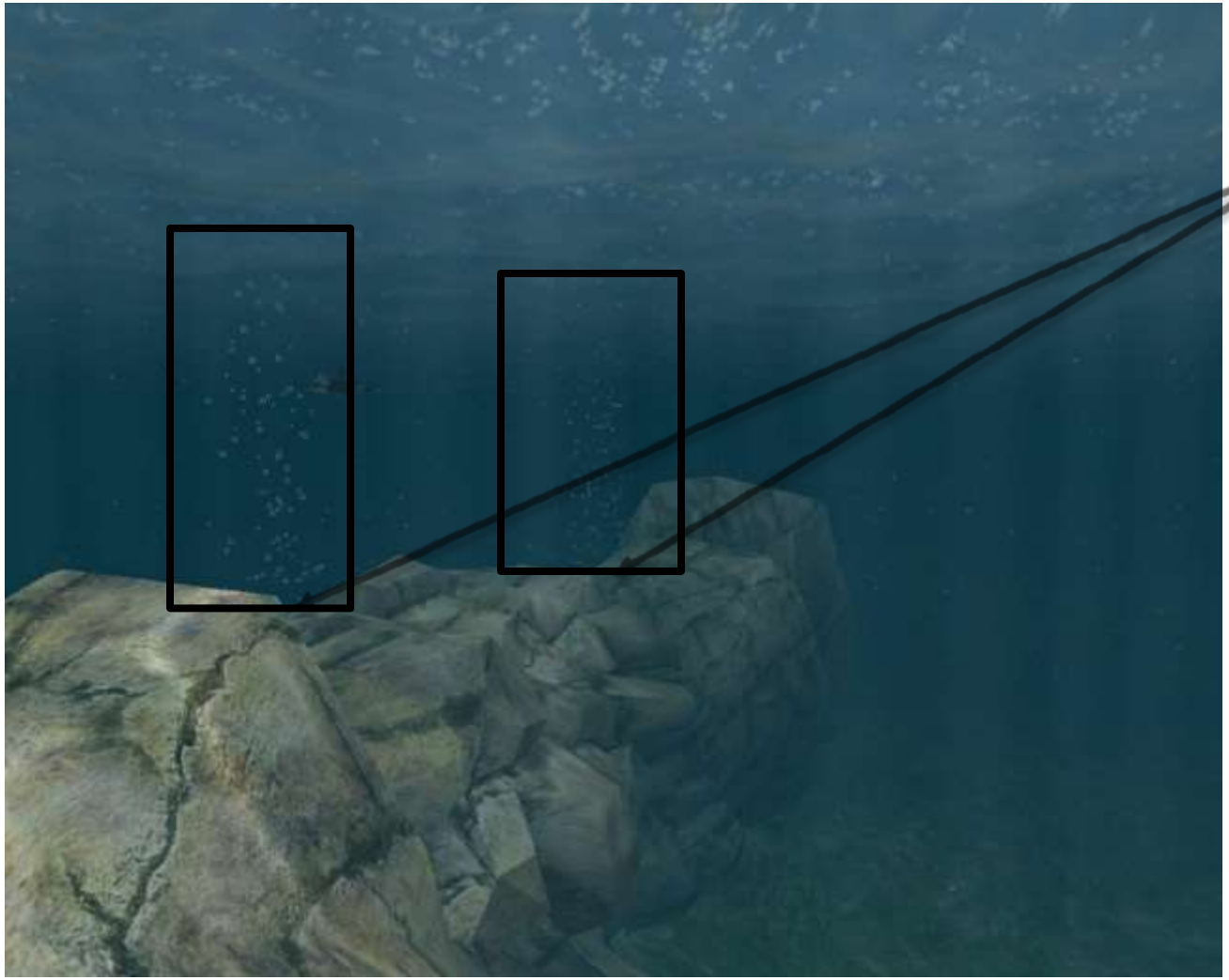
- Покадровое полное обновление данных на *CPU/GPU*
- Статические буфера
- Гибридная схема

Статический буфер

- Частицы интегральны, $\mathbf{p}(t) = f(t)$
- Схема порождения частиц – интегральна
  - Определенное количество частиц порождается с определенной частотой
- Один буфер на все экземпляры эффекта хранит начальные параметры частиц
- Каждая частица имеет уникальный идентификатор, влияющий на вариацию поведения частицы, $\mathbf{p}_{\boldsymbol{id}}(t) = f(t, id)$
- Обсчет поведения частицы производится на вершинном шейдере

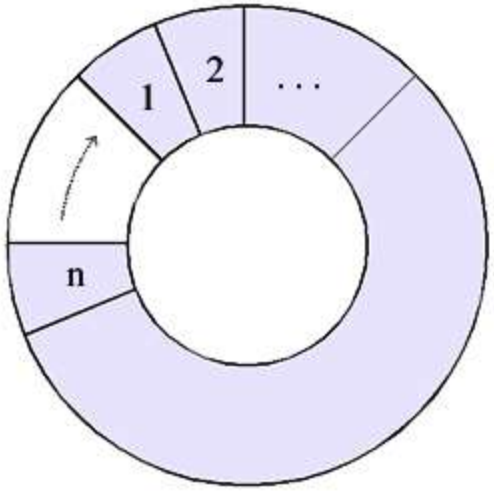
Статические буфера, пример

Статические буфера, пример



Гибридная схема

- Параметры частиц – интегральны
- Схема порождения частиц – произвольна
- Реализуется циклическим буфером
- Мэпируются только новые частицы
- Отрисовка в 2 вызова
- **+**
  - Производительность обработки частиц
- **-**
  - Управление временем жизни частиц происходит на CPU
  - Частицы в буфере должны бать расположены в порядке возрастания оставшегося времени жизни

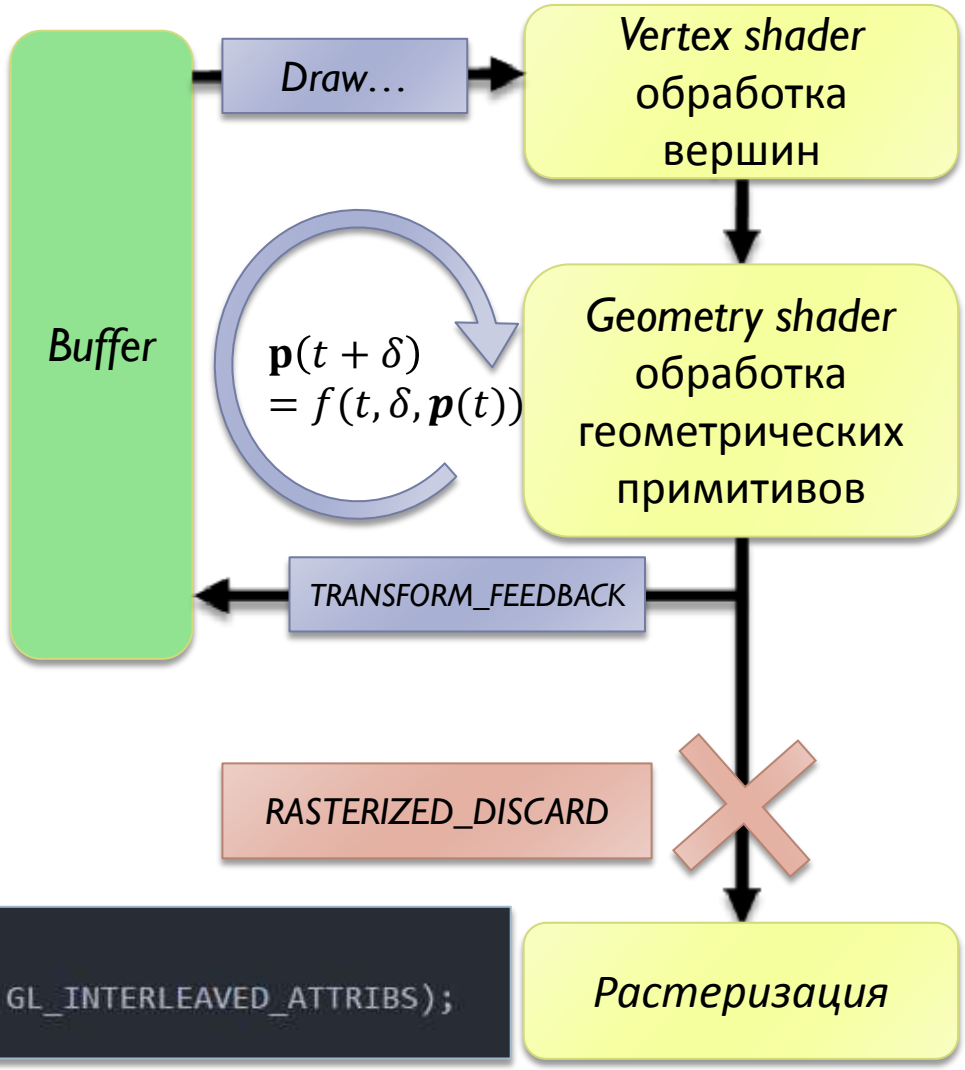


Гибридная схема, пример

Гибридная схема, пример

Обновление частиц на GPU

- Обработка частиц на вершинном-геометрическом шейдер
  - Результат вершинного/геометрического шейдера записывается в буфер (GL_TRANSFORM_FEEDBACK)
  - Примитивы не доходят до растеризации (GL_RASTERIZER_DISCARD)
- Обработка частиц на *CUDA/compute shader, glsl ≥4.3*




```
out vec2          out1;
out int           out2;
out unsigned int out3;

void main()
{
   out1 = vec2(30, 30);
   out2 = -30;
   out3 = 3000000000u;
}
```

```
char * out[3] = {"out1", "out2", "out3"};
glTransformFeedbackVaryings(program, 3, out, GL_INTERLEAVED_ATTRIBS);
```

Производительность визуализации частиц

- Большая нагрузка на фрагментный процессор
- Большой *overdraw* (количество растровых операций на экранный пиксель)
- Низкая производительность при сложном фрагментном шейдере

Производительность визуализации частиц

- Большая нагрузка на фрагментный процессор
- Большой *overdraw* (количество растровых операций на экранный пиксель)
- Низкая производительность при сложном фрагментном шейдере
- Решение – отрисовка во внеэкранную текстуру меньших размеров

Отрисовка частиц во внеэкранную текстуру

Желаемая функция смешивания цветов – blend функция

$$R_{rgb} = S_{rgb} * S_a + D_{rgb} * (1 - S_a)$$

1 итерация

$$S0_{rgb} * S0_a + D_{rgb} * (1 - S0_a)$$

2 итерация

$$S1_{rgb} * S1_a + (S0_{rgb} * S0_a + D_{rgb} * (1 - S0_a)) * (1 - S1_a)$$

$$S_{n_{rgb}} * S_{n_a} +$$

$$(S_{n-1_{rgb}} * S_{n-1_a} + (\cdots + D_{rgb} * (1 - S0_a)) * (1 - S_{n-1_a})) * (1 - S_{n_a})$$

Формула наложения всех частиц

Отрисовка частиц во внеэкранную текстуру

- Blend-функция отрисовки частицы P в текстуру

$$S_{rgb} = P_{rgb} * P_a$$

$$S_a = (1 - P_a)$$

$$R_{rgb} = S_{rgb} + D_{rgb} * (1 - S_a)$$

$$R_a = S_a * D_a$$

$$R_{rgb}(0) = (0, 0, 0)$$

$$R_a(0) = 1$$

Начальные условия

- Наложение текстуры на экран

$$R = T_{rgb} + T_a * D_{rgb}$$

Наложение текстуры на экран

Наложение текстуры на экран

Проблемы в области перепада глубин

- Недостаток разрешения для теста глубины
- Некорректная работа техник, использующих текстуру глубины

Проблема низкого разрешения текстуры глубины

Решение – отрисовка частиц в экранном разрешении в областях перепада глубин и в низком разрешении в остальной части экрана

- Отрисовка частиц во внеэкранную текстуру
- Маркировка в экране областей перепада глубин, используя буфер шаблона
- Отрисовка частиц в экранном разрешении в помечанных областях

Выделение границ перепада глубин

- Фильтр Собеля выделения границ

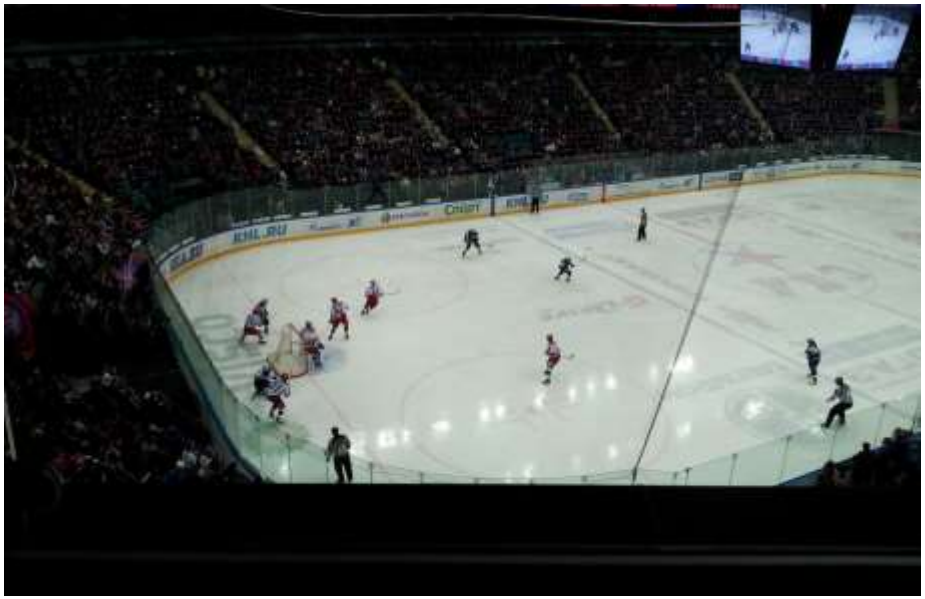
- Дилатация границ

Фильтр Собеля

$$G_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} \qquad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} \qquad G = \sqrt{G_x^2 + G_y^2}$$

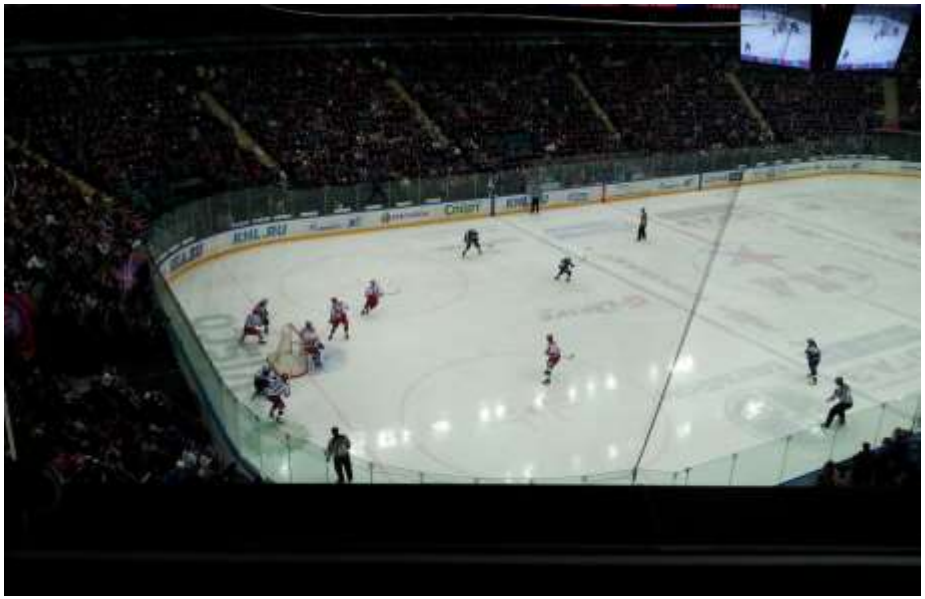
Фильтр Собеля

$$G_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} \qquad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} \qquad G = \sqrt{G_x^2 + G_y^2}$$



Фильтр Собеля

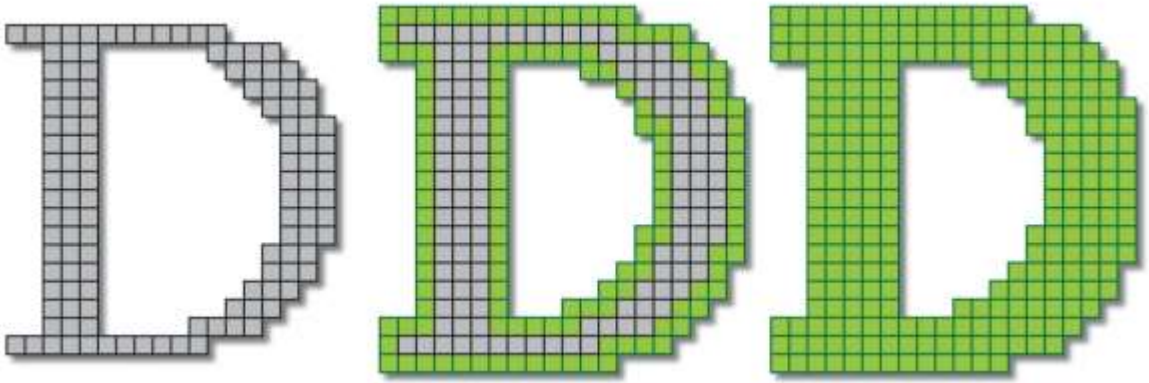
$$G_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} \qquad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} \qquad G = \sqrt{G_x^2 + G_y^2}$$

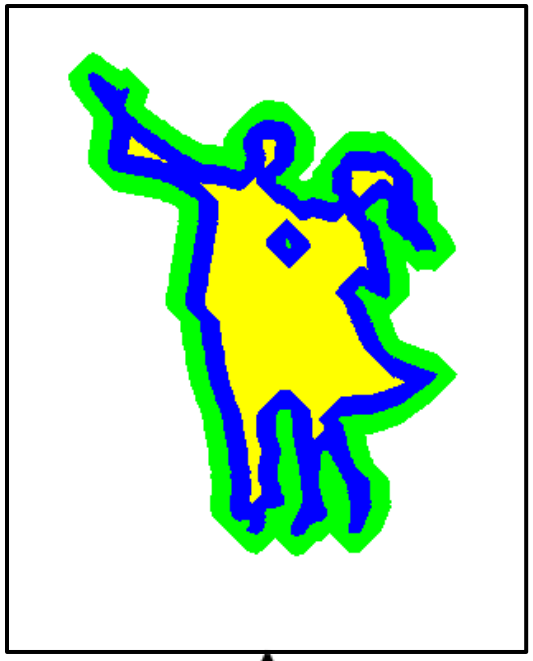


Дилатация

Элемент морфологической обработки изображения

$$A \oplus B = \{z \in E | (B^s)_z \cap A \neq \varnothing\}$$

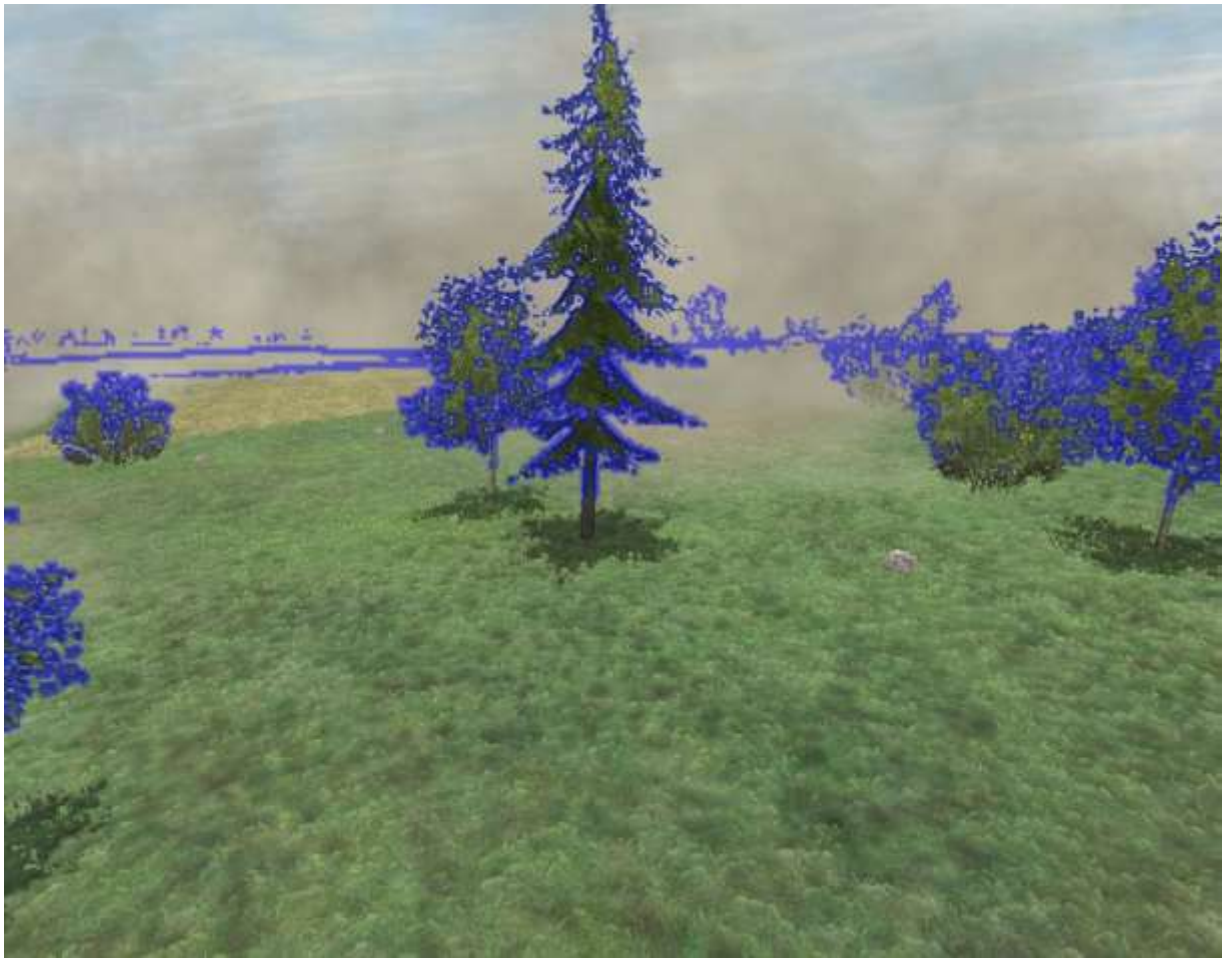




Дилатация (зеленый), эрозия (желтый) и оригинальное изображение (синий)

Результаты маркировки экранных областей

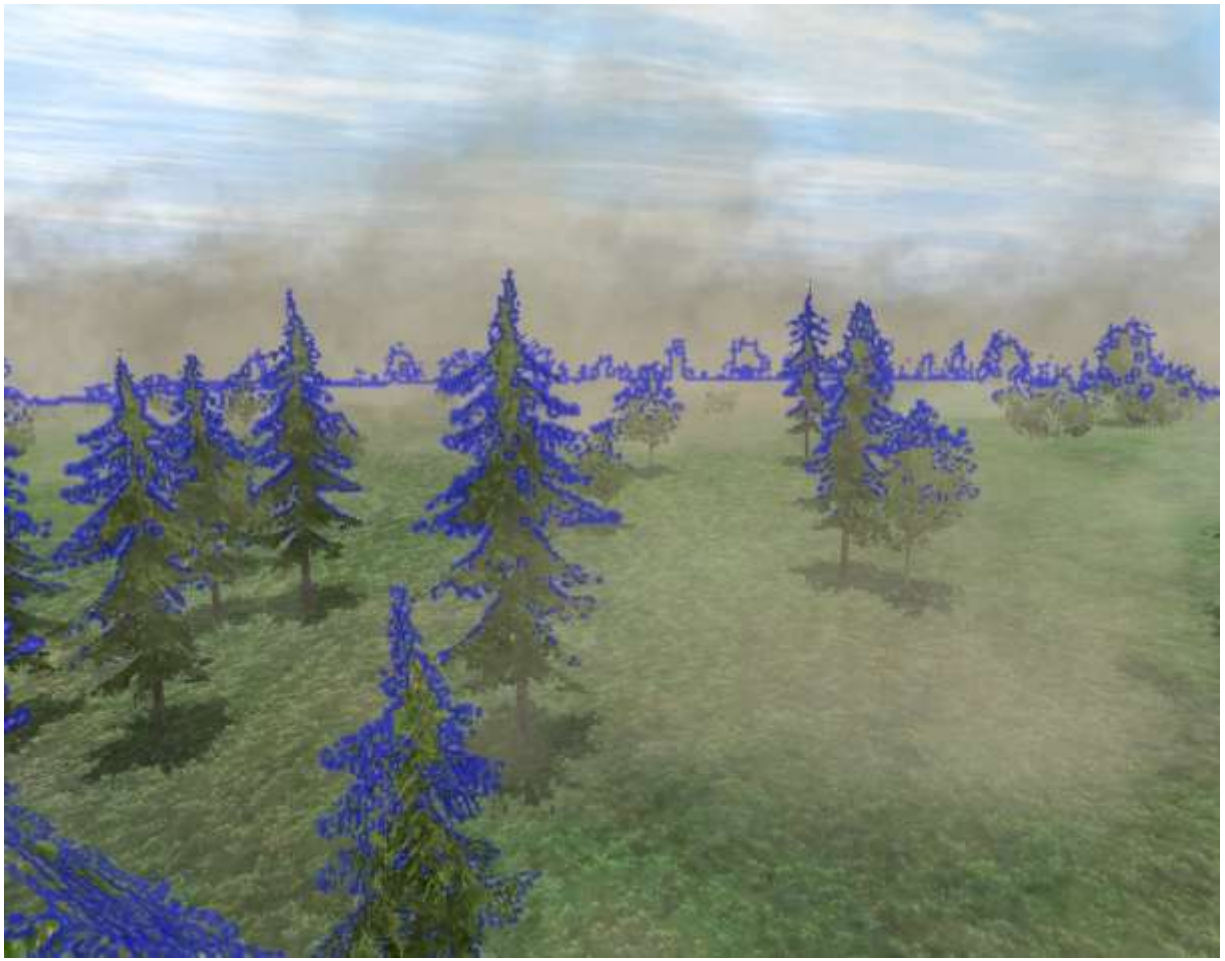
Результаты маркировки экранных областей



Результаты маркировки экранных областей

Результаты маркировки экранных областей

Результаты маркировки экранных областей



Результаты маркировки экранных областей