

# Лекция 8. Карты нормалей

Илья Макаров

**ИТМО**

2 ноября 2022  
Санкт-Петербург

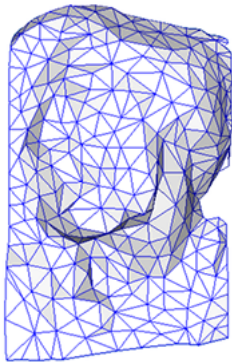


### Карты нормалей

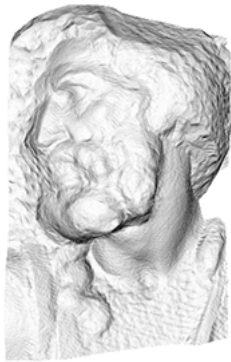
Позволяют имитировать неровности и шероховатости поверхности, значимо экономить вершины.



original mesh  
4M triangles



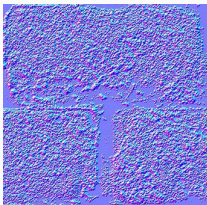
simplified mesh  
500 triangles  
[mse.itmo.ru](http://mse.itmo.ru)



simplified mesh  
and normal mapping  
500 triangles

### Карты нормалей

Координаты кодируются в RGB каналы. А сама карта загружается в видеопамять в виде текстуры.



#### Замечание:

- карты нормалей могут быть достаточно "тяжелыми";
- необходимо найти некоторый баланс между количеством вершин и размерами карты нормалей.

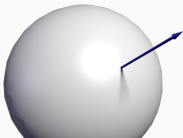
### Tangent space

- Сама карта связывается с вершинами также как и диффузная текстура.
- Однако нормали определены в пространстве треугольников (**tangent space / image space**).

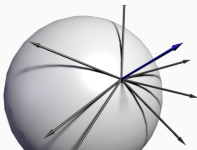
Нам же нужно перейти в пространство модели, чтобы корректно применить освещение в шейдере.

### Построение tangent space

Имеем вектор направленный перпендикулярно нашему треугольнику (вектор нормали):

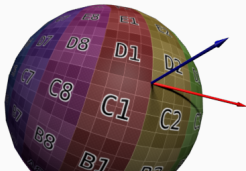


Хочется построить вектор в плоскости обрабатываемого треугольника:

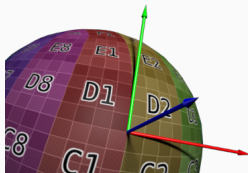


### Построение tangent space

Удобный вариант привязаться к текстурным координатам:



И затем дополнить до базиса:



## Как вычислять вектора

Для треугольника достаточно разрешить уравнение (систему уравнений):

$$\begin{aligned}\Delta pos_1 &= \Delta uv_1.x \cdot \vec{T} + \Delta uv_1.y \cdot \vec{B} \\ \Delta pos_2 &= \Delta uv_2.x \cdot \vec{T} + \Delta uv_2.y \cdot \vec{B}\end{aligned}\tag{1}$$

Относительно векторов  $\vec{T}$  и  $\vec{B}$ . Для перехода из одного пространства в другое, нужно построить соответствующую матрицу:

## Как вычислять вектора

Для перехода из одного пространства в другое, нужно построить соответствующую матрицу:

$$\begin{bmatrix} N_x & T_x & B_x \\ N_y & T_y & B_y \\ N_z & T_z & B_z \end{bmatrix} \quad (2)$$

Для обращения преобразования же достаточно просто посчитать обратную матрицу:

$$\begin{bmatrix} N_x & T_x & B_x \\ N_y & T_y & B_y \\ N_z & T_z & B_z \end{bmatrix}^{-1} = \begin{bmatrix} N_x & T_x & B_x \\ N_y & T_y & B_y \\ N_z & T_z & B_z \end{bmatrix}^T \quad (3)$$



### Шейдеры

Вычислим позицию векторов в пространстве камеры:

```
vertexNormal_cameraspace = MV3x3  
    * normalize(vertexNormal_modelspace);  
vertexTangent_cameraspace = MV3x3  
    * normalize(vertexTangent_modelspace);  
vertexBitangent_cameraspace = MV3x3  
    * normalize(vertexBitangent_modelspace);
```

Построим матрицу TBN:

```
mat3 TBN = transpose(mat3(  
    vertexTangent_cameraspace,  
    vertexBitangent_cameraspace,  
    vertexNormal_cameraspace  
));
```

## Шейдеры

Получим нашу нормаль в пространстве TBN:

```
vec3 TextureNormal_tangentspace = normalize(  
    texture(NormalTextureSampler, UV).rgb * 2.0 - 1.0);
```

Применим наше освещение в пространстве TBN:

```
LightDirection_tangentspace = TBN  
    * LightDirection_cameraspace;  
EyeDirection_tangentspace = TBN  
    * EyeDirection_cameraspace;
```