

# 電子ホログラフィ方式の線画 3D 表示における GPU 実装方式

西辻崇<sup>†</sup> David Blinder<sup>‡</sup> 角江崇<sup>§</sup> 下馬場朋禄<sup>§</sup> 伊藤智義<sup>§</sup>

東京都立大学<sup>†</sup> Vrije Universiteit Brussel, IMEC<sup>‡</sup> 千葉大学<sup>§</sup>

## 1. 序論

生理的に最も自然な 3 次元映像を再生可能な電子ホログラフィ方式において、計算量の大きさは重要課題の 1 つである。ホログラフィの計算処理は 3 次元像からの光伝搬計算で得られる計算機合成ホログラム (CGH: Computer-Generated Hologram) の生成で構成される。3 次元像は、原理的には自発光する点光源で構成するとみなせるため、光伝搬計算は各点光源から発せられる球面波の重畳として計算できる。CGH は重畳した球面波が作る平面上に形成する波面から得られる。そのため、球面波の幾何的対称性を考えれば、CGH 計算は画像描画の問題として解釈でき、コンピュータグラフィクス技術の応用により高速化が可能である。

波面の幾何的対称性が顕著になる線のみで構成される 3 次元像を対象にした CGH の高速計算手法として、著者らは CG-line 法[1]や、解析的な手法[2]を提案している。しかし、[1]はメモリのランダムアクセスが生じるため GPU (Graphic Processing Unit) 等による高速化が容易でなく、[2]は GPU 実装に適しているものの、描画可能な 3 次元像が直線と円弧に限定される。両手法とも、3 次元像の複雑度に従って計算時間が増大するため、描画自由度と GPU 等への高速実装を両立するアルゴリズムが必要である。

本研究では、CG-line 法をベースに、CGH の各画素について独立、かつ、メモリアクセスを伴わない計算手法を開発した。NVIDIA GeForce RTX 2080Ti GPU に実装し、CPU 実装した[1]に比較して最大 46 倍、GPU 実装した[2]に比較して最大 7 倍の高速化に成功した。

## 2. 提案手法

$x$  軸上に直線状に並ぶ無限長の点光源列が作る波面は、

$$L(x, y) = \frac{1}{\zeta} \int_{-\infty}^{\infty} \exp\left(\frac{i\pi}{\lambda\zeta} [(x-u)^2 + y^2]\right) du \quad (1)$$

と定義できる。 $(x, y)$  は波面の座標、 $\zeta$  は奥行再生

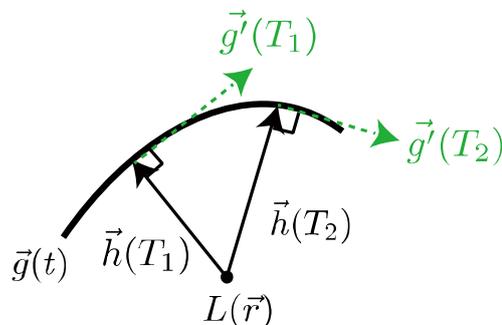


図1 提案手法の概要

距離、 $\lambda$  は光源波長、 $i$  は虚数単位である。式(1)はフレネル積分により、

$$U(y) = \frac{i+1}{\zeta} \sqrt{\frac{\lambda\zeta}{2}} \exp\left(\frac{i\pi y^2}{\lambda\zeta}\right) \quad (2)$$

となり、 $y$  軸方向に分布する 1 次元パターンとなる。CG-Line 法では、式(2)の無限長、かつ、直線の制約に当てはまらない、同じ奥行に存在するすべての曲線に対して式(2)を適用する。曲線上をトレースし、法線方向に事前計算した式(2)の値を積算することで CGH を作成していた。そのため、事前計算値へのメモリアクセスが多く発生し、GPU 等への実装に適していなかった。

提案手法では CGH の各画素における曲線との法線ベクトルを求め、法線上での曲線との距離を計算し、式(2)から波面を求める。概要を図1に示す。奥行 $\zeta$ に位置する曲線を $\vec{g}(t) = (x(t), y(t))$ と定義すると、任意点 $\vec{r} = (\alpha, \beta)$ と曲線上の点を結ぶベクトル $\vec{h}(t)$ は、

$$\vec{h}(t) = (x(t) - \alpha, y(t) - \beta) \quad (3)$$

となる。ただし、 $0 \leq t \leq 1$ である。 $\vec{h}(t)$ が曲線 $\vec{g}(t)$ の法線になる条件は、

$$\vec{h}(t) \cdot \vec{g}'(t) = 0 \quad (4)$$

となる。式(4)を満たす $j$ 番目の $t$ を $T_j$ とすると、法線ベクトルのノルムは、

$$|\vec{h}(T_j)| = \sqrt{\{x(T_j) - \alpha\}^2 + \{y(T_j) - \beta\}^2} \quad (5)$$

となる。したがって、 $\vec{r}$ における波面は、波面計算の線形性と式(2)より、

$$L(\alpha, \beta) = \sum_j U(|\vec{h}(T_j)|) \quad (6)$$

となる。本稿では、一例として、円弧と直線の

Fast calculation of line-drawn computer holography with GPU acceleration

<sup>†</sup> Takashi Nishitsuji, Tokyo Metropolitan University

<sup>‡</sup> David Blinder, Vrije Universiteit Brussel, IMEC

<sup>§</sup> Takashi Kakue, Toyomoshi Shimobaba, Tomoyoshi Ito, Chiba University

事例について、提案手法を具体化する。

円弧の場合、最大で2本の法線が得られる。円の中心と任意点を結ぶ線は、必ず円の法線になることから、元となる円の半径を $R$ 、中心座標を $\vec{c}$ 、各法線ベクトルを $\vec{h}_1, \vec{h}_2$ とすると、

$$|\vec{h}_1| = |\vec{r} - \vec{c}| - R \quad (7)$$

$$|\vec{h}_2| = |\vec{r} - \vec{c}| + R \quad (8)$$

となる。そのため、 $\vec{r}$ における波面は、

$$L(\alpha, \beta) = U(|\vec{h}_1|) + U(|\vec{h}_2|) \quad (9)$$

となる。円弧の存在範囲に入っていない $\vec{h}_1, \vec{h}_2$ を除外すれば、任意の円弧について波面を計算できる。

直線の場合、任意点と直線が最短距離となる線が法線となるため、点と直線の最短距離を求めればよい。直線の方程式を、 $l, m, n$ を定数として、

$$lx + my + n = 0 \quad (10)$$

と定義すれば、任意点 $\vec{r} = (\alpha, \beta)$ との最短距離は、

$$\delta = \frac{|l\alpha + m\beta + n|}{\sqrt{l^2 + m^2}} \quad (11)$$

となるため、 $\vec{r}$ における波面は、

$$L(\alpha, \beta) = U(\delta) \quad (12)$$

となる。直線が線分の場合は、式(12)の計算において領域判定が必要となる。

### 3. 評価・考察

円弧のみで構成される Seigaiha (奥行 8 層) と円と直線で構成される SimpleShape (奥行 12 層) の 2 つの 3D モデルを用いて計算時間、再生画質を評価した。比較として、CPU 実装した CG-Line 法[1]と GPU 実装した解析的手法[2]を評価した。なお、画質評価には線が存在する奥行における真値画像と数値再生像の PSNR (Peak Signal to Noise Ratio) の平均値を求めた。評価条件は、波長: 532 nm, CGH 解像度: 1,920 × 1,080 pixels, CGH 画素間隔: 8 μm である。また、計算機環境として、GPU: NVIDIA GeForce RTX 2080Ti, CPU: AMD Ryzen 9 3950X 3.5 GHz, Memory: DDR4-2132 64GB, OS: Microsoft Windows 10 Professional 64 bit, Compiler: Microsoft Visual C++ 2019 compiler を用いた。

計算時間の評価結果を表 1 に、画質の評価結果を表 2 に示す。また、提案手法で作成した両モデルの CGH に対する数値再生像の例を図 2 に示す。提案手法は、比較手法より高速であり、画質も改善、または同程度の劣化に抑えられることが分かった。CG-Line 法に対する画質改善効果の要因は、CG-Line 法が曲線上の法線方向に式(2)を適用するため、曲線の外側において生じてしまう法線間のギャップを、提案手法は埋められてい

ることに起因すると考えられる。

表 1 計算時間の比較

[ms]	Proposed	[1](CPU)	[2](GPU)
Seigaiha	3.91	50.0	28.4
SimpleShape	0.802	37.0	2.98

表 2 画質の比較

[dB]	Proposed	[1](CPU)	[2](GPU)
Seigaiha	21.0	19.3	20.0
SimpleShape	21.8	21.2	22.5

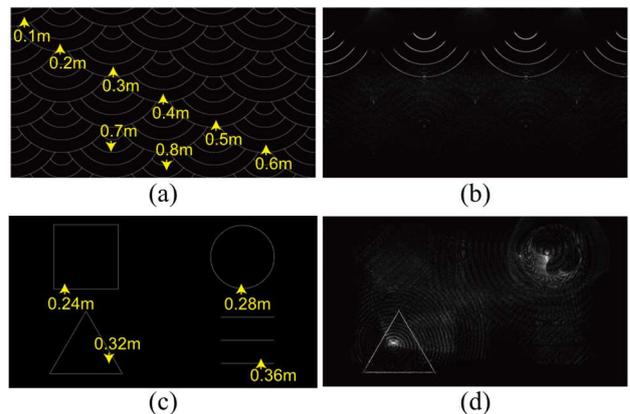


図 2 3D モデルと数値再生像の例

(a) Seigaiha モデルの真値画像と奥行距離、(b) 0.3 m における数値再生像、(c) SimpleShape モデルの真値画像 (一部) と奥行距離、(d) 0.32 m における数値再生像

### 4. 結論

線で構成される 3 次元像を再生可能な CGH 作成手法である CG-Line 法を効率的に GPU 動作する方式を開発し、画質、計算速度の点で有効な結果を得られた。提案手法は任意曲線に拡張可能であり、実用上の優位性は高い。

### 5. 参考文献

- [1] T. Nishitsuji, T. Shimobaba, T. Kakue, and T. Ito, "Fast calculation of computer-generated hologram of line-drawn objects without FFT," Opt. Express, 28(11), 15907–15924 (2020).
- [2] D. Blinder, T. Nishitsuji, T. Kakue, T. Shimobaba, T. Ito, and P. Schelkens, "Analytic computation of line-drawn objects in computer generated holography," Opt. Express 28(21), 31226–31240 (2020).

### 謝辞

本研究は JSPS 科研費 20K19810 の助成を受けたものです