

J-15 実際のカメラから推定した画像ベースの高速被写界深度レンダリング

Fast Image-Based Depth of Field Rendering Estimated From A Real Camera

佐藤 倫也[†] 土橋 宜典[†] 山本 強[†]
Tomoya Sato Yoshinori Dobashi Tsuyoshi Yamamoto

1 はじめに

被写界深度 (Depth of Field: 以後 DOF) は映像技術において重要な要素である。あるオブジェクトに焦点を合わせると、焦点からはずれている領域のオブジェクトはピン트가はずれてぼけが現れる。一般に、コンピュータグラフィックスでは、理想的なピンホールカメラのモデルを用いてレンダリングされている。そのため、DOF 効果は考慮されず、得られる画像はすべてのピクセルがはっきりしており、不自然な画像となる。DOF 効果を考慮することで、より写実的で自然な画像を生成することができる。また、焦点が合っているオブジェクトや領域に、観測者の注意を引くことができ、オブジェクトの重要性を印象づけることができる。DOF をレンダリングする手法は、これまでにいくつか開発されているものの、高精度なものは処理時間がかかり、高速なものは精度が十分ではないという問題がある。

本研究の目的は、DOF 効果を高精度かつ高速にレンダリングすることである。本論文では、実際のカメラで撮影した画像からぼけ関数を推定し、これを利用することで、高速に DOF をレンダリングする画像ベースの手法を提案する。

2 従来研究

DOF をレンダリングするためには、一度画像をレンダリングし、その後で画像処理により DOF 効果を加える方法がある。Potmesil らがこの手法により DOF をレンダリングするアルゴリズムを最初に開発した [1]。しかし、このアルゴリズムはソフトウェアで実装されており、非常に処理時間がかかる。そのため、高速化手法がいくつか提案されている。Mulder らは人間の視覚システムを考慮した手法を提案している [2]。この手法では、グラフィックスハードウェアを用いることで高速な処理が行えるものの、精度の面で十分とはいえない。

DOF はマルチパスレンダリングの手法を用いることでもレンダリングすることができる。これは、視点をわずかにずらしながら複数回シーンをレンダリングし、それらを合成することで DOF 効果を得る。高精度に DOF をレンダリングする手法として代表的な分散レイトレーシングでも同様の手法が用いられている [3]。しかしこれらの手法では、高精度な結果が得られるものの、単一の画像を作成するのにシーンを複数回レンダリングしなければならないため、処理時間がかかるという問題がある。

以上のアプローチとは別に、Kolb らが開発したコンピュータグラフィックスでリアルなカメラのモデルをシミュレーションする手法がある [4]。この手法は DOF に特化したものではないが、実際のカメラをシミュレーションするものであるため、結果的に、上記のものより

も高精度な DOF 効果を生成することが可能である。しかし、レイトレーシング法を用いたレンダリング手法であるため、非常に時間のかかる処理となっている。本研究では、この手法を高速化することで、高精度かつ高速な DOF レンダリングを行うのが狙いである。

3 提案手法

本研究では Kolb らの手法 [4] を高速化することで、高精度かつ高速な DOF レンダリングを行うのが、最終的な目標である。高速化は、同一のシーンで焦点の合っている画像と合っていない画像を作成して、DOF によるぼけの程度を表す関数 (ぼけ関数と呼ぶ) を推定し、ぼけ関数を用いてレンダリングを行う。ぼけ関数を推定するための画像は、Kolb らのモデル [4] を用いて、カメラパラメータを様々に変更することで作成する。しかし、本論文では、提案手法により DOF を高速にレンダリングできるかを検証するための実験として、実際のカメラで撮影した画像を用いることにする。すなわち、撮影した画像からぼけ関数を推定する。また、本来であれば、このぼけ関数は深度 (デプス) 値の関数でなければならないが、本論文では、実験段階であるので、デプス値は固定としている。以下、ぼけ関数の推定とレンダリングアルゴリズムの詳細について述べる。

3.1 ぼけ関数の推定

まず、黒い紙に小さな白い点を打ったものを用意し、カメラでその白い点に焦点を合わせた画像を撮影する。次に焦点を変えずに、紙とカメラの距離を離して同様に画像を撮影する。このとき、白い点はぼける。図 1 に画像から白い点の部分を抜き出したものを示す。図 1 左は白い点に焦点の合っている画像で、焦点距離は 20cm である。図 1 右は焦点距離から白い点を 30cm 遠ざけた (すなわちカメラからは 50cm 離れた) 画像である。



図 1 撮影した画像の一部

これら画像から白い点 (円) のピクセルの輝度値を調べ、ぼけの強さの分布を求める。円の座標を中心からの距離 r と角度 θ で表し、図 1 左の輝度値を $I(r, \theta)$ 、右の輝度値を $I_b(r, \theta)$ とおく。ぼけの強さは $I_b(r, \theta)/I(r, \theta)$ で計算できる。中心から円周までのぼけの強さを θ 方向に 10 度刻みでサンプリングした結果を図 2 に示す。図の点群が測定された値である。データを正規分布と仮定し、これをガウス関数でフィッティングする。ガウス関数を

$$f(r) = N \exp\left(-\frac{(r - r_c)^2}{2\sigma^2}\right) \quad (1)$$

[†]北海道大学大学院工学研究科

Graduate School of Engineering, Hokkaido University

とおき、パラメータの推定を行う。ここで、 N 、 r_c 、 σ^2 はピークの強さ、位置、幅を表す。パラメータの推定には最小2乗法を用いた。推定されたパラメータを表1に示す。また、図2の曲線が推定された関数 $f(r)$ である。

表1 推定されたガウス関数のパラメータ

N	5.8116×10^{-1}
r_c	4.2853×10^{-4}
σ^2	5.5114×10^{-6}

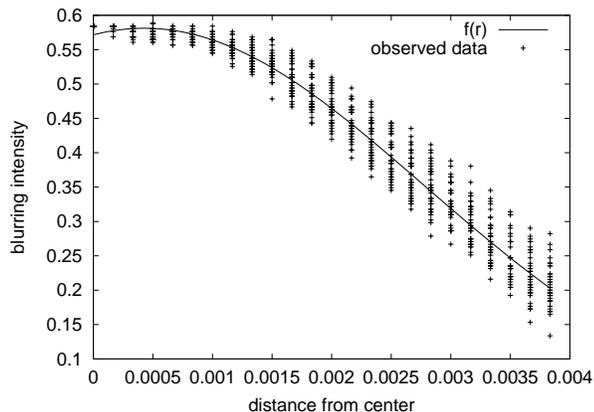


図2 ぼけの分布と推定した曲線

3.2 DOFのレンダリング法

実際に画像にDOF効果をレンダリングするには、テクスチャマッピングの機能を用いる。まず、前節で推定したぼけ関数から、 r に応じたぼけの値をもつテーブルを作成する。そして、このテーブルからテクスチャを作成する。作成したテクスチャを図3に示す。このテクスチャをぼけテクスチャと呼ぶ。



図3 ぼけテクスチャ

このぼけテクスチャを用いてレンダリングを行う。レンダリングアルゴリズムは以下のとおりである。
[アルゴリズム]

1. 画像をレンダリングする。
2. すべてのピクセルについて以下を繰り返す。
 - (a) デプスバッファからデプス値を読み込む。
 - (b) デプス値に対応するぼけテクスチャを選択する。
 - (c) 注目しているピクセルを中心とするピルボード (四角形の板) を配置し、ぼけテクスチャをマッピングする。この際、テクスチャは上書きされるのではなく、元の輝度値と掛け合わされる。

以上のアルゴリズムにより、DOFをレンダリングすることができる。ただし、本論文では実験段階であり、先に述べたようにデプス値は固定であるため、ステップ2(a)、2(b)は省略される。テクスチャマッピングにはグラフィックスハードウェアを利用できるため、高速に処理を行うことができる。

4 適用例

提案手法を用いてDOF効果をレンダリングした。パラメータは前節のぼけ関数推定時のものと同様になるように設定してある。すなわち、焦点距離は20cmであり、オブジェクト (四角形) は50cmの距離に置かれている。提案手法によりDOFをレンダリングした結果を図4に示す。図からわかるように四角形がぼけており、提案手法により、DOF効果をレンダリングできることを確認できた。また、画像生成の処理を1000回繰り返し、計算時間を計測したところ、約26.4秒かかった。すなわち、1枚の画像を生成するのにかかる時間は約0.026秒であり、十分高速に処理できることを確認できた。



図4 提案手法を適用した画像

5 まとめ

本論文では、実際のカメラで撮影した画像からぼけ関数を推定し、ぼけ関数を利用することで、高速にDOF効果をレンダリングできる画像ベースの手法を提案した。提案手法を用いることで、効果的にDOF効果をレンダリングできることが確認できた。しかし、提案手法の精度については評価がなされていないため、今後の課題としたい。また、本論文ではデプス値が固定であったため、デプス値に応じたぼけ関数の推定も課題として挙げられる。その後はKolbらの手法 [4] を実装し、カメラパラメータを変えた様々な画像を作成し、ぼけ関数の推定を行う予定である。

参考文献

- [1] M. Potmesil, I. Chakravarty, "A lens and aperture camera model for synthetic image generation", *Computer Graphics (Proc. of SIGGRAPH'81)*, pp. 297-305(1981).
- [2] J. D. Mulder, R. V. Liere, "Fast Perception-Based Depth of Field Rendering", *Proc. of ACM VRST2000*, 2000.
- [3] R. Cook, T. Porter, L. Carpenter, "Distributed Ray Tracing", *Computer Graphics (Proc. of SIGGRAPH'84)*, vol. 18, pp. 137-145(1984).
- [4] C. Kolb, D. Mitchell, P. Hanrahan, "A Realistic Camera Model for Computer Graphics", *Proc. of SIGGRAPH'95*, pp. 317-324(1995).