

# 計算機合成ホログラムのための点光源データ生成ソフトの開発

## The development of the point light source data generation software for the computer generated hologram

○上野凌<sup>1</sup>, 山口健<sup>2</sup>, 吉川浩<sup>2</sup>\*Ryo Ueno<sup>1</sup>, Takeshi Yamaguchi<sup>2</sup>, Hiroshi Yoshikawa<sup>2</sup>

In our laboratory, we are developing a software that generates a point light source data from the CG data in order to produce a computer-generated hologram (below CGH). Conventional soft have been prepared using the OpenGL in C++. The software presents an image of the object obtained by observing point and parameters can not be changed interactively. In this research, we have developed an interactive data generation system which can create data from multiple view points. Further to enrich the user interface the language is changed to C#

### 1. まえがき

当研究室では、計算機合成ホログラム (以下 CGH) を作製するために CG データから点光源データを生成するソフトの開発を行っている. 従来のソフト<sup>[1]</sup>はC++の OpenGL<sup>[2]</sup>を用いて作成されている. このソフトでは、単視点からオブジェクトを観察した画像のみ表示されているため、オブジェクトの周囲の環境がわかりにくい. そのため CGH のパラメータが計算しづらいという問題点があった.

本研究では、オブジェクトの周囲の環境を可視化するためにシーン機能の追加を行う. また、ユーザインターフェースを充実させ、操作性の向上を図るため C# に変更する.

### 2. 原理・方法

#### 2.1. CGH

CGH は、ホログラムの記録過程をコンピュータ上で行ったものである. CGH 作成の手順を Figure 1 に示す. (1) から(4) までの工程で 3D シーンの作成を行う. 作成したシーンから CGH のパラメータを設定しホログラムの作成を行う. 本研究では、(3) から(6)までの工程を行うソフトの開発を行う.

#### 2.2. シーンの実装

オブジェクトの周囲の環境がわかりにくいと、CGH の計算におけるパラメータの決定などで誤ったパラメータを利用してしまい、計算ミスにつながる. ホログラムを作り直すにはコストがかかるため、オブジェクトの周囲の環境を確認できるようにし、計算ミスを減らすためにシーンの実装を行う. シーンでは、Figure 2 のように点光源生成に使用する視点位置やホロ面の可視化を行う. 視点位置の可視化では、視点位置を計算し、点として表示することで可視を行う.

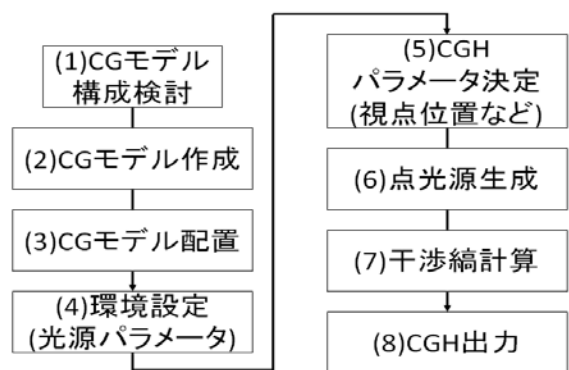


Figure 1. Flowchart of CGH creating

平面型で視点位置の  $x$  座標を計算するときは、下記の計算式を用いて計算している.

$$V_x = X_s + \frac{X_E - X_s}{Div_x - 1} \times i \quad (1)$$

$X_s$ は水平方向の視点移動範囲の最小値,  $X_E$ は水平方向の視点移動範囲の最大値,  $Div_x$ は水平方向の視点数,  $i$ は現在の水平方向の視点番号を表す. 同様に、視点位置の  $y$  座標を計算し、 $z$  座標に設定した距離を入れる.

円筒型、半円筒型では、平面型と同じように  $y$  座標を計算し、 $z$  座標に半径を入れた座標を  $y$  軸を軸にし、回転させることで求めている. 座標の回転角は、円筒型では式(2),

$$angle = \frac{360}{Div_x} [rad] \quad (2)$$

円筒型では式(3)で計算する.

$$angle = \frac{180}{Div_x - 1} [rad] \quad (3)$$

#### 2.3. わき見機能の実装

視点移動範囲の中心でオブジェクトが画面いっぱいに表示されていると、視点が移動した際にオブジェク

トが見切れてしまう。また、このことを回避するために画角を広くとると、オブジェクトの描画が小さくなってしまふ。そのため、描画範囲をオブジェクトに合わせて限定し、見切れてしまうことを回避しながら高解像度のオブジェクト画像を取得できるようにする。このことを以下、わき見と呼ぶ。また、点光源生成はレンダリングした 2 次元画像から計算するため、高解像度のオブジェクト画像を取得することで点光源の生成数を増やすことができる。

OpenGL では、画像取得領域を近端における視野空間の垂直及び水平領域、視野の近端及び遠端を設定し視野空間を限定することでわき見を行う。

平面型では、視点移動範囲における水平方向の両端の  $x$  座標から、オブジェクトの左右端の  $x$  座標を引くことで水平距離を求める。同様に視点移動範囲における垂直方向の両端の  $y$  座標から、オブジェクトの上下端の  $y$  座標を引くことで垂直距離を求める。視点位置がオブジェクトの左右、上下端よりも外にあるときは Figure 3 のようにオブジェクトが見切れてしまうため、三角形の比を用いて、視点位置に近い方の距離を計算し直す。 $x$  座標での計算式を以下に示す。

$$right = \frac{Near_z}{Far_z} \times right0 \quad (right0 > maxX) \quad (4)$$

$Near_z$  は視点の  $z$  座標からオブジェクトの近端までの距離、 $Far_z$  は視点の  $z$  座標からオブジェクトの遠端までの距離、 $maxX$  はオブジェクトの右端を示す。視点の  $x$  座標がオブジェクトの  $x$  座標の最大値より大きい場合は式(4)を、視点の  $x$  座標がオブジェクトの  $x$  座標の最小値より小さい場合は式(4)と同様にして視点からオブジェクトの左端までの距離を計算する。 $y$  座標でも同様の計算を行い、水平、垂直距離から必要な描画範囲を求め、描画範囲が大きい方に合わせてわき見を行う。

円筒型では、水平方向は常に中心を向いているため、垂直方向のみ平面型と同様に計算し、わき見を行う。

### 3. 結果

#### 3.1. シーン機能の実装

点光源生成に使用する視点の移動範囲、視点数から視点位置を計算し、点として表示することで視点位置の可視化を行った。

#### 3.2. わき見機能の実装

Figure 4 のように、わき見を行う前と比べて視野に合わせた大きさでオブジェクトの表示が可能となった。ポリゴン数 1,056 個のオブジェクトで点光源生成を行うと、点光源数は 23,406 個から 144,011 個となり約 6

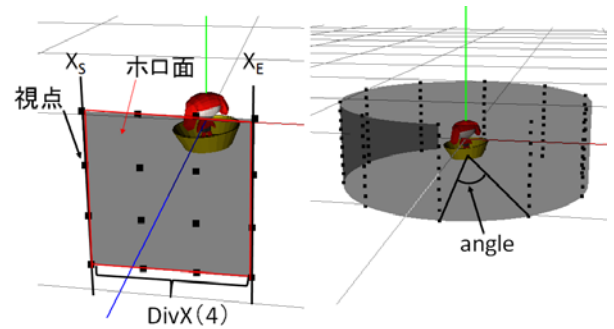


Figure 2. Viewpoint position and the hologram surface display

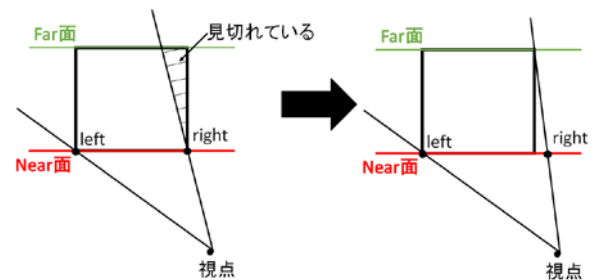


Figure 3. Parameter settings of inattentive

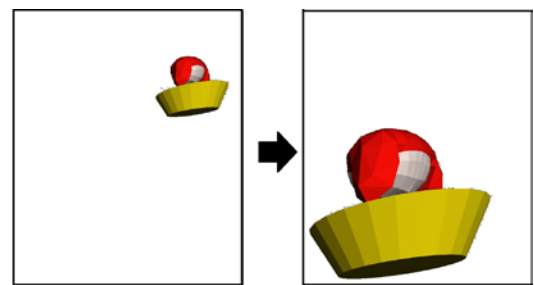


Figure 4. Result of inattentive

倍に増加した。

#### 4. むすび

わき見機能を実装したことにより、点光源数を増やす事が出来た。また、シーン機能の実装により、視点位置やホロ面などオブジェクトの周囲の環境が確認しやすくなった。

今後の課題として、ホログラムの表現の幅を広げるために、環境マッピングを利用したオブジェクトの映り込み表現を実装する。また、現在は、読み込めるファイルの種類が `fbx` と `obj` ファイルの 2 種類であるため、`lwo` など他のオブジェクトファイルを読み込めるようにすることがあげられる。

#### 5. 参考文献

- [1] 藤井知彦, 山口 健, 吉川 浩:「CG を利用した計算機合成ホログラムの隠面処理改善」, 電子情報通信学会信学技報, ITS2006-69, IE2006-254, pp147-152, (2007)
- [2] Davdi Wolff, "OpenGL4.0 シェーディング言語", 株式会社ボーンデジタル(2015)