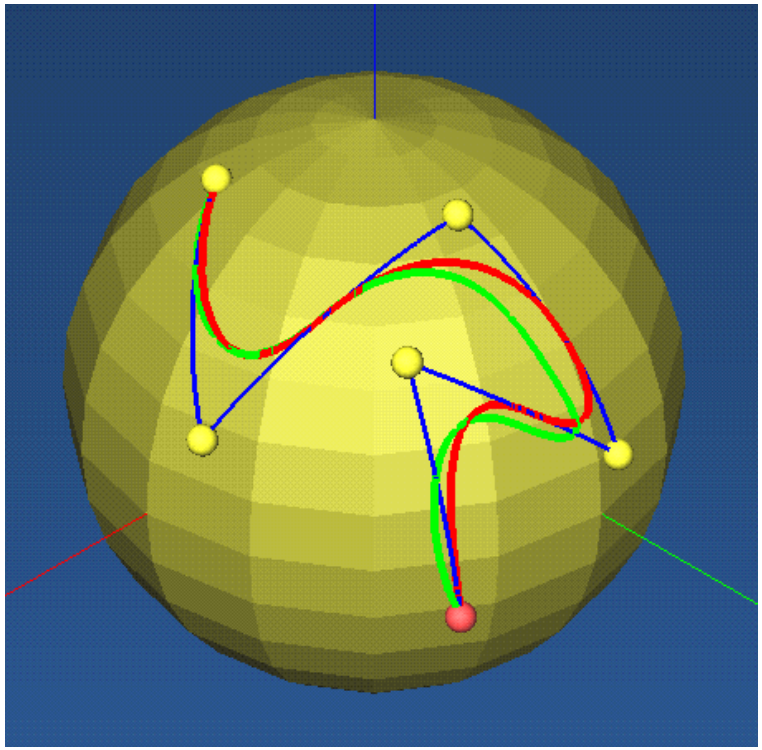
情報工学

2023年度後期　第6回 [11月29日]



静岡大学
工学研究科機械工学専攻
ロボット・計測情報講座
創造科学技術大学院
情報科学専攻

三浦　憲二郎

情報工学CGパート試験

- 日時　12月6日（水）　午前10:20～11:50
- 場所　工学部1号館32号室

講義アウトライン [11月29日]

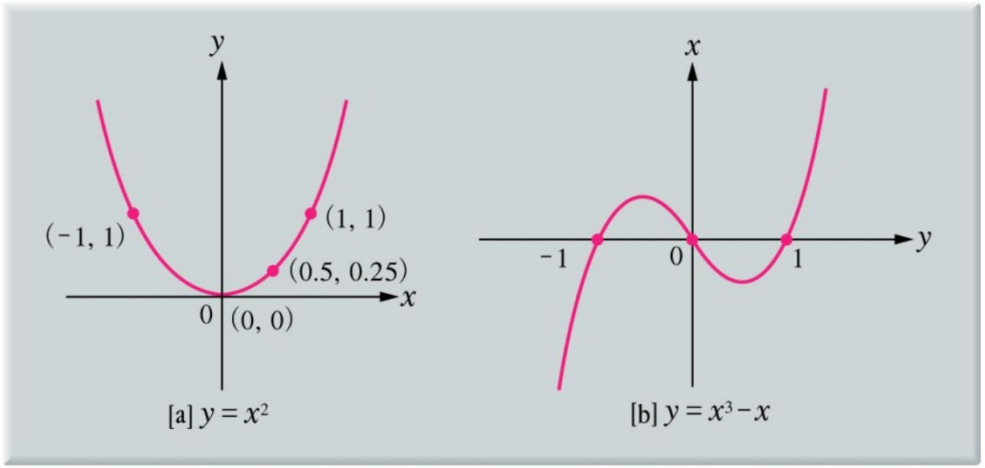
- ビジュアル情報処理
  - ２　モデリング
    - ２.３　曲線・曲面
- OpenGL
  - 色の取り扱い
  - シェーディング
  - 照明モデルと照光処理

拡散光
鏡面光
環境光

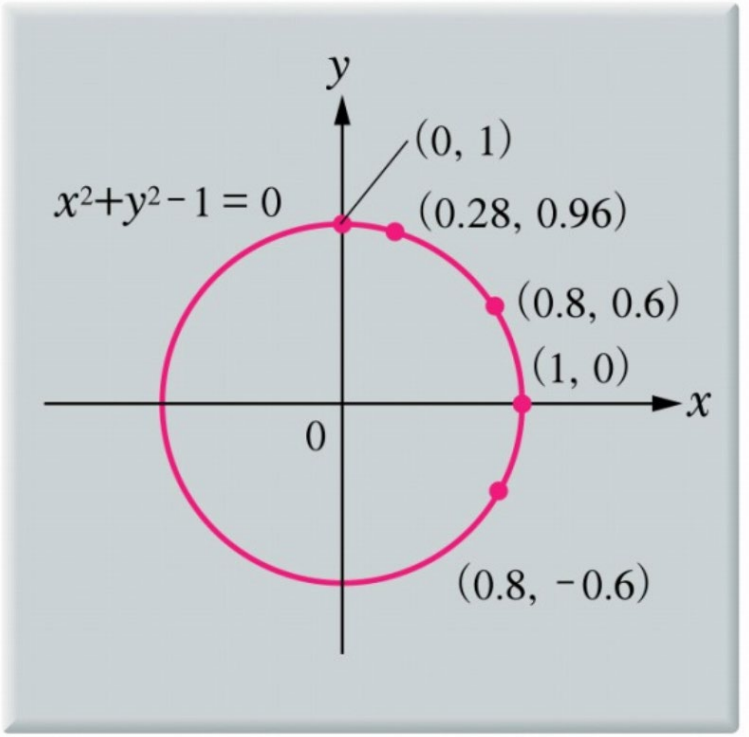
ビジュアル情報処理

- 2-3 曲線・曲面
  - 2-3-1 曲線の表現形式

■図3.9——陽関数表現の例



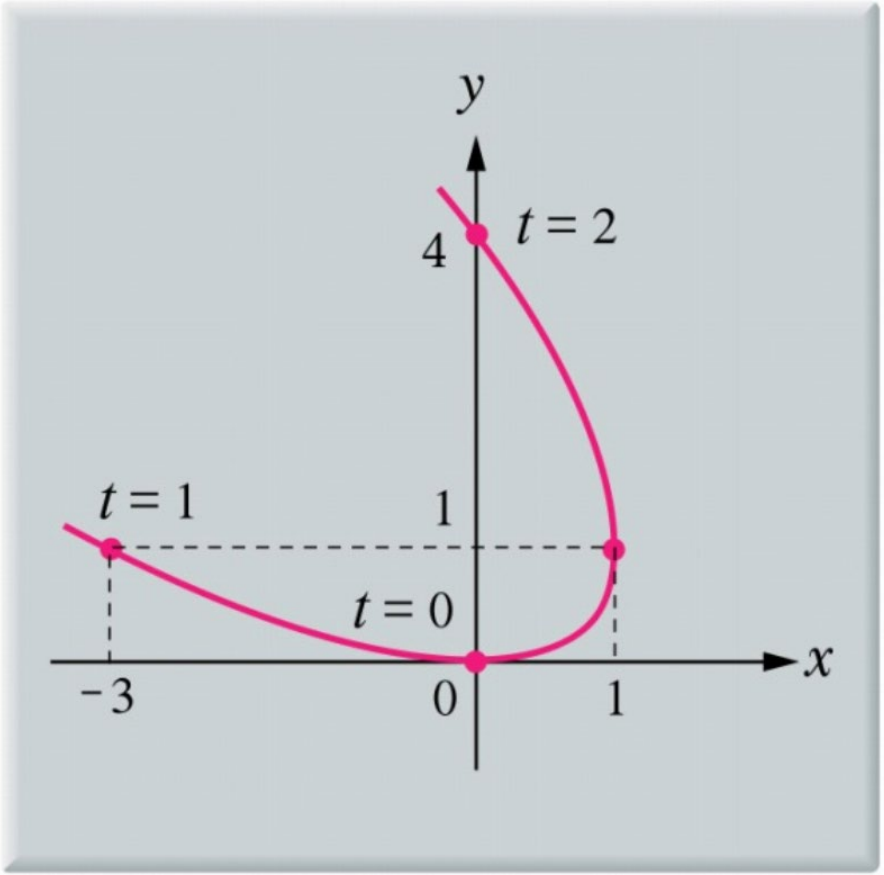
■図3.10——陰関数表現の例



ビジュアル情報処理

- 2-3-1 [2] パラメトリック表現

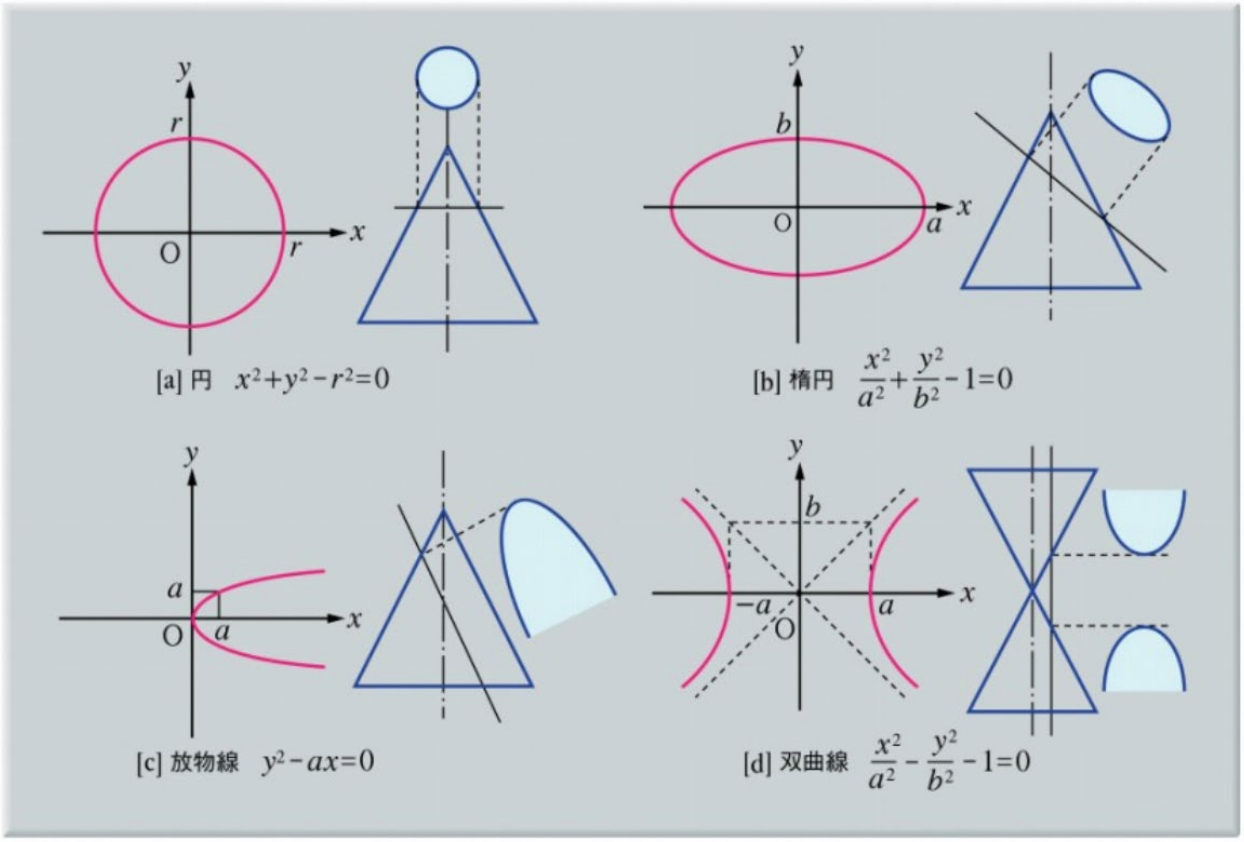
■図3.11——パラメトリック表現の例



ビジュアル情報処理

- 2-3-2 2次曲線

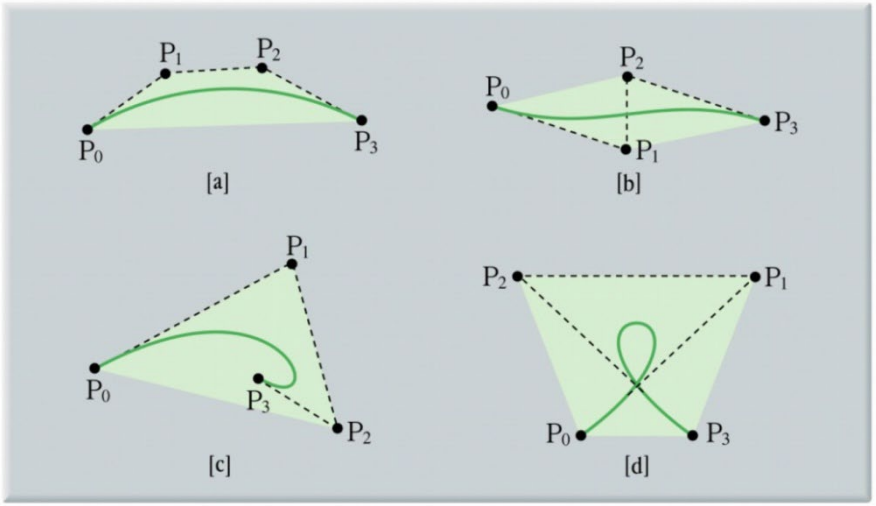
■図3.12——2次曲線の種類と円錐面との関係



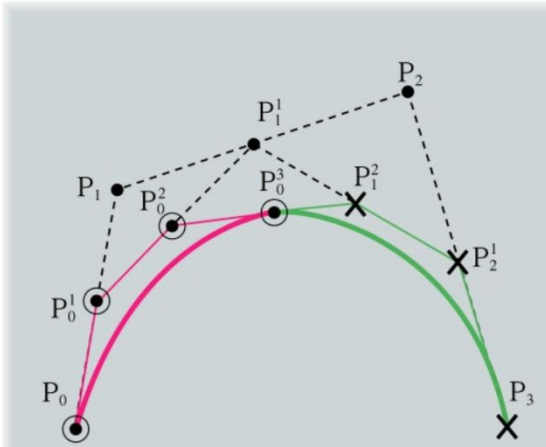
ビジュアル情報処理

•2-3-3 パラメトリック曲線　ベジエ曲線

■図3.13——3次ベジエ曲線の例



■図3.14——細分割法による3次ベジエ曲線の例

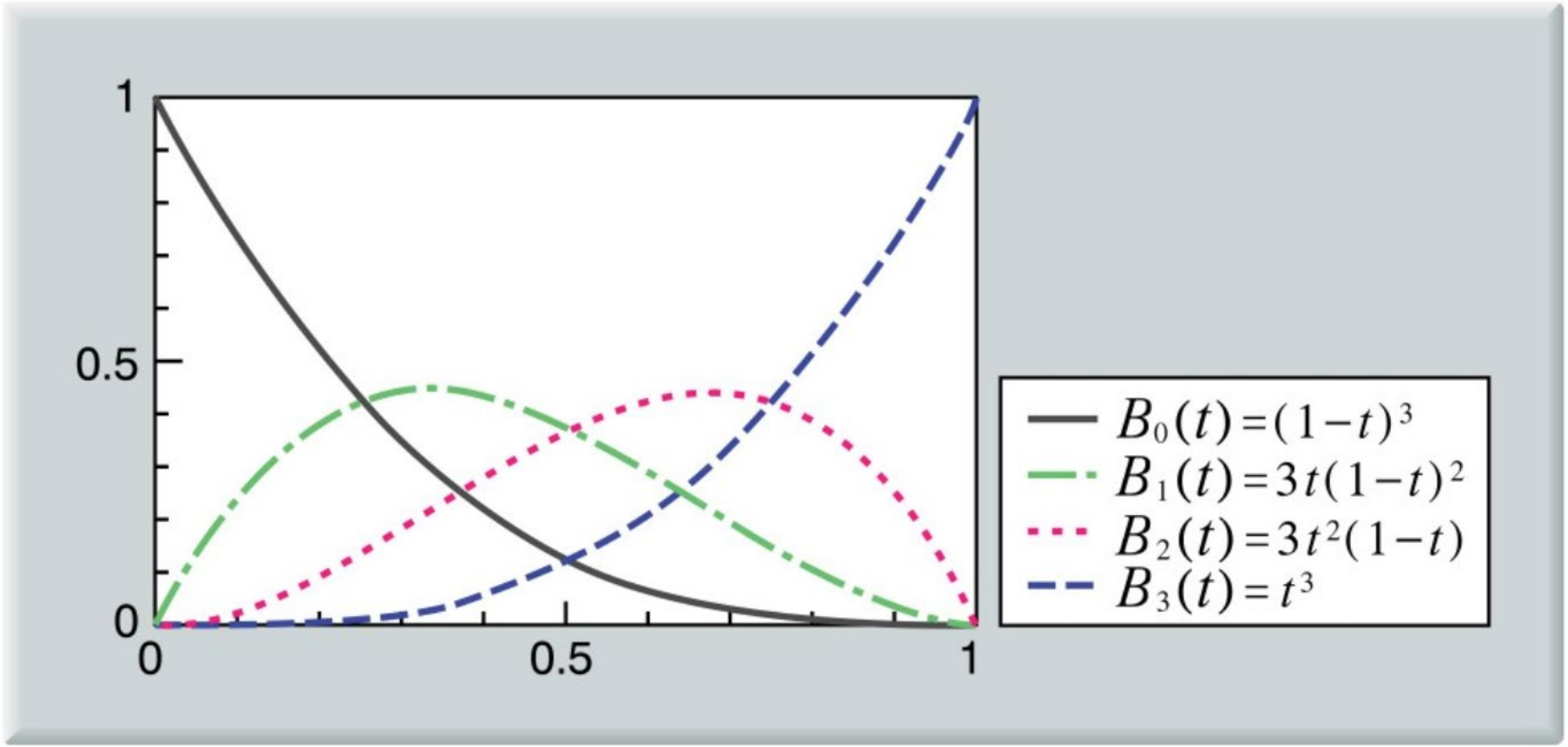


[1] まず，線分P_0P_1，P_1P_2，P_2P_3のそれぞれの中点を求め，順にP_0^1，P_1^1，P_2^1とする。
[2] つぎに，線分$P_0^1P_1^1$，$P_1^1P_2^1$のそれぞれの中点を求め，順にP_0^2，P_1^2とする。
[3] 最後に，線分$P_0^2P_1^2$の中点を求め，これをP_0^3とする。
[4] このとき，P_0^3はベジエ曲線上の点となり，この点で曲線を二分割できる。そして曲線の前半部分は4点P_0，P_0^1，P_0^2，P_0^3を制御点とするベジエ曲線となり，後半部分は4点P_0^3，P_1^2，P_2^1，P_3を制御点とするベジエ曲線となる。

ビジュアル情報処理

- 2-3-3 パラメトリック曲線　ベジエ曲線

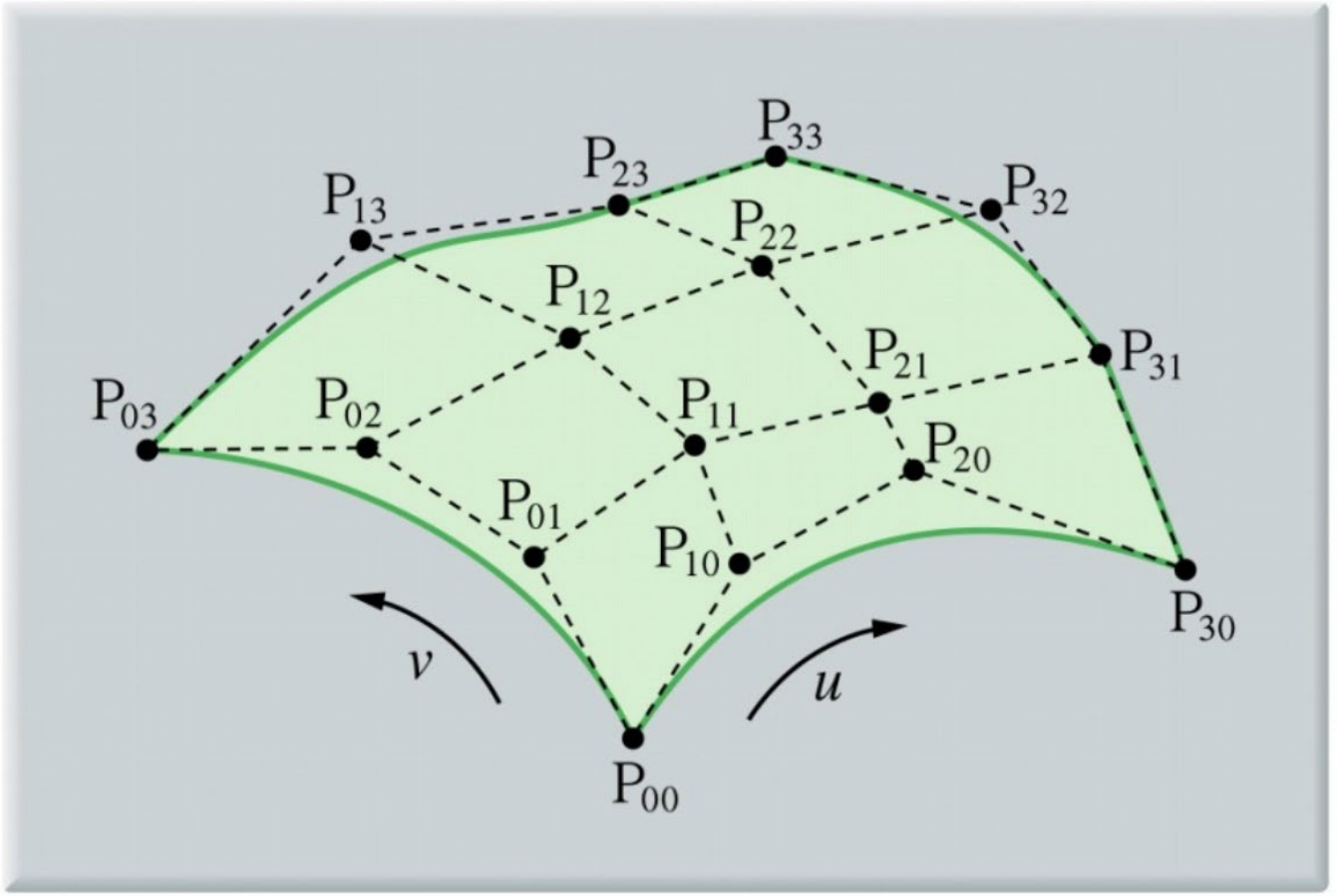
■図3.15——3次バーンスタイン基底関数のグラフ



ビジュアル情報処理

- 2-3-4 パラメトリック曲面　ベジエ曲面

■図3.17——双 3 次ベジエ曲面



OpenGL:色の取り扱い

1. RGBAモード

R, G, B, A ごとに値を指定

```
glutInitDisplayMode(GLUT_RGB);
```

2. カラーインデックスモード

各色に対応するインデックスナンバーを指定

```
glutInitDisplayMode(GLUT_INDEX);
```

RGBAモード

```
void
glColor3{b s i f d ub us ui}(TYPE r, TYPE g, TYPE b)

void
glColor4{b s i f d ub us ui}(TYPE r, TYPE g, TYPE b,
                             TYPE a)
```

b: byte, s: short, i: integer, f: float, d: double

ub: unsigned byte, us: unsigned short, ui: unsigned int

0<= float, double value <= 1

a: アルファ値　　色の「混合処理」,「変調処理」

シェーディング

1. フラットシェーディング

   立体の面や線を単一の色で描画する方法

2. スムースシェーディング

   複数の色を補間して描画する方法

```
void glShadeModel(GLenum mode)
mode: GL_FLAT, GL_SMOOTH
```

照明

1. 環境光 (ambient light)
   何度となく反射を繰り返し, 方向を特定できない光

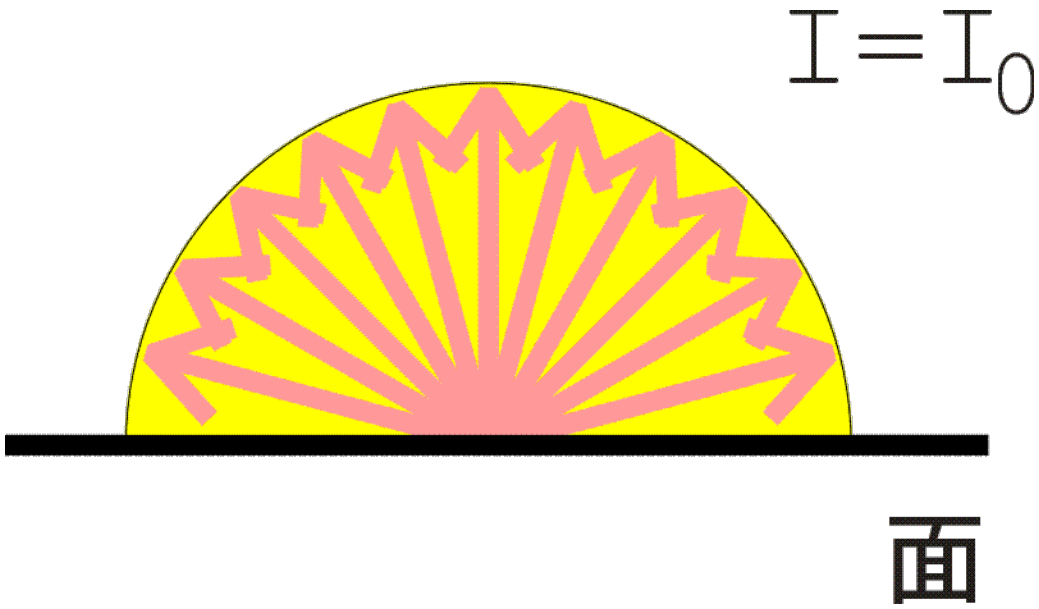
2. 拡散光 (diffuse light)
   すべての方向に均一に散乱する光

3. 鏡面光 (specular light)
   照明に対して特定の方向に強く反射する光

環境光

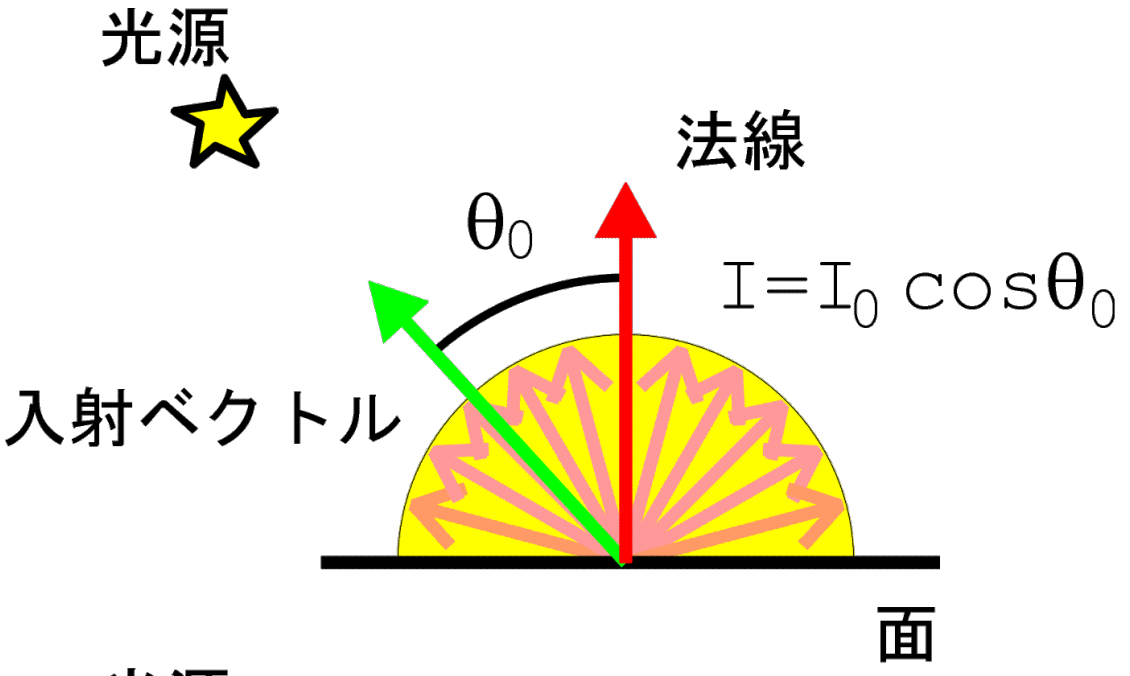
1. 環境光 (ambient light)

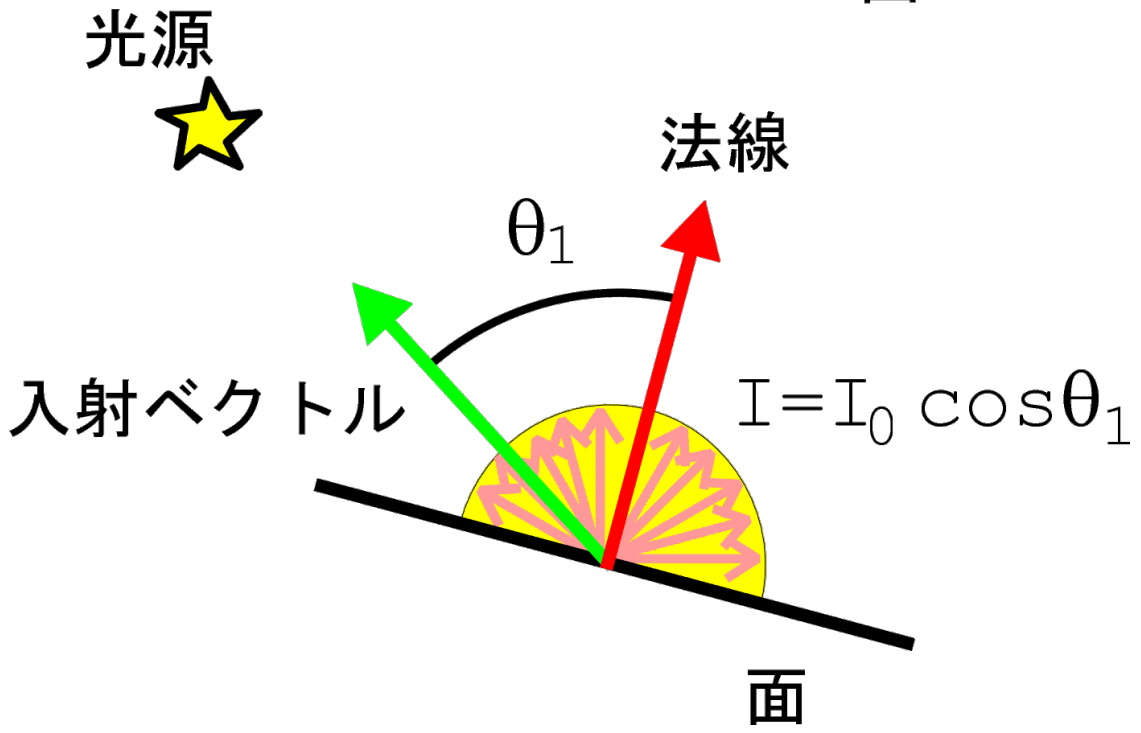
   何度となく反射を繰り返し, 方向を特定できない光



拡散光

2. 拡散光 (diffuse light)
   すべての方向に均一に
   散乱する光

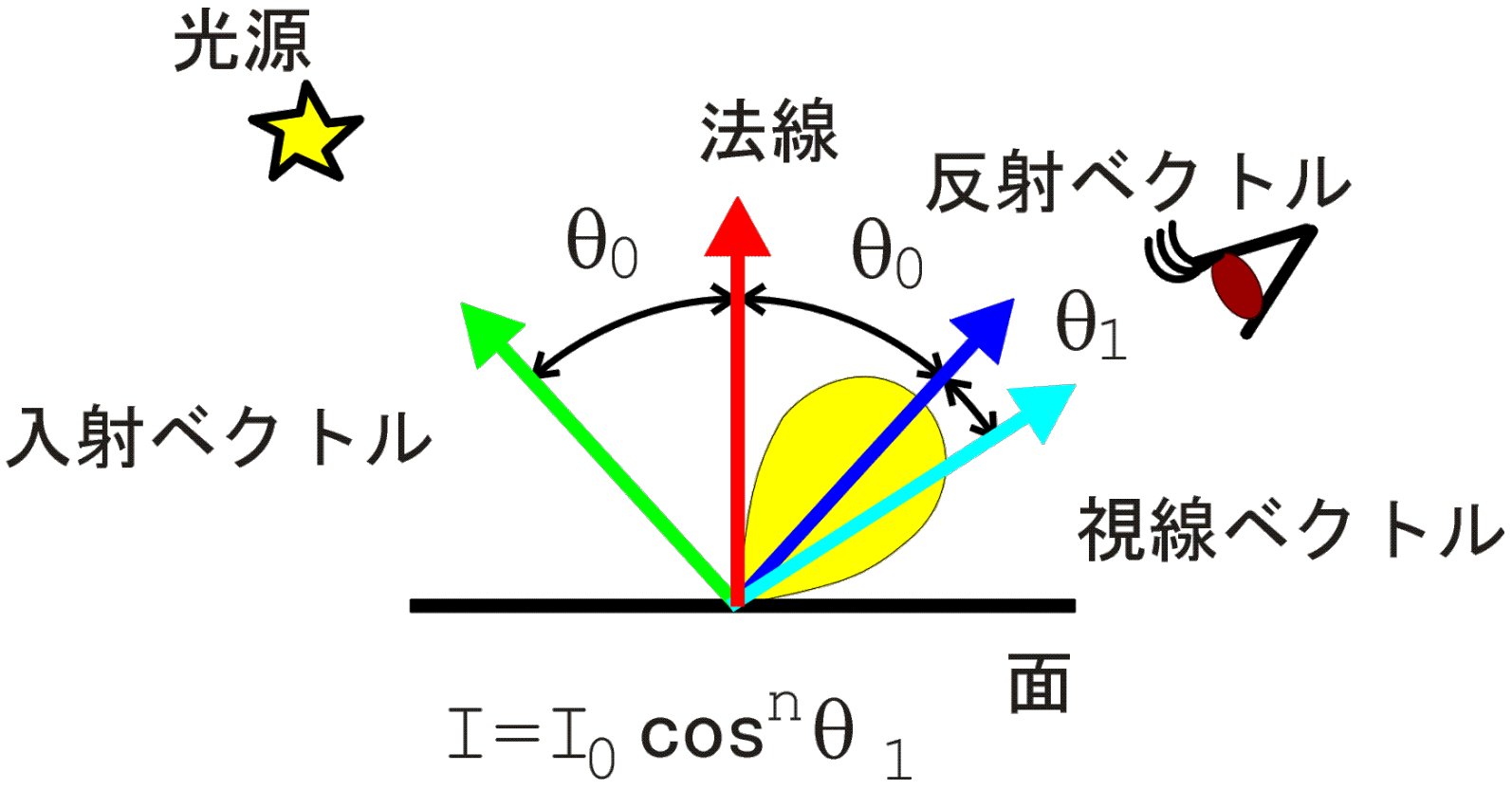




鏡面光

3. 鏡面光 (specular light)

照明に対して特定の方向に強く反射する光



照明の数と属性

GL_LIGHT0, GL_LIGHT1, … , GL_LIGHT7

少なくとも 8 個の照明が使える.

```
void
glLight{if}[v](GLenum light, GLenum pname,
                    TYPE param)
```

例

```
GLfloat color[] = {1.0f,0.0f,0.0f,1.0f};
glLightfv(GL_LIGHT0, GL_DIFFUSE, color);
```

glLight*()の引数 pname

パラメータ名	初期値	意味
GL_DIFFUSE	(1.0,1.0,1.0,1.0)	拡散光のRGBA値
GL_SPECULAR	(1.0,1.0,1.0,1.0)	鏡面光のRGBA値
GL_AMBIENT	(0.0,0.0,0.0,1.0)	環境光のRGBA値
GL_POSITION	(0.0,0.0,1.0,0.0)	照明の位置(x,y,z,w)

注意: 照明の位置の w 座標値が 0 の場合は無限遠
したがって，平行光線
0 でない場合は点光源

照光処理の有効化

有効化

```
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
/* glColor*()による色の指定は無効 */
```

無効化

```
glDisable(GL_LIGHTING);
/* glColor*()による色の指定が有効 */
```

光の減衰

平行光線の光は距離による輝度の減衰はない.

点光源の場合, 距離 d によって輝度が減衰する.

$$\text{減衰係数} = \frac{1}{k_c + k_l d + k_q d^2}$$

```
kc:GL_CONSTANT_ATTENUATION
kl:GL_LINEAR_ATTENUATION
kq:GL_QUADRATIC_ATTENUATION
```

材質の色

照明の 拡散光，鏡面光，環境光 に対する材質の特性

＋

放射光

```
void
glMaterial{if}[v](GLenum face, GLenum pname,
                  TYPE param)
```

例

```
GLfloat color[] = {1.0f,0.0f,0.0f,1.0f};
glMaterialfv(GL_FRONT, GL_DIFFUSE, color);
```

glMaterial*()の引数 pname

パラメータ名	初期値	意味
GL_DIFFUSE	(0.8,0.8,0.8,1.0)	拡散光のRGBA値
GL_SPECULAR	(0.0,0.0,0.0,1.0)	鏡面光のRGBA値
GL_AMBIENT	(0.2,0.2,0.2,1.0)	環境光のRGBA値
GL_SHININESS	0.0	鏡面の指数
GL_EMISSION	(0.0,0.0,0.0,1.0)	放射光のRGBA値

```
0.0 <= GL_SHININESS <= 128.0
```

面の法線ベクトルの指定

```
void
glNormal3{bsidf}(TYPE nx, TYPE ny,
                                TYPE nz)
```

b, s, i の場合は, 各値の範囲を[-1.0,1.0]に縮小

ベクトル版 `glNormal3{bsidf}v()`も存在する.

頂点カラーの計算

頂点カラー ＝ その頂点での材質からの放射 ＋
その頂点での材質の環境特性で測られる
グローバル環境光 ＋
適切に減衰した，すべての光源からの
環境，拡散，鏡面光

照光計算を実行した後，カラー値は[0,1]の範囲にクランプ
(RGBAモード)

材質放射

頂点カラー ＝ その頂点での材質からの放射 ＋
その頂点での材質の環境特性で測られる
グローバル環境光 ＋
適切に減衰した，すべての光源からの
環境，拡散，鏡面光

パラメータ GL_EMISSION に割り当てられたRGB値

```
glMaterialfv(GL_EMISSION, color);
```

グローバル環境光

頂点カラー ＝ その頂点での材質からの放射 ＋
その頂点での材質の環境特性で測られる
グローバル環境光 ＋
適切に減衰した，すべての光源からの
環境，拡散，鏡面光

グローバルな環境光 と
`glLightModelfv(GL_LIGHT_MODEL_AMBIENT, color0);`
材質の環境特性 の積
`glMaterialfv(GL_AMBIENT, color1);`
RGB値，別々に計算

光源からの影響

頂点カラー ＝ その頂点での材質からの放射 ＋
その頂点での材質の環境特性で測られる
グローバル環境光 ＋
適切に減衰した，すべての光源からの
環境，拡散，鏡面光

光源からの影響 ＝ 減衰係数 * スポットライト効果 *
(環境光の項 ＋ 拡散光の項 ＋ 鏡面光の項)

環境，拡散，鏡面光の項:
照明の各成分 と 材質の各成分 の 積

rotCube.cのinitLights()

```
void
initLights(void)
{
    GLfloat light0_diffuse[] = {0.9, 0.9, 0.9, 1.0}; /* 拡散成分 */
    GLfloat light1_diffuse[] = {0.5, 0.5, 0.5, 1.0}; /* 拡散成分 */
    GLfloat light_specular[] = {0.3, 0.3, 0.3, 1.0}; /* 鏡面成分 */
    GLfloat lmodel_ambient[] = {0.2, 0.2, 0.2, 1.0}; /* 周囲光 */

    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, lmodel_ambient );

    glLightfv ( GL_LIGHT0, GL_DIFFUSE, light0_diffuse );
    //glLightfv ( GL_LIGHT1, GL_SPECULAR, light_specular );

    glLightfv ( GL_LIGHT1, GL_DIFFUSE, light1_diffuse );
    glLightfv ( GL_LIGHT1, GL_SPECULAR, light_specular );

    glLightfv ( GL_LIGHT2, GL_DIFFUSE, light1_diffuse );
    glLightfv ( GL_LIGHT2, GL_SPECULAR, light_specular );

    glEnable ( GL_LIGHTING );
    glEnable ( GL_LIGHT0 );
    glEnable ( GL_LIGHT1 );
    glEnable ( GL_LIGHT2 );
}
```

rotCube.cのourDisplay()

```
void
ourDisplay(void)
{
  GLfloat material_color[4]={1.0, 0.0, 0.0, 1.0};/*拡散光成分*/
  GLfloat material_specular[4]={0.2, 0.2, 0.2, 1.0};/*鏡面光成分*/

  /* バッファのクリア */
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

  /* 鏡面光成分のセット */
  glMaterialfv(GL_FRONT, GL_SPECULAR, material_specular);
  ...
  glLightfv ( GL_LIGHT0, GL_POSITION, light_position0 );
  glLightfv ( GL_LIGHT1, GL_POSITION, light_position1 );
  glLightfv ( GL_LIGHT1, GL_POSITION, light_position2 );
  ...
  /* レッド */
  glMaterialfv(GL_FRONT, GL_DIFFUSE, material_color);
  drawFModel( &fCube );
  glFlush();
}
```

まとめ

- ビジュアル情報処理
  - 2　モデリング
    - 2.3　曲線・曲面
- OpenGL
  - 色の取り扱い
  - シェーディング
  - 照明モデルと照光処理