自由視点画像生成のための フォーカルスタックによる多層シーン表現

石川 玲奈[*1]　森 尚平[*2]　上田 栞[*1]　デニス・カルコーフェン[*2]　斎藤 英雄[*1]

Abstract – 多層画像 (MPI) によるシーンの 3 次元構造表現に基づく自由視点画像生成は，描画が高速であり，さらに MPI を機械学習により獲得することによって高精度化が可能な方法として，近年盛んに研究が行われている．本稿では，自由視点画像生成の入力である多視点画像から多層画像を生成する際に，フォーカルスタックへ変換してから多層画像を生成するフレームワークを提案する．本手法は，フォーカルスタックを経由することにより，任意の位置に MPI を生成できかつ局所的なノイズの影響を緩和するといった有利な効果が期待できる．本稿では，フォーカルスタックを用いる上で必要とされる要件を明確にし，それにより得られる利点について議論するために，計算資源と精度について性能評価を行った結果を示す．

Keywords : **Multi-layer scene representation, multi-plane images, focal stack**

1　はじめに

多視点画像からの自由視点画像生成は Plenoptic Image to Image ビューモーフィング [15] を筆頭に，明示的なプロキシを求める手法 [3, 9] や深層学習でプロキシを代替する手法 [17] が提案されてきた．とりわけ，Virtual Reality (VR) やAugmented Reality (AR) においては，首尾一貫した高品質高性能な描画性能が要求されており [18]，その実現が目下の課題となっている．具体的解決案の一つとして，多層シーン表現 [10, 7] が注目されている．

多層シーン表現は，奥行方向に平行に並ぶ画像群 (Multi-Plane Image (MPI)) [16, 23]，あるいは同軸球面群 [1, 2] から構成され，各画像は RGBα 値を保有する．RGB 値は色を，α 値はその点の存在の曖昧さを表現している [12]．この構造により，GPU を用いたラスタリゼーションによる高効率な描画が可能になる．多層シーン表現は，ライトフィールドをより疎な奥行表現に簡素化したものといえよう [10]．

深層学習により，多視点画像からの RGB+α 値の推定が微分可能な描画を含む目的関数として定義できるようになった [7]．高精度かつ頑健な多層画像推定が可能となった一方，既存手法では入力上限枚数が，例えば $N = 5$ と限定されている [10]．これは，Convolutional Neural Network (CNN) が N 個のPlane Sweep Volume (PSV) を入力とする構造になっていることで，入力画像枚数の増加に応じて，元の画像枚数に PSV のレイヤー数 D 倍のメモリが必要となり，即座にその上限に達することに起因する．

そこで我々は，多視点画像を一旦フォーカルスタック（焦点を変えながら同一視点で撮影した画像群）としてまとめ，そこから MPI を生成する手法を提案する．この手法は，ネットワークに対するメモリ量が一定になるだけでなく，潜在的に局所的な画像ノイズに対する頑健性の向上が期待される．さらには，フォーカルスタックの形をとることで，上限なく任意の枚数の画像を入力できる．これは言い換えれば，MPI の生成に従来手法より多くの画像の情報を取り入れることができることを意味する．

本稿では，以下の 3 点について論じる．

- 多視点画像群から生成したフォーカルスタックをネットワークへの入力として MPI を推定するフレームワークの提案
- フォーカルスタックを生成する上で満たされるべき理論的境界
- フォーカルスタックを用いることで，少数の入力画像を用いる既存手法よりも，必要となる入力データ量と位置姿勢推定や画像の局所的ノイズに対する頑健性において優れていることの検証

2　自由視点画像生成フレームワーク

提案手法は，位置姿勢が既知の画像群からのフォーカルスタック合成（2.1節），CNN による MPI の生成（2.2節），自由視点画像の描画（2.3節）の 3 ステップで構成される．

2.1　フォーカルスタックの生成

本ネットワークは，画像群から生成される固定長 D のフォーカルスタックを入力として受け付ける．ため，事前に設定された 3D プロキシに従って撮影され

[*1]慶應義塾大学
[*2]グラーツ工科大学

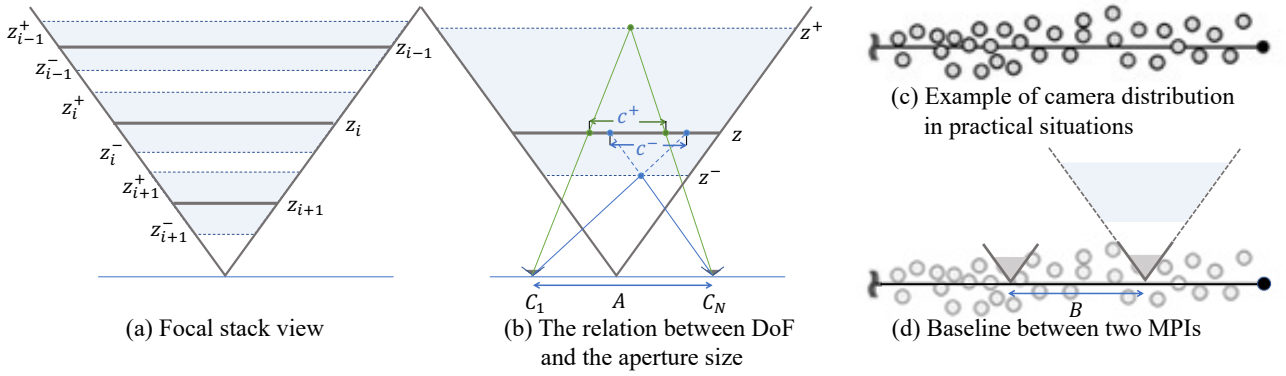


図 1　合成開口サイズと基線長
Fig. 1　The illustration of the aperture size and the baseline

た連続フレーム，すなわち動画像を用いることができる．故に，ユーザーが画像を一枚ずつ厳密な位置姿勢で撮影する（例えば [10]）代わりに，記録された大量の画像群から必要な範囲の画像を自動的に選択できる．尚，入力画像群は合成開口カメラ [19] として捉えられ，その開口やカメラ間の基線長は Plenoptic Sampling Theory を満たす必要があることに注意されたい [5]（3章参照）．また，フォーカルスタックの合成を通して，局所的なノイズ（画像ノイズやトラッキングノイズ）の影響を軽減できることが期待される．

提案手法では，開口サイズ A 内で撮影された，N 枚の校正済み画像群 $\mathcal{I}_k = \{I_k,\ [\mathbf{R}_k|\mathbf{t}_k], \mathbf{K}_k\} \in \mathbf{I}^{\mathrm{MV}}$ からフォーカルスタックを生成する．ここで，I_k は k 枚目の画像を，$\mathbf{R}_k$ and $\mathbf{t}_k$ は相対的な $\mathbb{SO}(3)$ の回転行列と三次元移動ベクトル表し，$\mathbf{K}_k$ は 3×3 カメラ行列を意味する．$z_i\ (i \in \{0, 1, ..., D-1\},\ z_i > z_{i+1}) \in \mathbf{z}$ の奥行き位置に配置される D 枚の画像から構成されるフォーカルスタックを生成するためには，各入力画像を，目的の合成開口カメラ画像 $\mathcal{I}_{\mathrm{tgt}}$ に並行な画像平面 z_i に投影した結果を，各ピクセルについて総和したうえで正規化する必要がある．合成開口カメラ位置 $\mathbf{K}_{\mathrm{tgt}}$ 及びその前方に向かう法線ベクトル $\mathbf{K}_{\mathrm{tgt}}$ を与えられた時，その画像から画像への投影をするためのホモグラフィ行列は，

$$\mathbf{H}_{k,\mathrm{tgt}} = \mathbf{K}_{\mathrm{tgt}} \frac{\mathbf{R}_{k,\mathrm{tgt}} - \mathbf{t}_{k,\mathrm{tgt}}\mathbf{n}_{\mathrm{tgt}}^{\mathrm{T}}}{z_i} \mathbf{K}_k^{-1} \tag{1}$$

によって計算される．従って，$\tilde{\mathbf{u}} = [u, v, 1]^{\mathrm{T}}$ における正規化画素値 $c_i(\tilde{\mathbf{u}})$ は

$$c_{\mathrm{tgt}}(\tilde{\mathbf{u}}) = \frac{\sum_{k=0}^{N-1} I_k(\mathbf{H}_{k,\mathrm{tgt}}^{-1}\tilde{\mathbf{u}})}{\sum_{k=0}^{N-1} \mathbb{1}_k(\mathbf{H}_{k,\mathrm{tgt}}^{-1}\tilde{\mathbf{u}})}, \tag{2}$$

となる．ここで，$\mathbb{1}$ は 0 または 1 の値をとる，以下に表す指示関数である．

$$\mathbb{1}_k(\mathbf{u}) = \begin{cases} 1 & \text{if } \mathbf{u} \text{ lays within } k_{\mathrm{th}} \text{ camera FoV} \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

2.2　MPI の生成

本提案手法では，生成した D 層から成るフォーカルスタックを，U-Net [14] を基にした CNN に入力し，同じく D 層から成る MPI を推定する．

ネットワーク構造．元来の U-Net は，そのアップサンプリング構造により，チェッカーパターンのアーティファクトを生成することで知られているため，双線形補間アップサンプリングに置き換えた．各層が奥行値の逆数において等間隔に配置された $W \times H \times 4 \times D$ 画素の MPI を想定し，ネットワーク $\mathcal{N}$ には各層の各画素の RGBα 値を出力させる．

目的関数．我々は微分可能描画関数を実装し，描画された画像に関してネットワークを最適化する．新しい視点画像を生成するために， 以下のような “over” α 合成を用いる [13]:

$$\mathcal{R}^{\mathrm{Over}}(\mathbf{I}_{\mathrm{MPI}}) := \sum_{i=0}^{D-1} \left(C_i^{\mathrm{MPI}} \alpha_i^{\mathrm{MPI}} \prod_{j=i+1}^{D-1} (\mathbf{1} - \alpha_j^{\mathrm{MPI}}) \right) \tag{4}$$

ここで，C_i^{MPI} と α_i^{MPI} はそれぞれ，MPI の i 層目の RGB 値と α 値を示す．N 視点の画像の描画を $\mathcal{R}_N^{\mathrm{Over}}$ としたとき，我々が最適化する損失関数は，ネットワークの重み $\mathbb{W}$ を用いて，以下で定義される．

$$\arg\min_{\mathbb{W}} \mathcal{L}(\mathcal{R}^{\mathrm{FS}}(\mathcal{R}_N^{\mathrm{Over}}(\mathcal{N}(\mathbf{I}^{\mathrm{FS}})), \mathbf{z}),\ \mathbf{I}^{\mathrm{FS}}) \tag{5}$$

2.3　自由視点画像生成

我々はLocal Light Field Fusion (LLFF) で提案されている，n 近傍の MPI を合成する方法を想定する [10]．提案手法では，撮影平面上の有効範囲内であれば任意の位置にフォーカルスタックを生成できるため，

グリッド状に均一に MPI を配置することが可能である．よって，滑らかな視点の遷移が期待される [8].

3 理論的境界

提案するフレームワークにおいて，ネットワークはフォーカルスタックから MPI の RGBα 値を求める必要があるため，入力フォーカルスタックの最低 1 層において，空間内の点に焦点が当たっていることを保証せねばならない．これは，ネットワークがぼけ画像から先鋭な画像を推定せずに済むためである．本章では，与えられたパラメータから，合成開口カメラの開口サイズ，すなわち一つのフォーカルスタックを生成するため画像の撮影有効範囲の大きさと，生成すべき MPI の間隔（基線長）の理論値を算出する．

3.1 最大合成開口サイズ

D 層からなるフォーカルスタックにおいて，z_i ($i \in \{0, 1, ..., D-1\}$, $z_i > z_{i+1}$) 番目のレイヤーの被写界深度を $[z_i^+, z_i^-]$ とする．一般性を損なわない平面世界 (図 1) では，空間内の全ての点が被写界深度内に含まれる必要があるため，隣り合う被写界深度が隙間なく配置されるように

$$z_{i+1}^+ \geq z_i^- \tag{6}$$

を満たす必要がある．

被写界深度の内，最も遠い点と近い点をそれぞれ c_i^+ と c_i^- とするとき，図 1 (b) 内に観察される相似の関係から，以下が成り立つ．

$$z_i^- = \frac{Az_i}{A + c_i^-} \text{ and } z_i^+ = \frac{Az_i}{A - c_i^+}. \tag{7}$$

式 7を式 6に代入すると，開口サイズの上限

$$A \leq \frac{z_i c_{i+1}^+ + z_{i+1} c_i^-}{z_i - z_{i+1}} \tag{8}$$

を得る．この式とカメラパラメータの関係を明確化するため，c^- と c^+ を視野 θ_{fov} とピクセル単位での画像サイズ W_{px} によって表現すると，

$$c^- = c^+ = \frac{2C_{\text{px}} \tan(\theta_{\text{fov}}/2)}{W_{\text{px}}} \tag{9}$$

となる．ここで，C_{px} は許容される最大視差を示す．式 9を式 8に代入することで，以下の開口サイズの上限値が取得される．

$$A \leq \frac{4C_{\text{px}} z_i z_{i+1} \tan(\theta_{\text{fov}}/2)}{W_{\text{px}}(z_i - z_{i+1})}. \tag{10}$$

ここで，各層の間隔の逆数

$$\Delta z^{-1} = \frac{1}{D-1}\left(\frac{1}{z_{D-1}} - \frac{1}{z_0}\right) \Leftrightarrow \frac{z_i - z_{i+1}}{z_i z_{i+1}} \tag{11}$$

表 1　3章で使用される記号
Table 1　Reference for symbols in 3.

Symbol	Unit	Definition
W_{px}	pixels	Camera image width
θ_{fov}	radian	Camera FoV
A	meter	Synthetic aperture size
D	none	Number of layers
C_{px}	pixels	Maximum circle of confusion
B	meter	Baseline between two MPIs
N	none	Number of MPIs
z_i	meter	i_{th} layer or focal distance
$z^{-/+}$	meter	Close/far depth of field (DoF) range
$c^{-/+}$	meter	Circle of confusion for $z^{-/+}$

を用いると，式 10 は

$$A \leq \frac{4C_{\text{px}} \tan(\theta_{\text{fov}}/2)}{W_{\text{px}} \Delta z^{-1}} \tag{12}$$

と書き換えることができる．さらに，多視点カメラの四角錐同士が z_{D-1}^- において必ず重なりを持つことを保証する必要があるので，以下の条件も同時に満たす必要がある．

$$A \leq 2z_{D-1} tan(\theta_{\text{fov}}/2). \tag{13}$$

以上より，合成開口カメラの開口サイズの最大値は式 12と式 13から，

$$A \leq \min\left(\frac{4C_{\text{px}} \tan(\theta_{\text{fov}}/2)}{W_{\text{px}} \Delta z^{-1}},\ 2z_{D-1} tan(\theta_{\text{fov}}/2)\right) \tag{14}$$

と算出される．具体例を挙げれば，$D = 32$, $\theta_{fov} = \pi/3$, $W_{px} = 256$, $z_0 = 9.0$ m, $z_{D-1} = 1.0$ m，の時，開口サイズの最大値 A は 0.31 m となる．

3.2 MPI 間の最大基線長

式 14は 1 つの MPI の品質を保証するものであったが，保証される範囲もまた限定的である．1 つのシーンを隙間なく描画するために，複数の MPI 間を遷移させる．MPI 間で描画結果のエイリアシングを防ぐには，サンプリング理論を満たすに十分な数の MPI を生成する必要がある．これには，文献 [10] で算出される最大基線長 B を使用する．

$$B \leq \min\left(\frac{2C_{\text{px}} \tan(\theta_{\text{fov}}/2)}{w_{px} \Delta z^{-1}},\ z_{D-1} \tan(\theta/2)\right). \tag{15}$$

ここで，B は式 14の開口サイズ A の丁度半分の値になっていることに注意されたい．これは，無制限の画像群からフォーカルスタックを生成においては遮蔽部分を考慮する必要がないからである [5].

4 性能評価

我々の設計したネットワークとフォーカルスタックから MPI が生成できることを確認する．また，その強みと考えられるメモリ消費量と入力画像に含まれるノイズの影響について議論する．

4.1　データセット

文献 [21] と同様に Blender[6] を用い合成画像データセットを作成した．実画像を用いないことで，3章にて議論した理論的境界を満たすカメラ配置や位置姿勢及び画像に対するノイズを操作できるようにした．水平方向に 56.2475° の画角内に 256×256 画素を持つ仮想カメラを用意し，2.1節にて算出される合成開口サイズ内で，1 シーン当たり $11 \times 11 = 121$ 枚の画像を 120 シーン生成した．各シーンには *Thigi10K*[22] の中から，画角が埋まるよう無作為に選択した物体を $1.0m$ から $10.0m$ の深度範囲に配置した．

本稿では，Albumentations library[4] を用いて，画像撮影時に発生する ISO ノイズ，モーションブラー，カメラ位置ノイズの 3 種類を付与した．ISO ノイズ，モーションブラーの強度 I_k^{iso}, $I_k^{blur} \in [0, 1]$ とし，ガウス関数でカメラ位置ノイズの大きさを決定した後，カメラ位置ノイズを同スケールで生成する．カメラ位置ノイズは，ISO ノイズやモーションブラーによって引き起こされると考えられるので，正しい座標を原点とした極座標系の半径 I_k^r を ISO ノイズやモーションブラーの大きさによって定義する．

$$I_k^r = max(I_k^{iso}, I_k^{blur}) \tag{16}$$

カメラ位置ノイズの方向，即ち極座標系における角度は，無作為に決定する．本実験で用いたノイズの強度は I_k^{iso}, I_k^{blur}, I_k^r の順に $[0, 0.3]$, $[0, 25]$, $[0, 0.05]$ にスケーリングされ，Albumentations library の関数に引数として渡した．

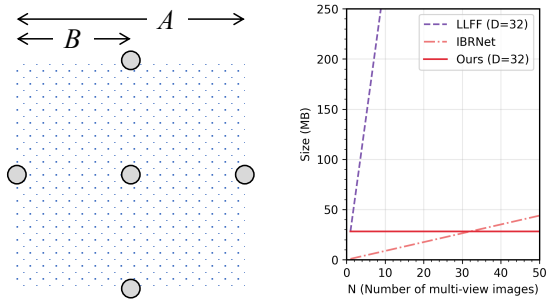
4.2　比較手法

提案手法，LLFF[10]，IBRNet[20] の 3 つを比較する．比較手法のいずれも再学習は行わず事前学習済み重みを用いた．尚，近年目覚ましい発展を見せる NeRF[11] に関連した手法はいずれも各シーンで再学習が必要なため比較対象から外した．

LLFF[10] は，我々の手法と同様，MPI を生成する手法であるが，入力画像のカメラ位置ごとに PSV を生成し，それらを一括してネットワークに入力するという点で提案手法とは異なる．よって，そのネットワークの構造上，使用画像枚数は 5 枚に限定されており，ユーザーがはサンプリング理論に従って疎にグリッド状に配置されたカメラ位置で写真を撮影する必要がある．IBRNet[20] は，輝度と体積密度を推定するビュー補間関数を学習する手法である．IBRNet は GPU メモリの限界が許す限りの枚数の入力画像を受け付けるが，MPI のように高速に描画できる媒体を生成するわけではないため自由視点画像生成自体は実時間実行できない．

表 2　ネットワーク推論で必要となる画像サイズ
Table 2　Image size for a network inference

Alg.	Input image size
LLFF	$W \times H \times C \times N \times D$
IBRNet	$W \times H \times C \times N$
Ours	$W \times H \times C \times D$



(a) Viewpoints for an inference　(b) Data amount for an inference

図 2　ネットワーク推論毎の入力視点と容量（VGA 画像）
Fig. 2　Number of viewpoints and data amount per a network inference (VGA resolution)

4.3　ネットワーク入力データ量

3 手法すべてで異なるネットワークを利用するため，メモリ消費量を公平に比較することが困難である．よって，各手法で必要とされる入力データ量の比較を行う（表 2及び図 2）．

図 2a に示された中心の 1 視点を生成するために取り込まれる画像枚数及び範囲を考える．提案手法の場合，図 2a に示された $A \times A$ の範囲内に存在するすべての画像をフォーカルスタックとして取り込む．そのため，1 回の MPI 生成（ネットワーク推論）当たりフォーカルスタックないし MPI のレイヤ数 D に依存する．入力画像中に局所的な誤差が含まれていても他で相殺されることが期待される．また，フォーカルスタックは任意の仮想視点で生成されるため，それぞれの間隔はサンプリング定理により保証された間隔 B を正確に保てる．

LLFF の場合，図中に示された 5 つの丸で示された画像のみが利用される．画像は PSV へと拡張され，それが視点数分必要となる．つまり，LLFF では N 近傍の画像の枚数分の PSV を入力する必要があるため，1 回の MPI 生成（ネットワーク推論）あたり N と MPI のレイヤ数 D に依存する．また，5 枚のうちのいずれかに誤差が含まれていた場合，致命的な描画誤差につながることも考えられる．加えて，ユーザをその視点にうまく導いて画像を撮影する必要がある．

IBRNet は N のみに依存するが，適切な N や視点間

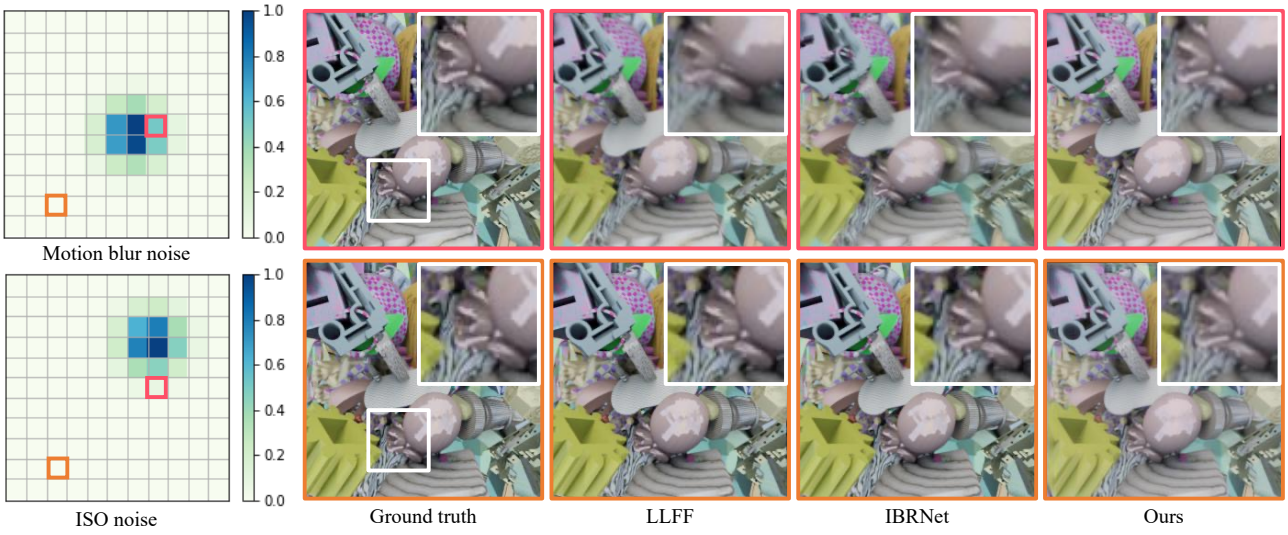


図 3　出力結果の定性的比較
Fig. 3　Qualitative comparison of rendered images

隔に関する議論はされておらず，さらには 1 枚の画像を描画するためにネットワーク推論が必要になることに注意されたい．さらに，撮影視点に関する指針は示されていない．よって，密に撮影された多視点画像を持っている場合は近い画像のみが入力され，より広い視点を取り込むためには N を大きくする必要がある．

4.4　ノイズ耐性

比較する 3 つの手法にどの程度のノイズ耐性があるのかを評価するため 4.1節で述べたデータセットを用いて自由視点画像生成を行った．尚，LLFF にはネットワークの限界値である $N = 5$，IBRNet には我々が用いた GPU のメモリで扱える最大の視点数 $N = 30$ を与えた．我々の手法では $N = 121$ 視点分の情報を取り込んだ 1 視点の MPI のみを生成している．本設定では，LLFF にはサンプリング定理を満たすに十分な密度の視点数を与えていることに相当する．

図 3に (a) 与えた視点位置に依存したノイズを可視化した図と (b) 2 視点で自由視点画像を描画した結果を各手法について示す．ノイズのない地点（図 3a の橙色の位置）ではいずれの手法も良好な結果が得られている．他の手法と比べると，我々の手法に関しては多少のぼやけた結果が得られている．一方で，誤差の多い地点に近づくにつれて，提案手法は一定の精度を保っているものの，他の手法ではボケた画像が得られていることが分かる．総じて，提案手法では有効範囲内全体で一定の安定した描画結果が得られ，他の手法では描画位置に応じて異なる精度の結果が得られていると言える．

5　考察

本稿では，フォーカルスタックから MPI を生成し，新たな視点の画像を描画するフレームワークを紹介してきたが，本手法の使用にはいくつかの制限が考えられる．

第一に，本手法は複数視点の画像をフォーカルスタックとしてエンコードすることで遮蔽部分を少ない MPI で保証することができる．よって，設定された合成開口サイズより外側に新視点を生成しても遮蔽部分が保証されない．そのため，合成開口サイズより大きな範囲の自由視点画像を生成したい場合は，2.3で述べたように複数の MPI を生成し，ブレンドする必要がある．第二に，本手法は大局的なノイズ，すなわち多くの入力画像にノイズが含まれる場合，焦点のあったフォーカルスタックが生成されず，結果出力される MPI もボケが大きくなってしまう．

本手法を利用するためには，必ずしも多視点画像の撮影から始める必要はない．フォーカルスタックを起点とすることも可能である．例えば，一眼レフなど，フォーカルスタックそのものを撮影できるカメラを使用すれば，既存手法では撮影困難な場面においても自由視点画像を生成できるだろう．例えば，薄闇での撮影において画像を撮影する場合，他の手法で必要な全焦点画像は限られた開口や高い ISO 値のせいで画像ノイズを避けることが難しい．また，そういった場面ではカメラ位置姿勢推定も困難になるだろう．一方，我々の手法の場合，開口を広げ，ISO 値を下げることができるため，ある視点でフォーカルスタックを撮影すれば，その近傍の自由視点画像生成が可能だろう．あるいは，顕微鏡写真について考えれば，レンズの微

小さにより被写界深度が非常に浅く，全焦点画像を得ることは困難であるが，フォーカルスタックは容易に収集することができる．

6 むすび

本稿では，位置姿勢が既知の多視点画像群から新視点画像を生成する新たなフレームワークとして，入力画像群より生成されるフォーカルスタックをネットワークへの入力とすることで MPI を生成し，自由視点画像を描画する手法を提案した．従来の MPI 生成法のように，1 つ，あるいは複数のカメラ位置に生成した PSV をネットワークへの入力とするのではなく，フォーカルスタックを入力とすることで，一定のデータ容量でより多くの視点情報を入力できる．これにより，提案手法は局所的にノイズが大きく加わっている画像の周辺においても，頑健に自由視点画像を生成可能であることを示した．

参考文献

[1] B. Attal, S. Ling, A. Gokaslan, C. Richardt, and J. Tompkin. Matryodshka: Real-time 6dof video view synthesis using multi-sphere images. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 441–459. Springer, 2020.

[2] M. Broxton, J. Flynn, R. Overbeck, D. Erickson, P. Hedman, M. Duvall, J. Dourgarian, J. Busch, M. Whalen, and P. Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4):86–1, 2020.

[3] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *Proc. Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 425–432, 2001.

[4] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. doi: 10.3390/info11020125

[5] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 307–318, 2000.

[6] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[7] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck, N. Snavely, and R. Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] P. Kellnhofer, L. Jebe, A. Jones, R. Spicer, K. Pulli, and G. Wetzstein. Neural lumigraph rendering. In *CVPR*, 2021.

[9] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, p. 31–42. Association for Computing Machinery, New York, NY, USA, 1996. doi: 10.1145/237170.237199

[10] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[11] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[12] E. Penner and L. Zhang. Soft 3d reconstruction for view synthesis. 36(6), 2017.

[13] T. Porter and T. Duff. Compositing digital images. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, p. 253–259. Association for Computing Machinery, New York, NY, USA, 1984. doi: 10.1145/800031.808606

[14] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[15] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics (TOG)*, 25(3):835–846, jul 2006. doi: 10.1145/1141911.1141964

[16] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision (IJCV)*, 32(1):45–61, 1999.

[17] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, vol. 41, pp. 703–735. Wiley Online Library, 2022.

[18] J. Thatte and B. Girod. Towards perceptual evaluation of six degrees of freedom virtual reality rendering from stacked omnistereo representation. *Electronic Imaging*, 2018(5):352–1, 2018.

[19] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane + parallax for calibrating dense camera arrays. In *In Proc. CVPR*, pp. 2–9, 2004.

[20] Q. Wang, Z. Wang, K. Genova, P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] L. Xiao, A. Kaplanyan, A. Fix, M. Chapman, and D. Lanman. Deepfocus: Learned image synthesis for computational displays. 37(6), 2018.

[22] Q. Zhou and A. Jacobson. Thingi10k: A dataset of 10, 000 3d-printing models. *CoRR*, abs/1605.04797, 2016.

[23] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *Proc. Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 2018.