

# Graphics with Processing



2018-10 照明と材質のモデル

<http://vilab.org>

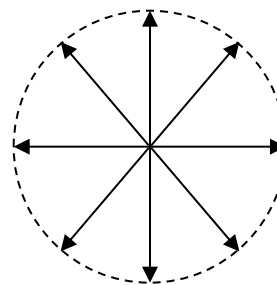
塩澤秀和

# 10.1\* レンダリングと光源

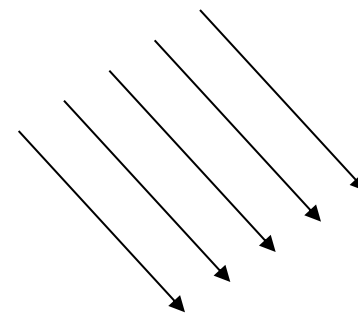
## レンダリング (p.122)

- 座標変換後の画像生成
  - 3次元シーン → 2次元画像
  - 色, 陰影, 質感などの表現
  - 光学現象のシミュレーション
  - 高品質 vs リアルタイム
- レンダリング関連技術
  - 隠面消去
  - ライティングとシェーディング
  - マッピング (テクスチャ・法線等)
  - 影付け, など...
- 高品質CGのレンダリング
  - レイトレーシング
  - 大域照明
  - ボリュームレンダリング, など...

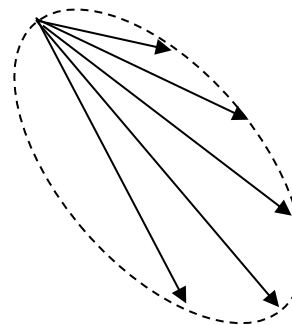
## 光源の種類 (p.144)



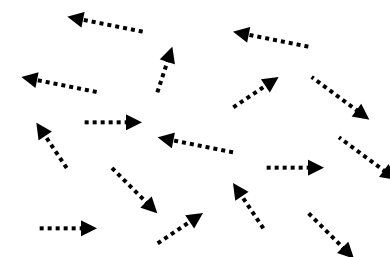
点光源  
(電球など)



方向光  
(太陽光など)



スポットライト



環境光 (壁などに  
何回も反射した  
間接光のモデル化)

# 10.2\* 照明の効果

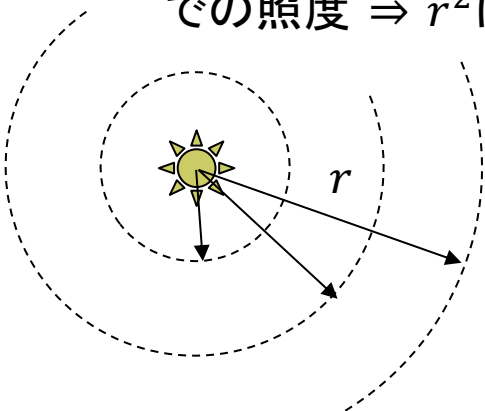
## 光源からの照明

### □ 照明の色

- 太陽光・蛍光灯 ⇒ 白
- 白熱電球 ⇒ 白っぽいオレンジ

### □ 照明の明るさ

- 光束＝光源から発する光の量  
(単位ルーメン lm)
- 照度＝単位面積あたりに当たる  
光の量(単位ルクス lx)
- 点光源から距離  $r$  離れた場所  
での照度 ⇒  $r^2$ に反比例



光源を中心とする  
球の表面積は  
 $S = 4\pi r^2$

## 反射と色の関係

入射光

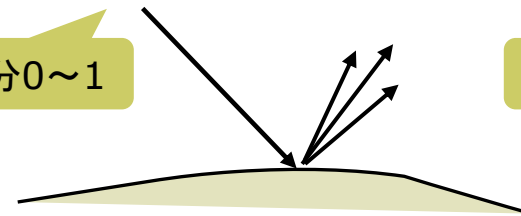
$$L = (L_R, L_G, L_B)$$

反射光

$$I = (I_R, I_G, I_B)$$

各成分0～1

各成分0～1



表面色(反射率)

$$k = (k_R, k_G, k_B)$$

物体の色＝白色光の反射率(各成分0～1)

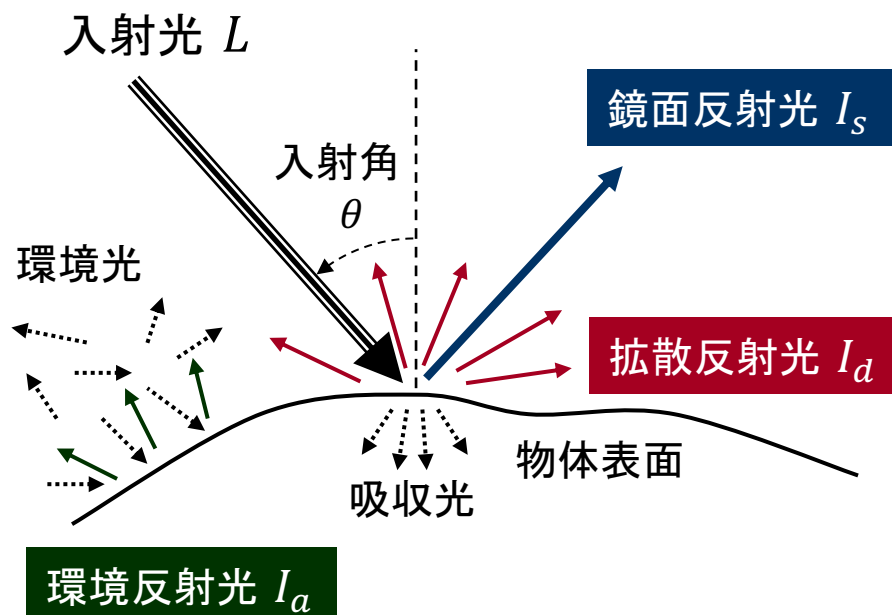
### □ 反射光＝表面色 × 入射光

- 色の見え方は照明にも影響される
- 光のスペクトル(RGB)ごとに計算  
⇒ ベクトルの「要素ごとの積」

$$(I_R, I_G, I_B) = (k_R L_R, k_G L_G, k_B L_B) \quad 3$$

# 10.3\* 反射光のモデル

## 反射光のモデル(p.141)



### 観測される色

- 各反射光(+放射光)の総和

$$I = I_d + I_s + I_a + I_e$$

※ それぞれにRGB成分があることに注意

## 反射光の種類

- 拡散反射光 ( $I_d$ : diffuse)
  - 物体表層で複雑に反射・透過・屈折することで拡散した光
  - 光の入射角に依存  $\Rightarrow$  立体感
- 鏡面反射光 ( $I_s$ : specular)
  - 物体表面に並んだ分子で鏡のようにきれいに反射した光
  - 見る角度に依存  $\Rightarrow$  “光沢”
- 環境反射光 ( $I_a$ : ambient)
  - 特定の光源ではなく, 空間全体の間接光に対する反射光
  - シーン全体が一様に照らされる
- 放射光 ( $I_e$ : emissive)
  - 物体自体からの発光
  - 周囲に関係なく一定の明るさ

# 10.4\* 材質属性のモデル

## 材質(マテリアル)属性

### □ “色”の設定

- 反射・吸収される光の波長は、物体表面の材質によって違う
- 白色光に対する反射スペクトル(分光分布)で材質をモデル化

### □ 拡散反射色( $k_d$ )

- 拡散反射の反射率(RGB)
- 表面が粗いほど、特定の波長が吸収され、残りの光が拡散する
- 通常の意味での物体の色

### □ 鏡面反射色( $k_s$ )

- 鏡面反射の反射率(RGB)
- 表面が滑らかだと、すぐに反射して着色の少ない光が増える
- 金属光沢, ハイライト, つや

### □ 環境反射色( $k_a$ )

- 環境光に対する反射率(RGB)
- 通常は拡散反射色と同じ色

### □ 放射光( $k_e$ )

- 電球などの発光スペクトル(RGB)
- 周囲に関係なく一定の色になる

$$I_e = k_e \quad (\text{常に一定の色})$$

## 材質による特徴

### □ 紙・木など

- 鏡面反射(光沢)がほとんどない

### □ プラスチックなど

- 若干の鏡面反射によるつやがある

### □ 金属など

- 強く白っぽい鏡面反射( $k_s \neq k_d$ )

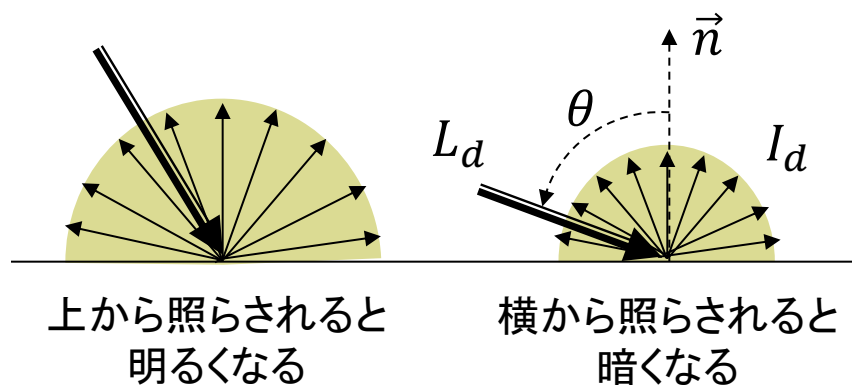
# 10.5\* 反射光の計算モデル

## 拡散反射光(p.144)

### □ ランベルト(Lambert)の余弦則

- 光がどの方向から入射しても、全方向に均等に拡散する場合
- 入射角余弦の法則より、表面の明るさは入射角のcosに比例

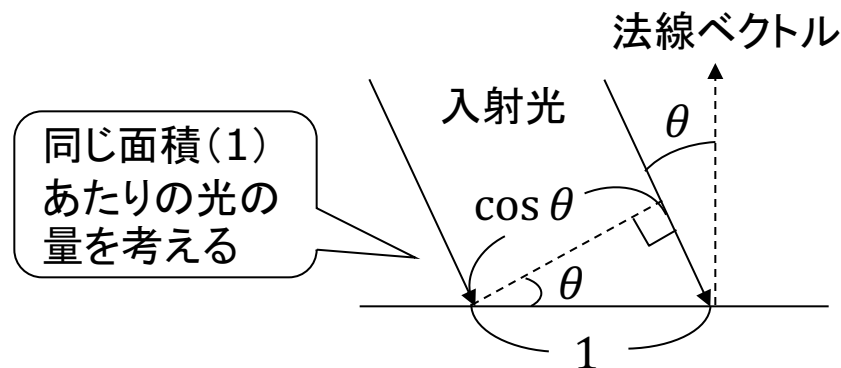
$$I_d = k_d L_d \cos \theta$$



$L_d$ : 入射光(拡散光用成分)     $I_d$ : 反射光  
 $k_d$ : 物体表面の拡散反射率     $\theta$ : 入射角

### □ 入射角余弦の法則

- 単位面積あたりに当たる入射光の量は入射角のcosに比例



## 環境反射光(p.144)

### □ 環境光による拡散反射光

- 環境光( $L_a$ )は、四方八方から均等に当たるので方向がない
- どこからでも同じ色に見える

$$I_a = k_a L_a \quad (k_a: \text{環境光の反射率})$$

## 10.6 照明と材質の関数

### 基本的な光源

- `pointLight(r, g, b, x, y, z)`
  - 点光源(例:電球)
  - $r, g, b$ : 光の色(HSBモードの場合は, 色相, 彩度, 明度)
  - $x, y, z$ : 光源の座標
- `directionalLight(r, g, b, nx, ny, nz)`
  - 方向光(例:太陽光, 天井照明)
  - $nx, ny, nz$ : 光の方向ベクトル
- `ambientLight(r, g, b)`
  - 環境光(間接光のモデル化)
  - 全方向から均等にあたる光
- サンプル
  - 3D (Basics) → Lights
  - **物体をおく前に, 光源をおくこと**

### 標準の光源

- `lights()`
  - 下記の光源を設定
  - `ambientLight(128, 128, 128)`
  - `directionalLight(128, 128, 128, 0, 0, -1)`

### 基本的な材質特性

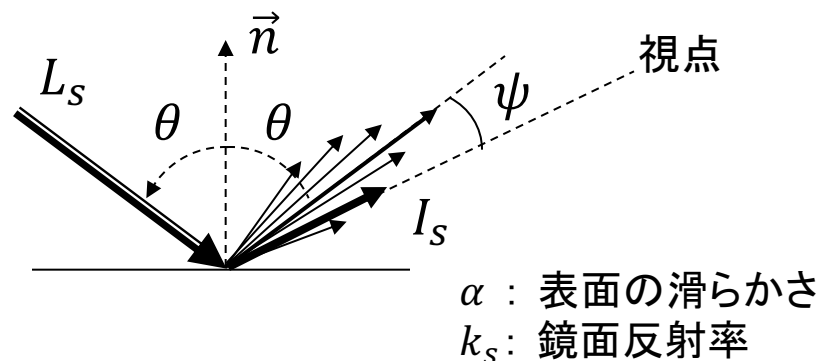
- `fill(色)`
  - 通常の色 = 拡散反射率( $k_d$ )
- `ambient(色)`
  - 環境反射率( $k_a$ )の設定
  - 無指定時にはfillと同じ色で計算
- `emissive(色)`
  - 放射光( $k_e$ )の設定(蛍光面)

# 10.7\* 光沢の表現

## 鏡面反射光(p.148)

- フォン(Phong)の反射モデル
  - 光がごく表層でほぼ完全に反射  
⇒ 反射光が正反射方向に集中
  - 従来, フォンのモデルで近似

$$I_s = k_s L_s \cos^\alpha \psi$$



- より正確なモデル(p.149)
  - ブリンの反射モデル
  - クック・トランスの反射モデル

## 鏡面反射の材質特性

- specular(色)
  - 鏡面反射率( $k_s$ )
- shininess(輝き)
  - 鏡面反射光の集中度( $\alpha$ )
  - 輝き: 10~50~500(金属)

## 光源のパラメータ

- lightSpecular(r, g, b)
  - 後に設置する光源に鏡面反射光用の成分( $L_s$ )を追加
  - 通常は光源と同じ色でよい
- lightFallOff(c1, c2, c3)
  - 光の減衰のしかたを変更する
  - 距離 $d$ として

$$\frac{1}{c_1 + c_2 d + c_3 d^2}$$



## 10.8 照明と材質の設定例

```
void draw() {  
  float a = radians(frameCount);  
  background(0);  
  perspective();  
  camera(0, -100, 200, 0,0,0, 0,1,0);  
  ambientLight(50, 50, 50); // 環境光  
  
  // 回転する点光源  
  float lx = 100 * cos(a);  
  float ly = lx - 120;  
  float lz = 100 * sin(a);  
  // ボタンを押すと鏡面反射成分を除く  
  if (!mousePressed)  
    lightSpecular(128, 128, 128);  
  pointLight(128, 128, 128, lx, ly, lz);  
  
  stroke(128);  
  line(lx, 0, lz, lx, ly, lz);  
  noStroke();  
  rotateY(a / 8);
```

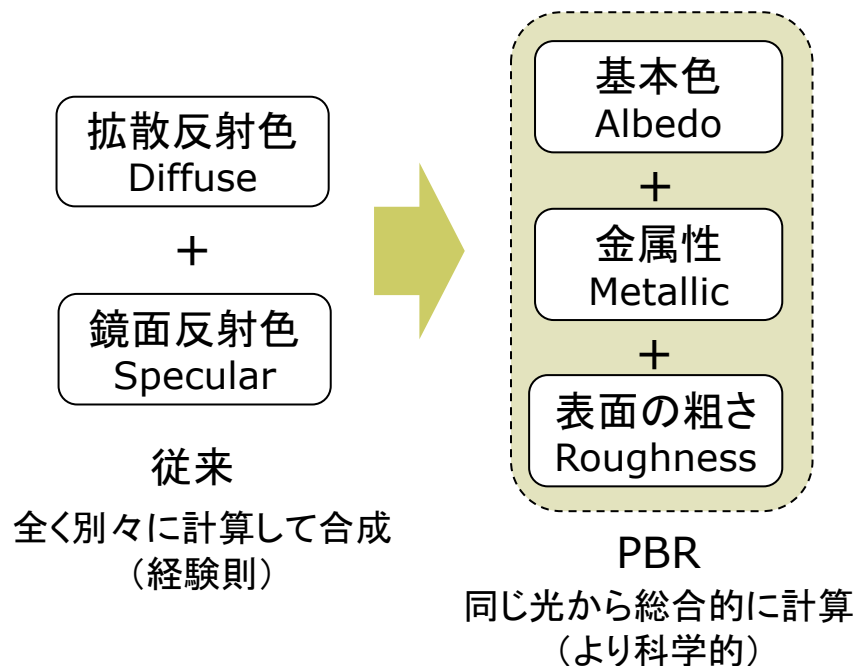
```
  pushMatrix();  
  translate(-40, -20, 0);  
  fill(0, 255, 0); specular(0);  
  sphere(20); // 鏡面反射のない球  
  translate(80, 0, 0);  
  specular(100, 100, 100); // 鏡面反射色  
  shininess(200 * mouseX / width);  
  sphere(20); // 鏡面反射のある球  
  popMatrix();  
  
  fill(0, 0, 255); // 鏡面反射色はついたまま  
  beginShape(TRIANGLES);  
  for (int x = -100; x <= 100; x += 10) {  
    for (int z = -100; z <= 100; z += 10) {  
      vertex(x, 0, z); vertex(x + 10, 0, z);  
      vertex(x + 10, 0, z + 10);  
    }  
  }  
  endShape();  
}
```

# 10.9\* 物理ベースレンダリング (PBR)

## Physically based rendering

- 光学現象をより忠実に再現
  - 厳密な定義はない言葉？
  - 最近ゲームなどで急速に採用

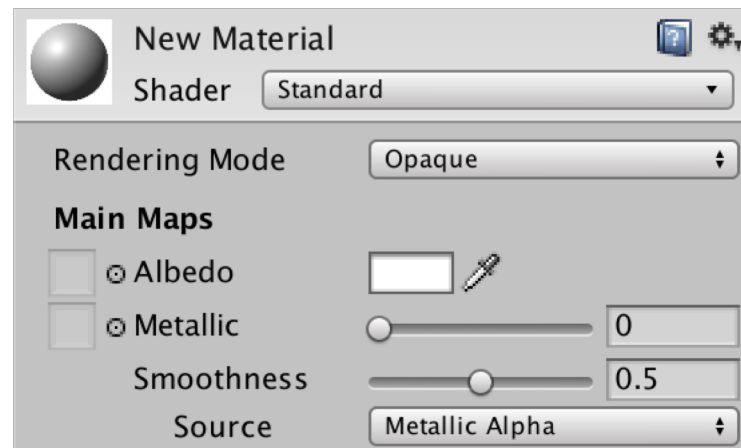
## □ 物理ベースの材質モデル



## □ 材質のマッピング

- テクスチャマッピング (次回)  
表面に模様画像を貼り付ける  
(基本色の分布のマッピング)
- 物理パラメータのマッピング  
金属性や粗さの分布を貼り付ける
- その他, 透過や法線 (微細な凹凸)

## □ 例: Unityの材質設定 (一部)



# 10.10 演習課題

## 課題

問1) ピンク色の紙に斜め45度から青白い光を照らすとどのような色に見えるか, 拡散反射光のRGB値を計算して考察せよ

- 色のRGB値は適当に設定せよ

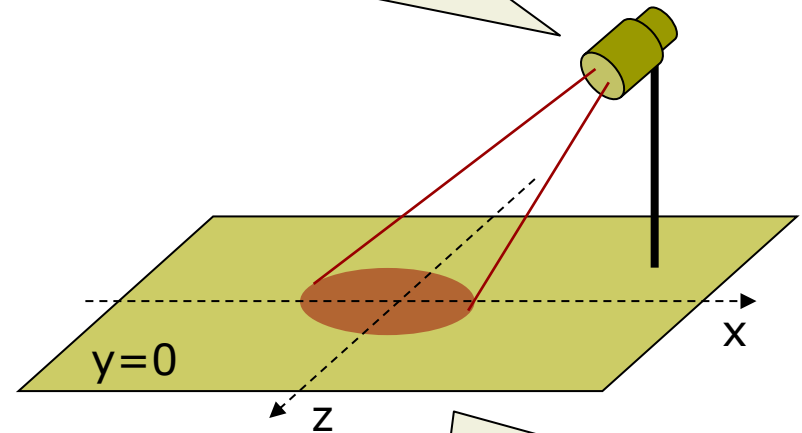
問2) スポットライト(下記)を使用したシーンを工夫して作成せよ

- 床など広い面は, タイルを敷き詰めるようにする(理由は次回)
- 環境光も少し照らすとよい

## スポットライト関数

- `spotLight(r, g, b, x, y, z, nx, ny, nz, 角度, 集中度)`
  - 角度: 光の範囲( $\sim \pi/2$ 程度)
  - 集中度: 1 $\sim$ 100 $\sim$ それ以上

```
// スポットライトの設置例
spotLight(255, 0, 0, 50, -50, -50,
-1, 1, 1, PI/2, 100)
```



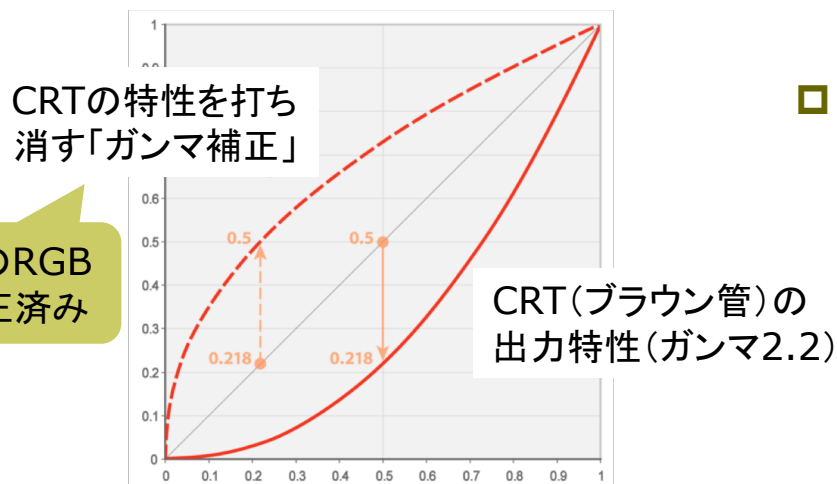
```
// 床の描画例
noStroke();
beginShape(QUADS);
for (x = -100; x < 100; x += 10) {
  for (z = -100; z < 100; z += 10) {
    vertex(x, 0, z);
    vertex(x, 0, z + 10);
    vertex(x + 10, 0, z + 10);
    vertex(x + 10, 0, z);
  }
}
endShape();
```

# 10.11 参考:PBRの使用技術

## 主な使用技術

### □ リニア(線形)色空間の利用

- 標準規格のRGBの値は、昔のCRT(ブラウン管)の特性で階調補正(ガンマ補正)がされている  
⇒ 暗い色が増強
- 物理的に正しい光の強さで計算するために、逆補正して計算後、表示時に再補正する



### □ HDRレンダリング

- HDR=High Dynamic Range
- 人間の広範囲な輝度の知覚に対応するため、RGB値に実数(16ビットで0~10<sup>12</sup>)を採用
- 薄明かりで細やかな描写
- 高輝度領域を抽出し、明るい光がにじむような処理も(ブルーム/グレア/グロー/光芒/閃光)

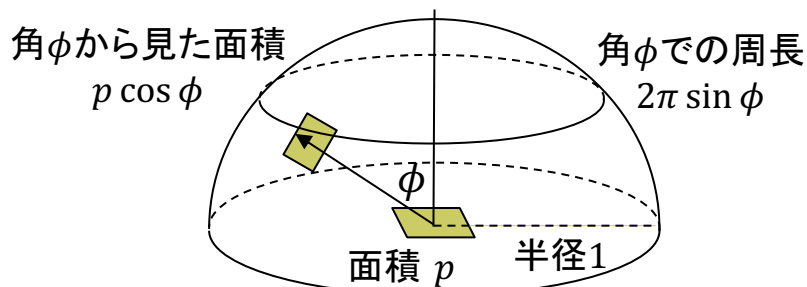
### □ IBL: 光源画像による照明

- IBL=Image Based Lighting
- 空、景色、窓などの実写画像に基づいて照明効果を計算  
⇒ 金属等への映り込みの表現
- 画像から光源の位置等を推定

# 10.12 参考:PBRの使用技術

- 光エネルギー保存則の考慮
  - 任意の点で単位面積当たり  
反射光の総量  $\leq$  入射光の総量
  - 例: 拡散反射率の正規化

表面の面積  $p$  を, それを覆う半球を通して  
見た面積を(全方向からの分)合計すると



$$\int_0^{\pi/2} p \cos \phi \cdot 2\pi \sin \phi d\phi = \pi p$$

反射光は $\pi$ 倍の面積に広がるのと同様なので,  
ある点に届く反射光は, 最大でも入射光の $1/\pi$

$$\pi \text{で割って正規化} \Rightarrow I_d = \frac{k_d}{\pi} L \cos \theta$$

- BRDF(双方向反射率分布関数)
  - ある方向から来た光が, ある点で,  
ある方向に反射する割合を表す
  - 保存則を考慮した反射率の一般化

簡単なBRDF

$m$ は「金属性」

$$= (1-m) \times \text{拡散反射率} + m \times \text{鏡面反射率}$$

- 高度な鏡面反射モデル
  - クック・トランスのモデルなど

$$\text{鏡面反射率 } R_s = \frac{F}{\pi} \frac{DG}{(\vec{n} \cdot \vec{L})(\vec{n} \cdot \vec{V})}$$

$F$ : フレネル項  $D$ : 微小面分布関数  $G$ : 幾何減衰係数  
 $\vec{n}$ : 法線ベクトル  $\vec{L}, \vec{V}$ : 光源および視点へのベクトル

- フレネル反射: 水平に近い入射光  
に対して反射率が高くなる現象
- 「表面の粗さ」の影響もモデル化