WebGL によるデータ可視化入門*

照明

陰山 聡

神戸大学 システム情報学研究科 計算科学専攻

2013.07.02

*情報可視化論 X021（2013 年前期） LR301 →演習室

照明

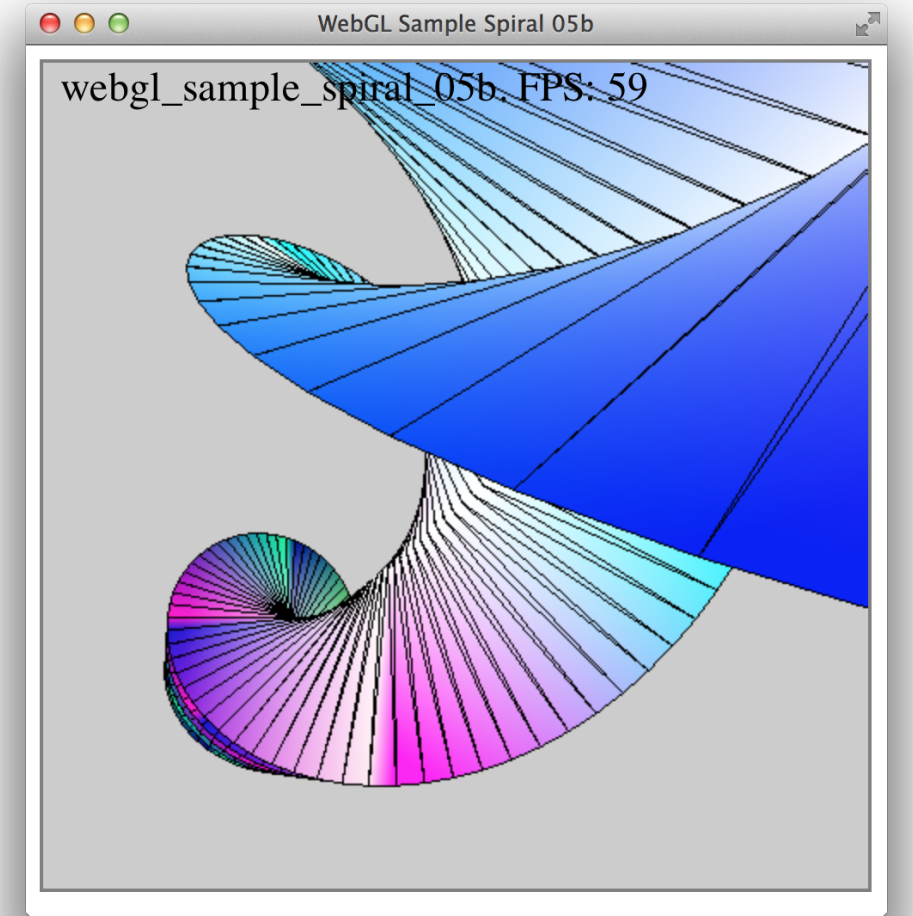
レポート課題

索引

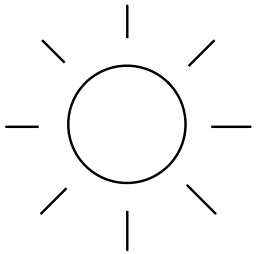
References

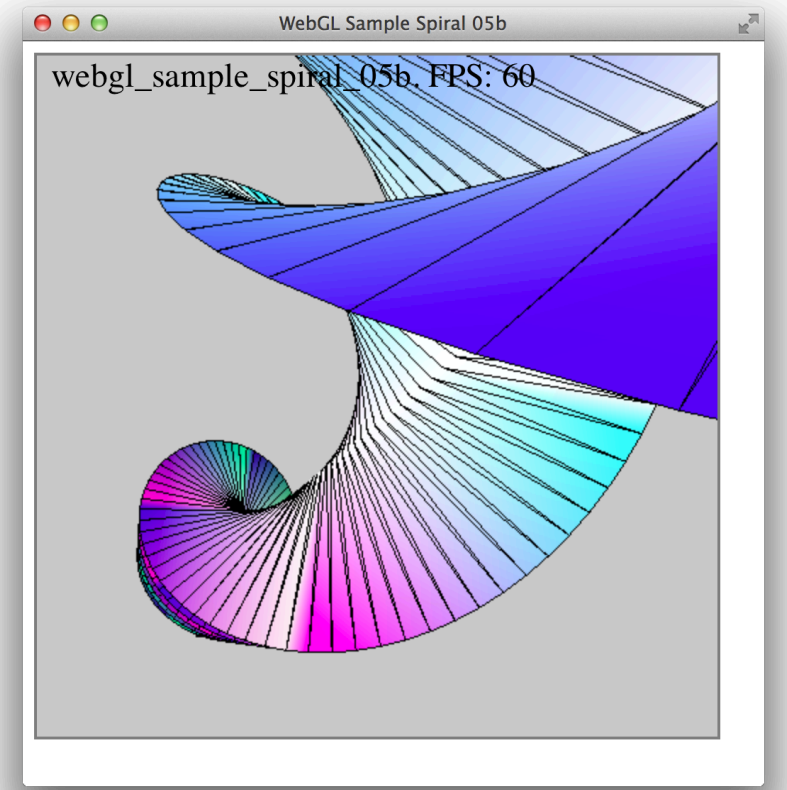
照明

- これまでは照明を全く考えていなかった
- 各頂点がそれぞれ指定された色で「光る」

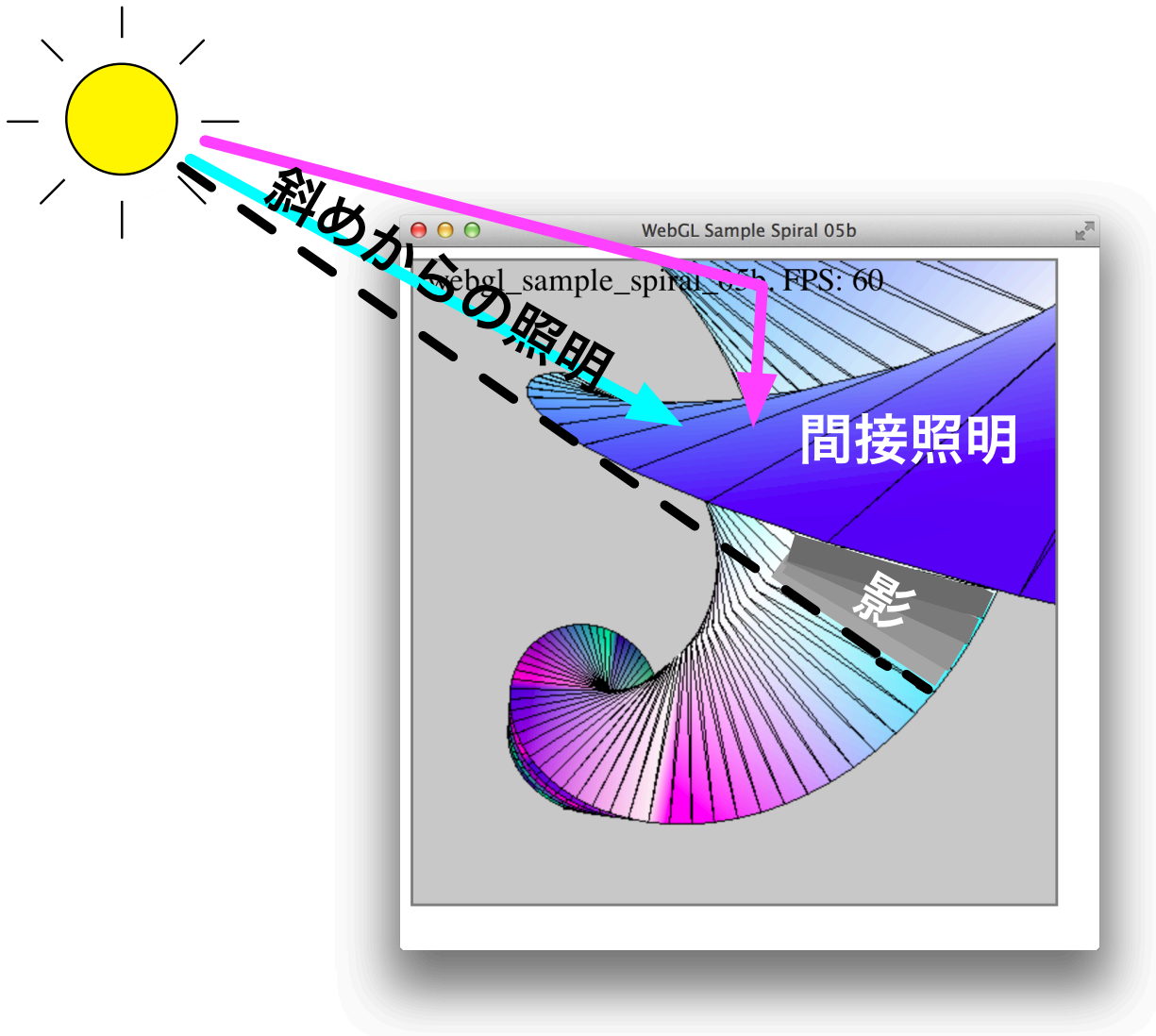


光源を置く





結構大変


斜めからの照明
WebGL Sample Spiral 05b
webgl_sample_spiral_05b, FPS: 60
間接照明
影

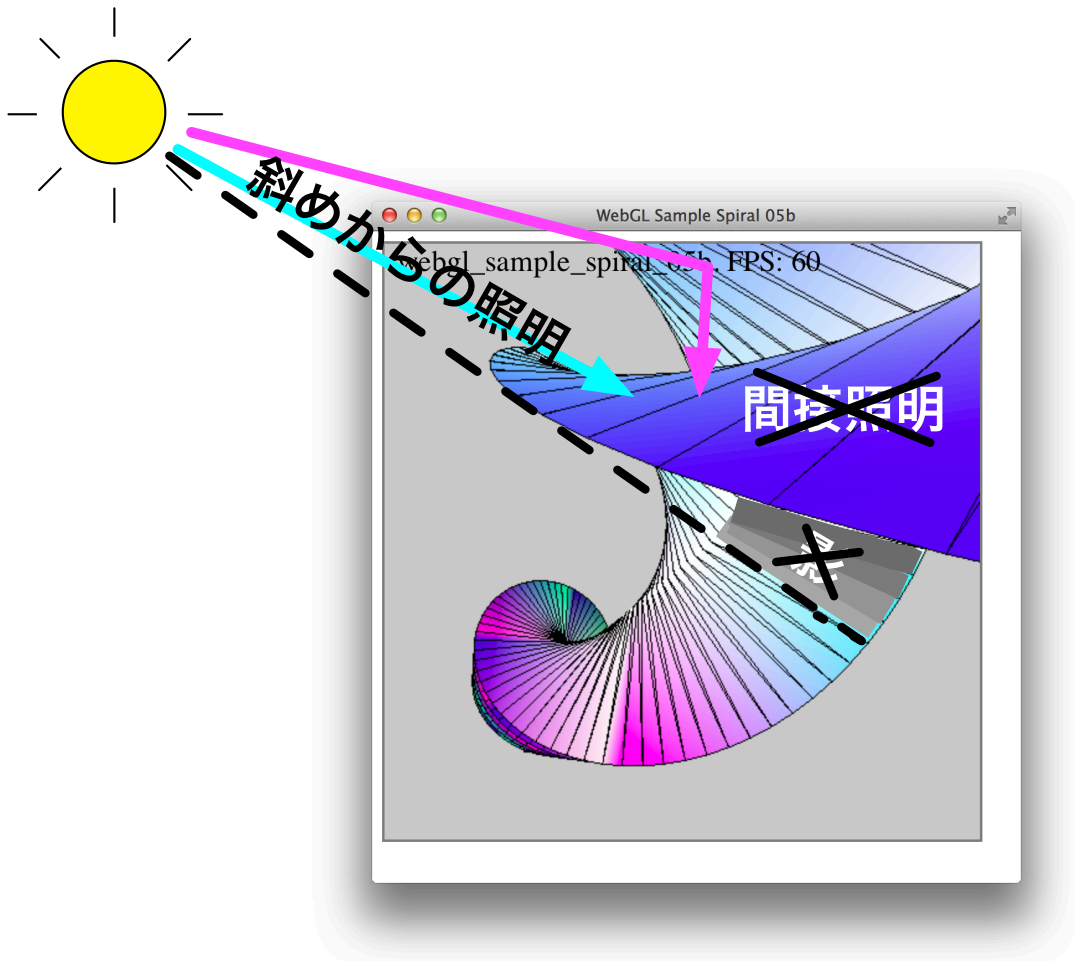
二つの照明モデル

ある頂点の照明状態を計算するとき、

1. 大域照明モデル： 他の物体の存在を計算に入れる。
2. 局所照明モデル： 他の物体の存在を無視する 。

局所照明モデル

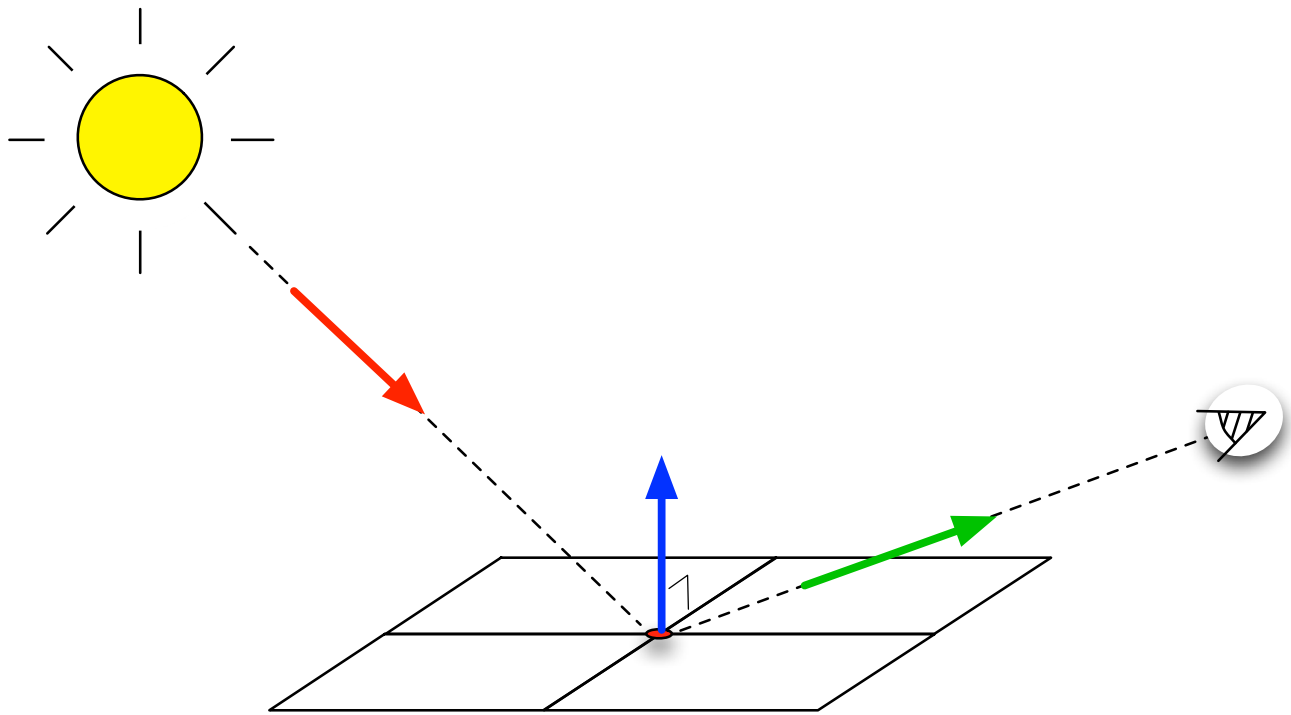
間接照明†や影はできない。



†後述する環境光によって擬似的にその効果を入れる。

局所照明モデル

ある頂点での色（反射）は、3 つの単位ベクトルの関数。



WebGL の照明

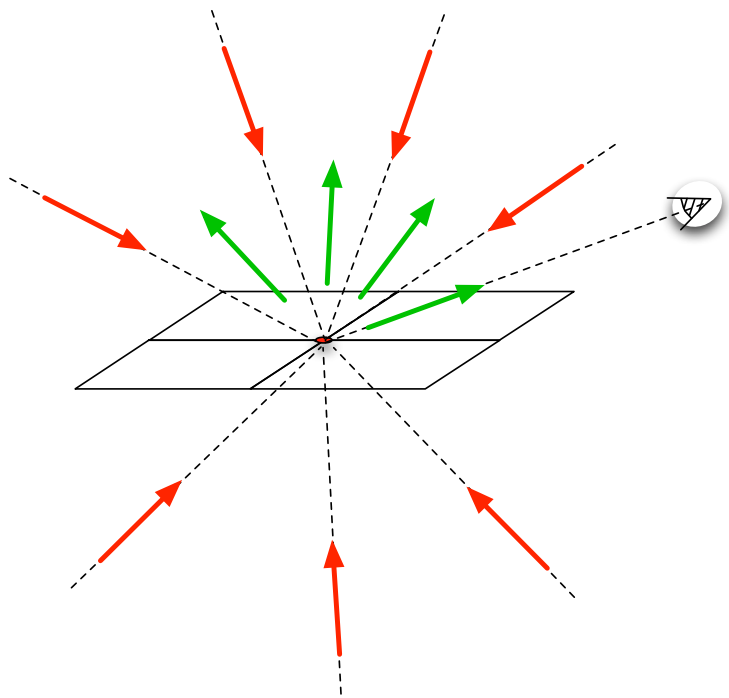
- 局所モデル
- 3 種類の光を考える。
  - 環境光（Ambient Light）
  - 拡散光（Diffuse Light）
  - 鏡面光（Specular Light）
- それぞれが独立に物体を照らす。
- それぞれの光は独立の色（RGB）を持つ。

物体の見え方（色や質感）の表現

- 3 つの色のそれぞれに対して頂点が独立に反射する。
  - 環境光反射（Ambient Reflection）
  - 拡散反射（Diffuse Reflection）
  - 鏡面反射（Specular Reflection）

環境光

- あらゆる方向‡からくる光による照明
- 間接照明の模擬
- 弱めに入れる。例： $(0.2, 0.2, 0.2, 1.0)$



‡面の裏側からも光が来ると考える。

環境光のシェーダコード

バーテックスシェーダ

```
uniform vec3 uAmbientLightColor;
varying vec3 vLightWeighting;

vLightWeighting = uAmbientLightColor;
```

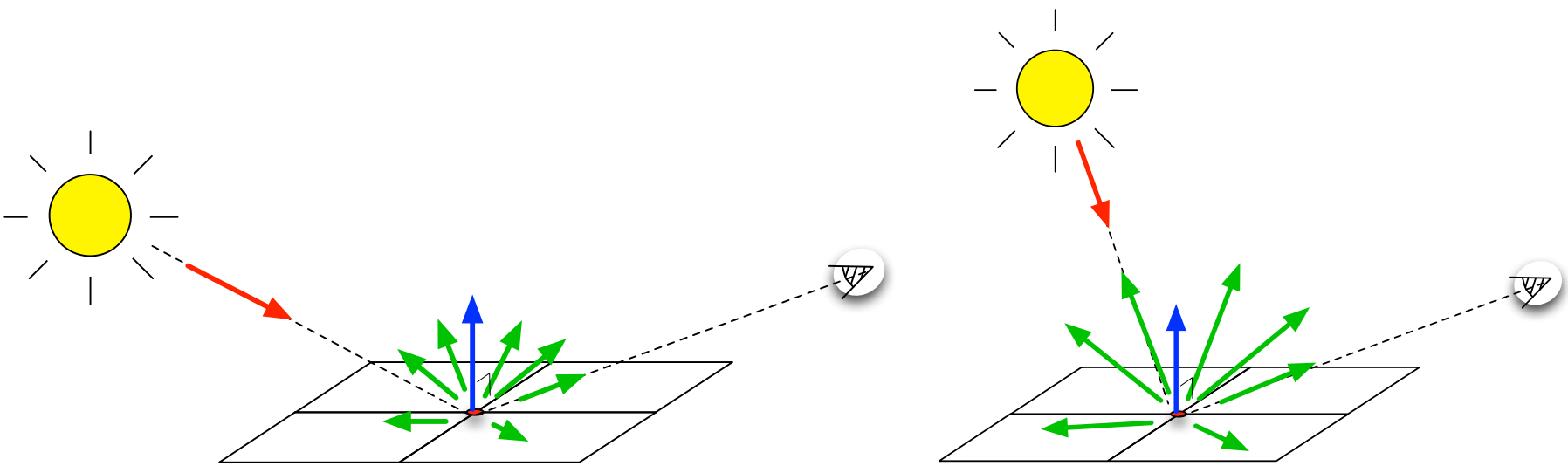
環境光のシェーダコード

フラグメントシェーダ

```
varying vec3 vLightWeighting;
vec4 texelColor = texture2D(uSampler, vTextureCoordinates);
gl_FragColor = vec4(vLightWeighting.rgb * texelColor.rgb,
texelColor.a);
```

拡散光

- 光の入射角に依存。
- 頂点の法線ベクトル $\boldsymbol{n}$ と、光の方向ベクトル $\boldsymbol{\ell}$ のなす角度で決まる。
- $\boldsymbol{n} \cdot \boldsymbol{\ell}$ の関数。
- カメラの方向 $\boldsymbol{e}$ には依存しない。



拡散光のシェーダコード

バーテックスシェーダ

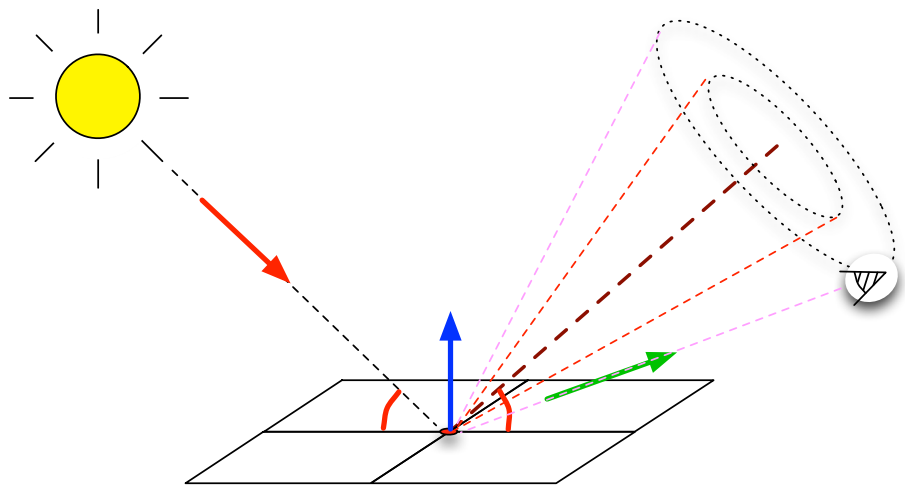
```
float diffuseLightWeightning = max(dot(normalEye,
     vectorToLightSource), 0.0);
vLightWeighting = uAmbientLightColor +
     uDiffuseLightColor * diffuseLightWeightning;
```

拡散光のシェーダコード

フラグメントシェーダ【変更なし】

鏡面光

- 金属のような光沢
- 頂点位置に鏡があるとして、入射する光が反射する方向にカメラがあるときに最も明るくなる。
- 頂点の法線ベクトル $\boldsymbol{n}$ と、光の方向ベクトル $\boldsymbol{\ell}$ と、カメラの方向 $\boldsymbol{e}$ の関数。
- 光沢度（鏡への近さ）を決めるパラメータを使う。反射する光がどれほど広がるか。



鏡面光のシェーダコード

バーテックスシェーダ

```
const float shininess = 32.0;
float rdotv = max(dot(reflectionVector, viewVectorEye), 0.0);
float specularLightWeightning = pow(rdotv, shininess);
vLightWeighting = uAmbientLightColor +
       uDiffuseLightColor * diffuseLightWeightning +
       uSpecularLightColor * specularLightWeightning;
```

鏡面光のシェーダコード

フラグメントシェーダ【変更なし】

サンプルコードを見る前に

様々な方向ベクトルのシェーダでの計算方法を解説。

頂点から光源へ向かう単位ベクトル

バーテックスシェーダ

```
// Get the vertex position in eye coordinates
vec4 vertexPositionEye4 = uMVMatrix * vec4(aVertexPosition, 1.0);
vec3 vertexPositionEye3 = vertexPositionEye4.xyz /
vertexPositionEye4.w;

// Calculate the vector (l) to the light source
vec3 vectorToLightSource = normalize(uLightPosition -
vertexPositionEye3);
```

法線ベクトル

法線ベクトルの変換行列は、モデルビュー行列（の 3 行 3 列部分）の転置の逆である[§]。

```
// Transform the normal (n) to eye coordinates
vec3 normalEye = normalize(uNMatrix * aVertexNormal);
```

[§]証明は次のページ

法線ベクトルの変換行列

面の法線ベクトルを $\boldsymbol{n}$, その面に接する任意のベクトルを $\boldsymbol{a}$ とする。

$$\boldsymbol{n} \cdot \boldsymbol{a} = \boldsymbol{n}^t \ \boldsymbol{a} = 0 \tag{1}$$

オブジェクトの変換行列を M、法線ベクトルの変換行列を N とすると、

$$\boldsymbol{a} \to \boldsymbol{a}' = M\boldsymbol{a}, \quad \boldsymbol{n} \to \boldsymbol{n}' = N\boldsymbol{n}$$

$\boldsymbol{n}'$ と $\boldsymbol{a}'$ が直交するから

$$0 = \boldsymbol{n}' \cdot \boldsymbol{a}' = (\boldsymbol{n}')^t \boldsymbol{a}' = (N\boldsymbol{n})^t \, (M\boldsymbol{a}) = \boldsymbol{n}^t N^t M \boldsymbol{a}$$

式 (1) より

$$N^t M = I$$

$$\therefore \quad N = (M^{-1})^t$$

M が直交行列なら $N = M$.

反射方向ベクトル

バーテックスシェーダに reflect という関数が組み込まれている。

```
// Calculate the reflection vector (r) that is needed for
specular light
vec3 reflectionVector = normalize(reflect(-vectorToLightSource,
normalEye));
```

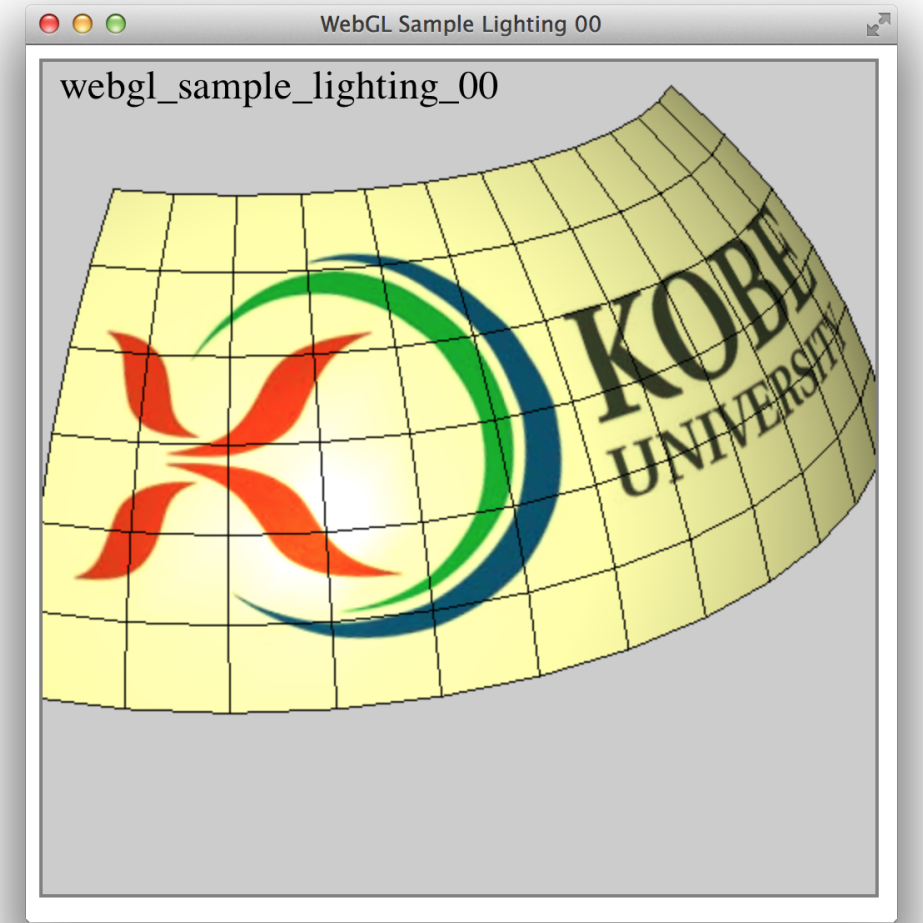
頂点からカメラ方向へのベクトル

原点（カメラ）からみた、頂点の位置ベクトルの逆。

```
vec3 viewVectorEye = -normalize(vertexPositionEye3);
```

サンプルコード 01

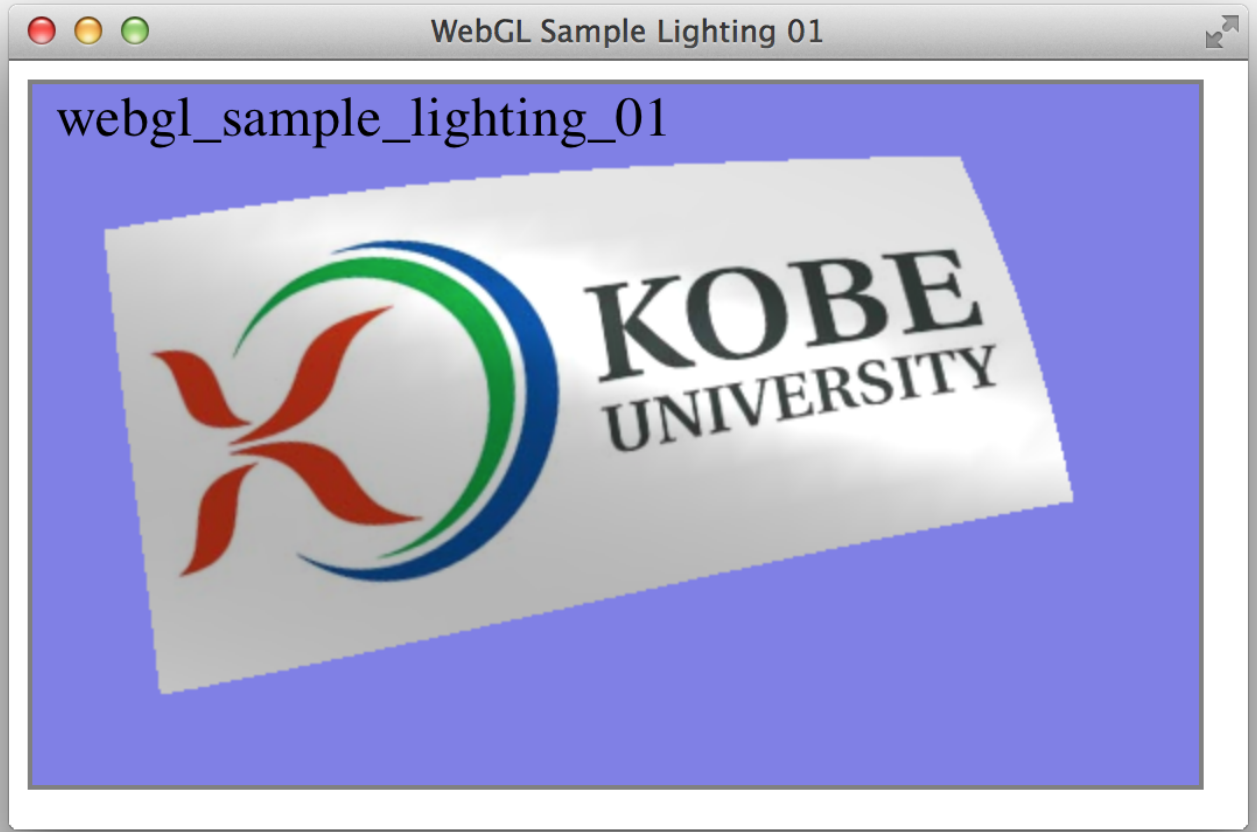
webgl_sample_lighting_00.html



サンプルコード 02

webgl_sample_lighting_01.html

頂点と法線の変形をバーテックスシェーダで計算している。



サンプルコード 03

webgl_spiral_kobe.html

レポート課題

- WebGL で任意の 3 次元物体を描け。
- 次の機能を使うとそれぞれ加点する。
  - アニメーション
  - テクスチャマップ
  - 照明
- いつものようにレポートと HTML ファイルをメールで提出。以下は必須。
  - 学籍番号と氏名、どのような物体を描いたか、描いた図形のキャプチャ図。
  - 自分の HTML ファイルをウェブで公開された場合の希望名（フルネーム／名字のみ／イニシャル[¶]）
- 締め切り： 7/16（火曜日）18:00
- いつものように何本か選んで講義のウェブページに掲載します[‖]。

[¶]イニシャルを希望する場合は、具体的に文字を指定してください。
[‖]最後のレポートなので、コメントを付ける予定。

索引

References