

イメージクラスターからの任意視点画像生成

Rendering Arbitrary Viewpoint Images from Image Cluster

巴図 斯仁
Siren Batu

土橋 宜典
Yoshinori Dobashi

山本 強
Tsuyoshi Yamamoto

北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

1 まえがき

近年、複数の実写画像から任意視点画像をレンダリングするイメージベースドレンダリング (Image-Based Rendering, IBR) 手法が注目を集め、数多くの研究が行われている [1][2][3][4][5]. しかし、これらの手法は、中間形式として3次元以上の空間を稠密に埋めるための大量の原画像データを必要とし、さらに、各画像を決められたサンプリング間隔で撮影する必要がある. そういった要求を満たすために、大量のカメラから構成されるカメラアレイ、あるいは機械的に厳密にコントロールできる特殊な収録システムが必要であった. 本研究では、予め記録された大量なカメラパラメータ付きイメージクラスターから任意視点映像をレンダリングする新しい IBR 手法を提案する. ここで、イメージクラスターとは、一定の空間内で撮影された一定の密度分布を持つ大量な画像の集合である. 本手法は入力画像に加え、シーンの深度情報を用いることにより従来の Plenoptic Function の縮退、近似型のアプローチと比べて、少ない画像セットから任意視点画像の生成を可能にするものである. また、入力画像が決められたサンプリング点で撮影されたものである必要がないため、撮影システムが柔軟であり、実現が容易である.

本研究では、最終的に自然画像 (実写画像) に適用することを目的としているが、本原稿では CG により生成された画像を用いて実装実験を行うことで、基本的なアルゴリズムを示し、その有効性と問題点を考察する.

2 提案手法

本手法ではまず、前処理として入力画像から簡易な三次元形状モデルを作成する. 新しい視点では、その形状モデルをレンダリングすることにより、シーンの深度情報を取得し、取得したシーンの深度情報および入力画像を用いて新しい視点での画像を生成する. 新しい視点での画像の生成はピクセル単位で行うが、各ピクセルの色を決定するプロセスは既知のピクセル間の補間問題、または、リサンプリング問題ではなく、入力画像クラスターの最適な画像から最適なピクセルを選択する最適化問題として定式化する. 本手法は、二つのステップで構成されている. まず、一つ目のステップで入力画像クラスターから最適な画像フレームを決定し、次のステップで最適なピクセルを決定する. 以下その詳細について述べる.

2.1 画像フレームの選択

まず、ピンホールカメラモデルに基づき、図 1-(a) のように、三次元空間内の任意の直線 l の画像平面上に投影される像を考える. x を直線上の任意の一点とし、 y

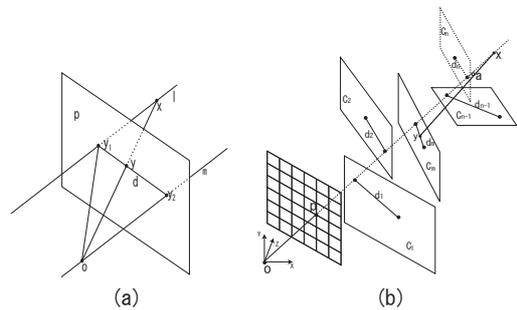


図 1 フレームの選択

をその投影点とする. ただし、カメラの投影面より後の点は投影点を持たない. すると、すべての点 y は線分 y_1y_2 上に投影される. ここで、点 o はカメラの中心、平面 P はカメラの投影面、直線 m はカメラの中心を通る直線 l と平行な直線であり、点 y_1, y_2 はそれぞれ直線 l, m のカメラ投影面 P との交点である. 直線 l, m は並行であるため、その交点は直線 l, m 上の無限遠点であると見なされる. したがって点 y_2 は直線 l 上の無限遠点の投影点でもある. 明らかに、直線 l がカメラの中心 o を通る時、 $d = 0$ となり、直線 l がカメラ投影面 P と平行である時、 $d = +\infty$ となる. 提案手法では点 y_1, y_2 の距離

$$d = \|y_1y_2\| \quad (1)$$

を最適なフレームを選択する際の目的関数として用いる. つまり、図 1-(b) のように点 o を新しい視点とし、点 p を画面上の一つのピクセルとする. 各カメラフレーム C_1, C_2, \dots, C_n における式 (1) による直線 op の投影線分の長さをそれぞれ d_1, d_2, \dots, d_n とした時、

$$d_m = \min(d_1, d_2, \dots, d_n). \quad (2)$$

を満たすフレーム C_m を最適なフレームとして用いる. なぜなら、式 (2) を満たすフレームを C_m とした時、図 1-(b) の角度 a も最小になる. ここでは、点 x は直線 op 上の任意の点、点 y は点 x のフレーム C_m 上の投影点である.

2.2 ピクセルの選択

図 2 のように、新しい視点 o において、一つのピクセル p を考える. 前述のように、点 o, p を通る直線の投影

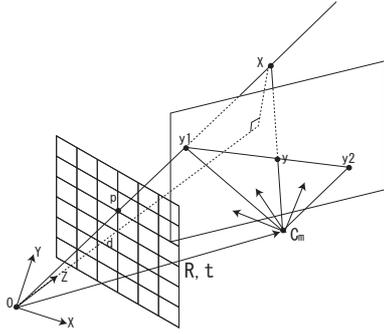


図2 ピクセルの選択

線分の長さを最小とするフレームを C_m とする。また、点 p に対応する三次元空間の点を x とすると、点 x のフレーム C_m における投影点 y は線分 y_1y_2 上にあることが保障され、次の式で計算される。

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{p}. \quad (3)$$

$$\mathbf{H} = \mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{d}, \mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

ここで、ベクトル \mathbf{p}, \mathbf{y} はそれぞれ点 p, y の同次座標系表現による三次元ベクトル、 \mathbf{R}, \mathbf{t} はそれぞれ新しい視点 o を基準にした、カメラフレーム C_m の回転行列と平行移動ベクトルである（ここでは、縦ベクトル表示を用いている）。 d は点 x の深度であり、 T はベクトル転置を表す。

3 実装実験

本実装で用いたすべてのデータは DirectX を用いて作成した。ターゲットシーンの構築に当たっては、一般的な屋外シーンを想定し、建築物、車、木などを含む複雑なシーンを生じた。また、カメラの位置は X, Y, Z それぞれ 200, 200, 500 単位の直方体の内部でランダムに生成し、カメラの向きはある一定の視線ベクトルをランダムに生成されたベクトルを軸に、区間 $[-\pi/4, \pi/4]$ 上でランダムに生成された角度回転させることで生成した。一方、深度マップは新しい視点で CG シーンをレンダリングし、深度バッファを直接読み取ることで取得しているが、実際の応用においては、通常、ポリゴンモデルがないため、前処理として、三次元形状復元手法などを用いて、入力画像から大まかなポリゴンモデルを作成する必要がある。本実装では合計 8192 枚の画像を用いた。

まず、もっとも理想的な条件のもとでの結果を示す。ここで、理想的な条件とは、画像の密度は十分高く、カメラのパラメータデータおよびシーンの深度データはともに正確であり、誤差がないことを指す。比較のため、図 3 に CG を用いてレンダリングした正確な画像を示した。さらに、図 4 に入力として用いた深度マップを示した。図 5 と図 6 は提案手法を用いて生成した結果画像である。図 5 は元画像 8192 枚すべてを用いて生成したものであり、図 6 は元画像から間引いて得られた 2048 枚の画像のみを用いて生成したものである。いずれも元画



図3 参照画像



図4 深度マップ



図 5 8192 枚画像による結果 図 6 2048 枚画像による結果像とほぼ同等の画質が実現されている。以上のように、理想的な条件のもとでは、比較的少ない入力画像であっても、よい画質が実現可能である。しかし、実際の応用においては、カメラパラメータおよびシーン深度データが一定の誤差を持つことが想定されるため、本実験では、入力データの誤差が画質に与える影響についても検証を行った。図 7 は深度データに 25% の相対誤差を付加した時の結果であり、図 8 はカメラ回転に 0.5° の誤差、平行移動ベクトルに 1.5% の相対誤差をそれぞれ付加した時の結果である。図に示されたように、提案手法はシーンの深度情報を必要とするものの、深度データの誤差に対してはロバストである。しかし、カメラパラメータの誤差により、画質が大きく劣化する。したがって、実際の応用においては、正確なカメラパラメータを取得することが主な課題となる。

図7 深度相対誤差：25% 図8 誤差：R 0.5° , T 1.5%

参考文献

- [1] S. E. Chen and L. Williams, "View Interpolation for Image Synthesis," Computer Graphics (SIGGRAPH' 93 Proceedings), pp. 279–288, Jul. 1993.
- [2] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System," Computer Graphics (SIGGRAPH' 95), pp. 39–46, Aug. 1995.
- [3] M. Levoy and P. Hanrahan, "Light field rendering," Computer Graphics (SIGGRAPH' 96), pp. 31–42, Aug. 1996.
- [4] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," Computer Graphics (SIGGRAPH' 96), pp. 43–54, Aug. 1996.
- [5] H. Y. Shum and L. W. He, "Rendering with Concentric Mosaics," Computer Graphics (SIGGRAPH' 99), pp. 299–306, Aug. 1999.