

# 簡易に撮影したフォーカスタック画像からの 多層シーンの再構成に向けて

上田 栞<sup>\*1</sup> 斎藤 英雄<sup>\*1</sup> 森 尚平<sup>\*2\*1</sup>

**Abstract** – 多層画像 (Multi-Plane Image; MPI) は、3次元シーン撮影・編集、仮想・拡張現実感などに応用できる。MPI はライトフィールドないしフォーカスタックから生成でき、先行研究では専用に設計された撮影装置を用いる方法が主流であった。我々は、カメラのフォーカスリングを手動で回転してフォーカスタックを撮影し、それを MPI 再構成に用いるパイプラインを提案する。その際、フォーカスリングの回転方法に依存して MPI の結果がどう変化するかを分析する。そのために、我々は2つのフォーカスタック撮影方法を想定し、各撮影方法とその潜在的な誤差を合成・実画像データセットを用いて評価した。さらに、奥行画像描画、遮蔽を考慮したボケ描画、フェンス削除といった MPI を利用した応用例を示す。

**Keywords** : focal stack imaging, casual photography, multi-plane image, view synthesis

## 1 はじめに

フォーカスタックは光学的に焦点位置を変えながら撮影した一連の画像群で、2次元グリッド状に配置した多視点画像 (ライトフィールド) の近似表現 (拡散反射面を想定) として知られている [1]。フォーカスタックを介してライトフィールド情報を取得するためには、フォーカス制御が可能な撮影装置が必要となる。フォーカスタックは、ライトフィールドディスプレイ [2] や全焦点画像生成 [3]、自由視点画像生成 [4] といった幅広い応用への入力として利用できる。

フォーカスタックの取得方法には全焦点画像を加工する方法かフォーカス制御機構を備えた光学装置を用いる方法が考えられる。前者には、全焦点画像に奥行依存のブラーカーネル [3] やニューラルレンダリング技術 [5] を用いて近似的なぼけを合成する方法がある。ただし、全焦点画像をそのカーネルを見つけるのは容易ではない [6]。合成開口法によって多視点画像から光学的により正確なぼけを実現する方法がある [7] が、カメラ校正にひと手間必要である。後者には、モータ駆動レンズ [8] や電圧駆動の液体レンズ [9]、動的な撮像素子 [10] が用いられる。しかし、市場に出回る大多数のカメラにはこうした機能は備わっていない。

新たなフォーカスタック撮影方法として、我々は撮影者が手動でレンズのフォーカスリングを回しながら撮影した動画からフォーカスタックを取得する2つの方法に関して検討・分析する。1つ目は、撮影者がフォーカスリングを可能な限り線形に回転させる方法である (連続回転)。2つ目は、撮影者が時計の針のように短い時間間隔でフォーカスリングの回転と

停止を繰り返す方法である (デルタ回転)。より良い撮影方法を検証・選択するため、それぞれで発生する潜在的な誤差をシミュレーションした合成画像データセットを作成する。手持ちによるフォーカスタックの空間的なずれは画像位置合わせにより補正する。

合成開口写真から多層画像 (Multi-Plane Image; MPI) を生成する先行研究 [4] に基づき、提案する撮影方法で得られるフォーカスタックから MPI を生成する。MPI は、カメラの奥行方向に正対して並ぶ全焦点画像群であるため、提案手法の応用例には、全焦点画像生成、奥行画像生成、遮蔽を考慮したボケ描画、フェンス削除といった、MPI がサポートできるものが含まれる。これらの応用例についても本稿で示す。

本研究の貢献は以下の3点にまとめられる。

- 簡易に撮影されたフォーカスタックから MPI を生成するパイプラインの提案 (図 1(a-d))
- 手動制御による2つのフォーカスタック撮影方法 (連続回転とデルタ回転) が、合成画像データセットにおける MPI 描画品質にどのような影響を与えるかの検証 (図 1(a))
- 実シーンでの応用例の提示 (図 1(e))

## 2 多層シーンの再構成パイプライン

図 1 に、我々が提案する簡易に撮影したフォーカスタック画像から MPI 生成のためのパイプラインを示す。まず、カメラのメタデータやレンズの開口サイズをもとに、MPI を生成することができる理論的な最小距離と最大距離の範囲を計算する。ユーザは、求めた距離範囲内でフォーカスタック動画を撮影する。これは、一度にフォーカスリングを回しきる (連続回

<sup>\*1</sup>慶應義塾大学

<sup>\*2</sup>グラーツ工科大学

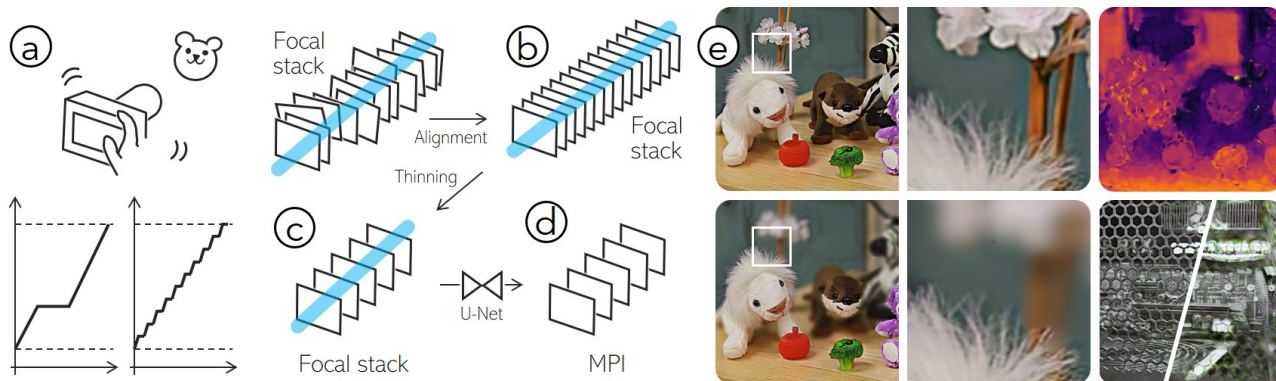


図1 手で撮影したフォーカスタック画像からMPI生成のためのパイプライン  
Fig.1 Pipeline of MPI generation from a casual focal stack with a manual lens camera.

転) (図1(a)左下), または, 短い時間間隔でフォーカスリングの回転と停止を繰り返す (図1(a)右下) ことによって行われる. 撮影されたフォーカスタックは, 隣接フレーム間でホモグラフィ変換によって位置合わせされる (図1(a-b)). その後, フォーカスタック動画の全フレームから MPI 生成に用いる画像を選択し (図1(b-c)), U-Net に似たネットワークに入力し, MPI を生成する (図1(c-d)).

### 2.1 2つのフォーカスタック撮影方法

一般に動画撮影用カメラは各フレームでの焦点位置を記録しない. そこで, 我々は, 最初と最後のフレームでの焦点位置のみを記録し, その間のフレームでは, 最小と最大の奥行値の逆数の差を等間隔で分割することにより焦点位置を求める. しかし, フォーカスタック動画撮影方法に依存して, 焦点位置が線形に現れるとは限らない. そこで, 我々は取得するフォーカスタックの焦点位置を可能な限り均等にするために2つの戦略について考察する (図2).

**連続回転:** 1つ目の戦略は, フォーカスリングを最小・最大距離間でできるだけ連続的に回転させながら撮影する方法である. この方法の懸念点は, (1) フォーカスリングの回転動作の線形性がユーザ依存となること, そして, (2) 撮影対象の奥行範囲が広くフォーカスリングを1度に回転しきれなかった場合, 回転動作が分離してしまうことにより連続性が失われてしまうことが挙げられる (図2(a-b)).

**デルタ回転:** 2つ目の戦略は, 1つ目の課題を克服する方法として考案した (図2(c)), フォーカスリングを最小距離から最大距離に達するまで, 一定のリズムで小刻みに回転し続ける方法である. この方法では, 回転速度とステップ間隔にばらつきが生じる, ステップ間隔が十分に小さくない場合, その量子化誤差が大きくなる懸念される.

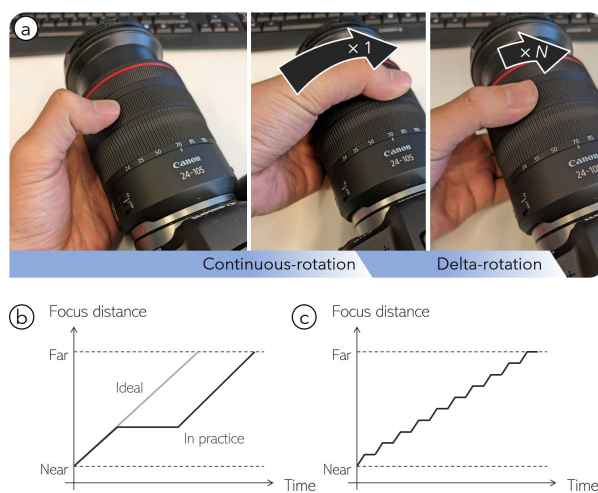


図2 2つのフォーカスタック撮影方法  
Fig.2 Two focal stack photography strategies.

### 2.2 フォーカスタック画像の選別

我々は, フォーカスタック動画から  $N$  枚選別してネットワークの入力として用いる. MPI を生成するために, フォーカスタック画像は, 全ての画素が  $N$  枚の画像のうち少なくとも1枚の画像で焦点が合っていることを保証しなければならない. 石川ら [4] は, 視野角や画像サイズといったメタデータと撮影する奥行範囲をもとに合成開口サイズの理論的境界を導出した. 一方, 我々は, 固定レンズの開口サイズとメタデータから, MPI 生成可能な奥行範囲を導出する. 開口サイズ  $A$ , 視野角  $\theta_{fov}$ , ピクセル単位での画像幅  $W_{px}$ , フォーカスタックの枚数  $N$  が与えられた場合, 最小焦点位置  $d_{min}$  と最大焦点位置  $d_{max}$  は,

$$\frac{1}{d_{min}} - \frac{1}{d_{max}} \leq \frac{4C_{px} \tan(\theta_{fov}/2)(N-1)}{AW_{px}} \quad (1)$$

によって制約される. ただし,  $C_{px}$  はピクセル単位での許容される最大視差である. 撮影設定が決まれば開口サイズとメタデータも確定するため, ユーザが最小焦点位置を設定すると共に最大焦点位置も計算可能で

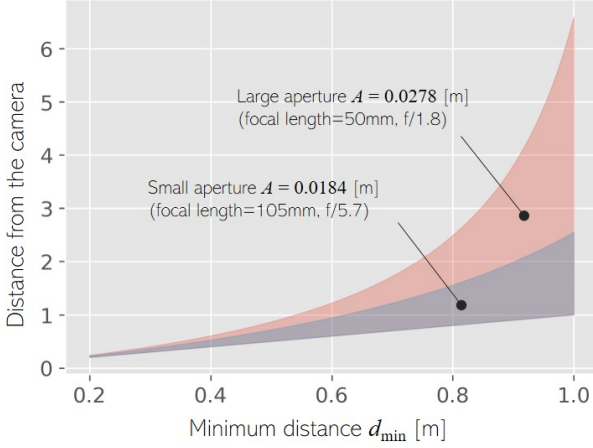


図3 開口サイズと対応奥行範囲の関係  
Fig. 3 Supporting depth ranges depending on the aperture sizes.

ある（逆も同様）．図3は， $N = 32$ ,  $W_{px} = 1920$  の場合の，MPI生成が可能な奥行範囲を撮影設定（焦点距離とF値）と共に示している．

我々は， $d_{min}$  と  $d_{max}$  の間で，フォーカスリングは一定のスピードで回転するという仮定に基づき，奥行値の逆数の差が可能な限り等間隔になるようにフォーカスタック動画から  $N$  枚の画像を選択する．

### 2.3 フォーカスタックの位置合わせ

カメラを固定せずにフォーカスタックを撮影すると，カメラの揺れにより時間の経過と共に光学中心や光軸のずれが発生する．結果として得られるMPIでの視覚的なアーティファクトを避けるためにこのような空間的なずれは修正する必要がある．フォーカスタック内の任意の2枚の画像間では，焦点位置が異なるほど焦点ぼけにより視覚的な違いが大きくなる．したがって，我々は焦点位置の差が小さい隣接フレーム間で位置合わせを行う．隣接する2枚の画像  $I_i$  と  $I_{i+1}$  間でホモグラフィ変換を推定し，変換を適用した後の全画素値  $I_i(H(\cdot))$  の誤差を最小化するように最適化する [11]．すなわち，

$$\arg \min_{\mathbf{p}} \sum_{\mathbf{x}} [I_i(H(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p})) - I_{i+1}(\mathbf{x})] \quad (2)$$

となるホモグラフィ変換パラメータ  $\mathbf{p}$  を求める．ただし， $\mathbf{x} = (x, y)^\top$  は2次元の画素位置， $\mathbf{p} = (p_1, p_2, \dots, p_8)^\top$  と  $\Delta \mathbf{p} = (\delta p_1, \delta p_2, \dots, \delta p_8)^\top$  はそれぞれ，ホモグラフィ行列のパラメータと推定される増分パラメータである．

この処理を，全ての隣接フレームペア  $i$  と  $i+1$  ごとに適用する．図4に，フォーカスタックの位置合わせ有無によるMPI描画結果の違いを示す．上段は，カメラを固定せず手持ちでフォーカスタックを撮影している．位置合わせを行わなかった場合，MPI描画結果が不安定になる（左上）のに対し，行った場合は

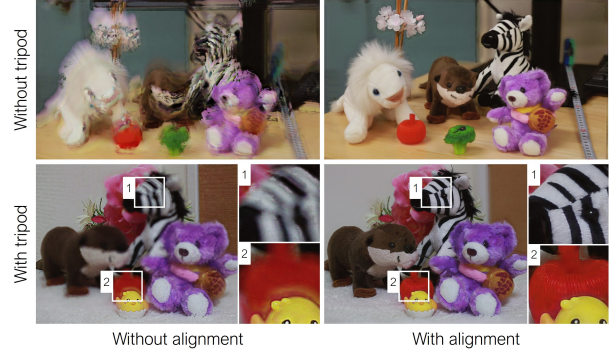


図4 フォーカスタックの空間的なずれと補正  
Fig. 4 Misalignment and correction.

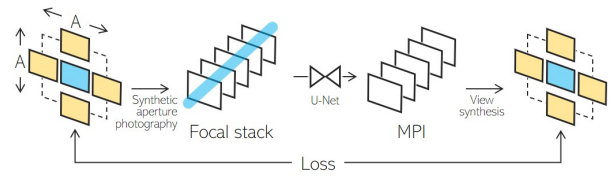


図5 MPI生成ネットワークの学習  
Fig. 5 Training by synthesis.

アーティファクトを抑制できている（右上）．下段は，三脚を用いてフォーカスタックを撮影している．この場合でも，位置合わせを行わなかった場合（左下）よりも行った場合（右下）の方がMPI描画品質が高い．

本稿では，簡潔のために静的なシーンに焦点を当てている．しかし，動的なシーンでは，画素単位の位置合わせが必要である．この場合，PatchMatchアルゴリズム [12] に基づく方法 [9, 3] が利用できる．

### 2.4 深層学習によるMPIの生成

我々は，フォーカスタックからMPIの生成において，石川ら [4] と同様のネットワーク構造を使用した．一方，ネットワークの学習には，効率化のために石川らと異なる損失関数を用いた．ライトフィールド（多視点画像）データセットが与えられると，5視点でのMPI描画画像を評価することで，MPI生成ネットワークを学習する（図5）．

まず，中心視点において， $N$  個の計算された焦点位置でフォーカスタックを生成する．生成された  $N$  枚のフォーカスタックはU-Netに似た畳み込みニューラルネットワーク (Convolutional Neural Network; CNN) に入力される．ネットワークは中心視点での  $N$  層のMPIを出力する．各MPI層は，フォーカスタック画像と同じ高さを持つ  $\text{RGB}\alpha$  画像である．出力されるMPI層は  $d_{min}$  と  $d_{max}$  の間で奥行値の逆数で等間隔に配置されていると仮定する．微分可能な描画関数は後方から前方の層に向かってアルファ合成を行い，新視点での画像を生成する [13]．

我々は，損失関数  $\mathcal{L}$  を設計し，5視点（中心，上，

下, 左, 右) での全体の損失を最小化する. すなわち, 最適化の目的関数は,

$$\arg \min_W \sum_{v \in V_{ref}} (\mathcal{L}(\mathcal{R}^v(I^{MPI}), I_{gt}^v) + \mathcal{L}(\mathcal{R}^v(I^{MPI}), I_{gt}^v)) \quad (3)$$

となる. ここで,  $\mathcal{R}^v(I^{MPI})$  と  $\mathcal{R}^v(I^{MPI})$  は共に, 視点  $v$  での MPI 描画画像である. ただし,  $\mathcal{R}^v(I^{MPI})$  は MPI をそのまま描画しているのに対し,  $\mathcal{R}^v(I^{MPI})$  は RGB 色を入力フォーカスタックの色に置換して描画している.  $I_{gt}^v$  は視点  $v$  での正解画像 (データセット内の参照視点画像) である.

損失関数  $\mathcal{L}$  は L1 loss,  $\mathcal{L}_{L1}$ , と VGG16[14] をバックボーンとした perceptual loss,  $\mathcal{L}_{Percept}$ , から構成され,

$$\mathcal{L} = \mathcal{L}_{Percept} + \lambda \mathcal{L}_{L1} \quad (4)$$

と表される. 本稿の実験では,  $\lambda = 0.1$  に設定している.

## 2.5 新視点画像生成

新視点画像生成  $\mathcal{R}^v(I^{MPI})$  は, MPI を生成した後に, 入力フォーカスタックの開口の範囲 (すなわち, 中心視点から半径  $A/2$  の範囲) で可能である.

## 3 性能評価

我々は, 2 種類の手動によるフォーカスタック撮影方法が含む潜在的な誤差が, 生成される MPI の品質にどの程度影響を与えるかを, 合成画像フォーカスタックと実画像フォーカスタックの両方で評価した.

### 3.1 データセット

我々は, 異なる誤差を独立にシミュレーションし, 定量的に評価するために, 合成画像データを使用した. Blender[15] を使用して描画された  $11 \times 11$  枚の多視点画像からなるライトフィールドデータセットを作成した. そこから, 合成開口法 [7, 4] によってフォーカスタックを生成した. カメラは水平方向と垂直方向に  $56.2475^\circ$  の視野角を持ち, 画像サイズを  $256 \times 256$  画素とした. データセットの各シーンは, Thingi10K データセット [16] からランダムに配置された 3 次元物体で構成される. 本実験では, 合計で 120 シーンを描画し, 学習, 検証, テストにそれぞれ 80 シーン, 20 シーン, 20 シーンを使用した.

また, 定性評価のために実シーンのデータセットも作成した. 実シーンのフォーカスタック動画を撮影するために, デジタル一眼レフカメラ (Canon EOS 6D, SIGMA 50mm F1.4 DG HSM) を使用した. 撮影中, F 値は 1.8, 画像サイズは  $1920 \times 1080$  画素, フレームレートは 30fps に設定した.

### 3.2 MPI 生成ネットワーク学習の詳細

我々は, PyTorch [17] v2.1.0 を用いて深層学習ネットワークを実装した. ネットワークの学習には NVIDIA

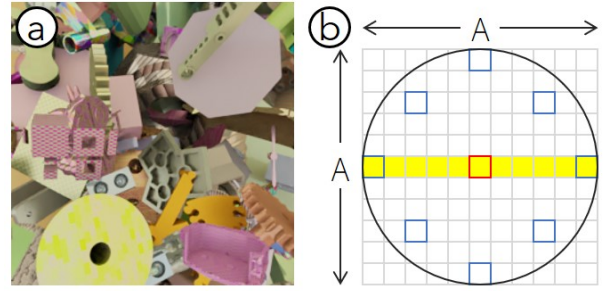


図 6 合成画像データセットのセットアップ  
Fig. 6 Our synthetic dataset setup.

Titan RTX 24GB VRAM を, 推論には, NVIDIA GeForce RTX 3080 10GB VRAM を用いた.

ネットワークは, 学習率が  $10^{-4}$ , 重み減衰が  $10^{-8}$ , モーメントが 0.9 の RMSprop を用いて, ネットワークを学習した. バッチサイズは 4, エポック数は 614 であった. 我々の実験では, フォーカスタックと MPI の枚数  $N$  を 32, 最小奥行  $d_{min}$  を 1.0 メートル, 最大奥行  $d_{max}$  を 10.0 メートルに固定した.

### 3.3 リング回転誤差の設計

MPI 描画品質におけるフォーカスリング回転誤差の影響を定量的に評価するために, 2 つのフォーカスタック撮影方法での理想的な撮影と潜在的な誤差をモデル化した. 回転中の全フレームの焦点位置は後述のシミュレーションにより得た. 全フレームの焦点位置から  $N$  個の焦点位置をサンプリングして合成フォーカスタックを生成し, ネットワークの入力とした. 図 7(a) は連続回転, (b) はデルタ回転でのシミュレーションとその結果を示しており, (a) と (b) それぞれの上段に, 全フレームのシミュレーションされた焦点位置とサンプリングされた距離を示す. 青色の線と点は奥行値で左縦軸に, 赤色の線と点は奥行値の逆数で右縦軸に対応している. 共に, 実線が全フレームの焦点位置で, 点がサンプリングされた距離である.

理想的な連続回転では, 焦点位置の逆数は  $1/d_{min}$  から  $1/d_{max}$  に達するまで, 1 フレームあたり  $v_{mean}$  だけ直線的に減少する. 潜在的な誤差として, 焦点位置の逆数の減少量を平均  $v_{mean} = 0.003$  と標準偏差  $v_{std} = \{0.001, 0.002, 0.003\}$  のガウシアン分布から得た. ただし, 減少量は負の値にならないように制限した. また, ユーザが途中でフォーカスリングの回転を一時停止することを想定して, フレーム  $i = \{50, 150, 250\}$  から, フレーム数  $p = 50$  の間焦点位置を一定にする. 3 通りの停止パターンをそれぞれ, 序盤停止, 中間停止, 終盤停止と呼ぶ. 理想的なリング回転では, 300 フレーム (30fps の場合, 10 秒間を想定) で焦点位置の移動が完了するように設定している.

理想的なデルタ回転では, 焦点位置の逆数は  $1/d_{min}$

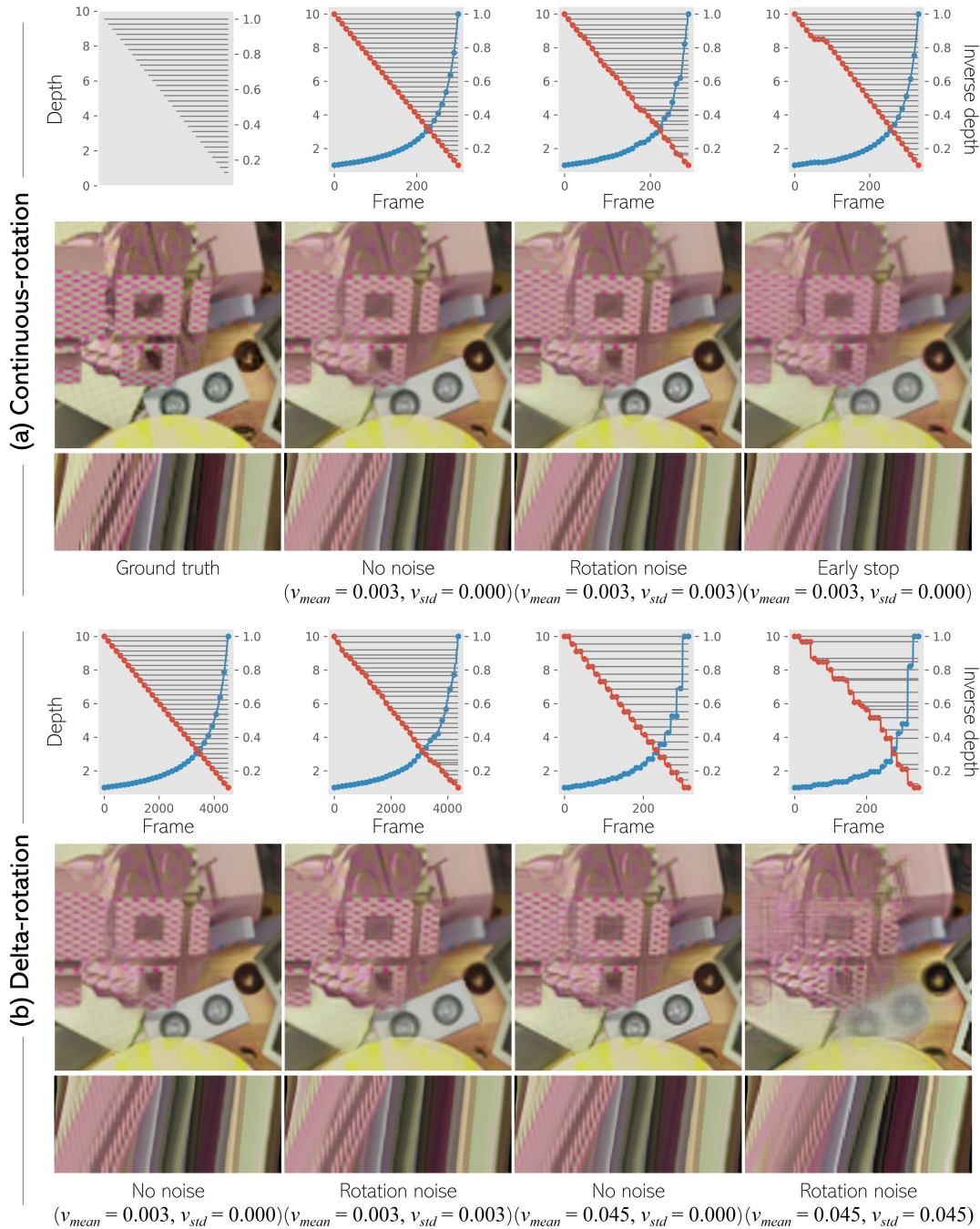


図7 合成画像データセットにおける MPI 描画結果の定量的比較  
 Fig. 7 Qualitative comparison of MPI rendering results on a synthetic dataset.

から  $1/d_{max}$  に達するまで,  $s = 15$  フレームごとに,  $v_{mean}$  ずつ減少する. 連続回転と同様, 潜在的な誤差として, 焦点位置の逆数の減少量を平均  $v_{mean} = \{0.003, 0.045\}$  と標準偏差  $v_{std} = \{v_{mean} \times 1/3, v_{mean} \times 2/3, v_{mean} \times 1\}$  のガウシアン分布から得た. ただし, 減少量は負の値にならないように制限した.

### 3.4 結果と考察

図7(a-b) の中段に, 合成画像データセットでの MPI 描画の結果を示す. また, 表1は, 合成画像データセットで実験した全フォーカスリングの回転方法における

精度を示している. 我々は, MPI 描画画像の精度として, 正解画像との PSNR, SSIM, LPIPS を計算した. 連続回転とデルタ回転を比較する. デルタ回転は, ステップ間隔が小さく十分な時間が与えられた場合 (図7(b), 1列目の4000フレームのように), 描画結果は連続回転と同様の品質である. しかし, フォーカスタック撮影の時間が限られておりステップ間隔が大きい場合, 図7(b), 3列目の300フレームのように, 品質は大幅に低下する. どちらの戦略も, ガウシアン標準偏差が大きいほど, 精度が低下する傾向にある.

表 1 合成画像データセットにおける MPI 描画結果の定量的比較  
Table 1 Quantitative comparison of MPI rendering results on a synthetic dataset

Focus ring modulation	$v_{mean}$	$v_{std}$	PSNR ( $\uparrow$ )	SSIM ( $\uparrow$ )	LPIPS ( $\downarrow$ )
Continuous Rotation (No noise)	0.003	0.000	21.748 (2.856)	0.8422 (0.0406)	0.0992 (0.0269)
Continuous Rotation (Rotation noise)	0.003	0.001	21.715 (2.839)	0.8385 (0.0413)	0.1006 (0.0272)
Continuous Rotation (Rotation noise)	0.003	0.002	21.609 (2.802)	0.8282 (0.0454)	0.1059 (0.0273)
Continuous Rotation (Rotation noise)	0.003	0.003	21.549 (2.716)	0.8221 (0.0439)	0.1116 (0.0270)
Continuous Rotation (Early stop)	0.003	0.000	20.702 (2.870)	0.7372 (0.0822)	0.1230 (0.0305)
Continuous Rotation (Middle stop)	0.003	0.000	21.026 (2.649)	0.7754 (0.0629)	0.1282 (0.0288)
Continuous Rotation (Late stop)	0.003	0.000	19.733 (2.582)	0.6568 (0.0902)	0.1592 (0.0289)
Delta Rotation (No noise)	0.003	0.000	21.741 (2.846)	0.8408 (0.0408)	0.0999 (0.0271)
Delta Rotation (Rotation noise)	0.003	0.001	21.679 (2.847)	0.8340 (0.0426)	0.1017 (0.0271)
Delta Rotation (Rotation noise)	0.003	0.002	21.690 (2.787)	0.8378 (0.0406)	0.1040 (0.0267)
Delta Rotation (Rotation noise)	0.003	0.003	21.398 (2.746)	0.8103 (0.0506)	0.1134 (0.0266)
Delta Rotation (No noise)	0.045	0.000	21.609 (2.718)	0.8269 (0.0419)	0.1088 (0.0268)
Delta Rotation (Rotation noise)	0.045	0.015	21.257 (2.815)	0.7897 (0.0533)	0.1191 (0.0270)
Delta Rotation (Rotation noise)	0.045	0.030	20.410 (2.236)	0.7139 (0.0656)	0.1780 (0.0264)
Delta Rotation (Rotation noise)	0.045	0.045	17.832 (2.557)	0.4662 (0.1369)	0.2221 (0.0315)



図 8 実画像データセットにおける MPI 描画結果の定量的比較  
Fig. 8 Qualitative comparison of MPI rendering results on a real-scene dataset.

特に、デルタ回転で連続回転と同じ時間で撮影した場合、移動量のガウシアンノイズによる誤りが描画品質の低下に大きく影響している (図 7(b), 4 列目)。

連続回転における停止は、どの停止パターンであっても、MPI 描画品質に大きい影響を与える。図 7(a) の中段にある中心視点からの MPI 描画結果を見ると、回転の途中停止による誤り (図 7(a), 4 列目) は、移動量のガウシアンノイズによる誤り (図 7(a), 3 列目) よりも描画品質が劣り、ぼけが強く発生している。

図 7(a-b) の下段に、図 6(b) に黄色い線で示した、中心視点を通る水平の 11 視点の画像で作成したエピソード画像を示す。エピソード画像は視差を表している。ノイズのない場合は視差の再構成に成功した一方、ノイズが加わると品質が劣化していることが見て取れる。

図 8 は、実シーンでのレンズの開口範囲で左上の視点と右下の視点からの MPI 描画結果を示す。フォーカ

スリングの回転以外の要素を可能な限り固定化するために、カメラは三脚に固定し、最小焦点位置と最大焦点位置を揃えた。デルタ回転では、回転を発生させる時間間隔を可能な限り均一にするために、メトロノームで回転タイミングをとった。この設定では、連続回転とデルタ回転で同等の結果となった。

#### 4 応用例

実シーンのフォーカスタックから取得した MPI を利用したアプリケーションを示す。先述の評価実験における MPI の描画結果は、MPI を利用した全焦点画像生成にあたる。ここでは、奥行画像生成、遮蔽を考慮したボケ画像生成、フェンス削除の応用例を示す。

MPI の各層のアルファ値は、その層の奥行における物体の確からしさを表す。したがって、各層の全画素値 (RGB) にその層の奥行値を割り当てることで、自



図9 MPI を利用した三つのアプリケーション

図10 Three applications using MPI.

由視点の奥行画像を描画できる。図10(a)に、その結果と元のRGB値での自由視点画像生成結果を示す。

MPIの各層は、カメラの光軸方向に沿って明示的に分離されており、異なる層に異なるブラーカーネルを適用すると奥行依存のブラーカーネルを実装できる。そのため、エッジを考慮する必要なく、遮蔽を考慮したボケ画像が生成できる。我々は、遮蔽を考慮したボケ画像生成として、近距離層をぼかす前景のぼかし処理と遠距離層をぼかす背景のぼかし処理を生成した。ぼかし処理対象の層は、カーネルサイズが $25 \times 25$ のガウシアンフィルタでぼかした。この結果、図10(b)で示すように、前景と背景の物体の境界が明瞭なままぼかし処理が適用できている。

MPIでは、近距離層を描画から除去することでその奥行きにある物体を取り除くことができる。その有用な例として、フェンスや格子を除去してその後ろにある物体を可視化するフェンス除去がある。その一例として、コンピュータの内部を外側から撮影した。図10(c)で示すように、コンピュータの内部の部品やRTXのロゴがケースによって元々は隠されているが、フェンス除去によって可視化されている。

## 5 むすび

本稿では、簡易に撮影したフォーカスタックからMPIを生成するパイプラインを示した。我々は、2つのフォーカスタック撮影方法を提案し、合成画像データセットを用いて定量的・定性的に評価した。また、実シーンでのフォーカスタックを用いた定性的な評価を行った。その結果から、特に撮影時間が限られている場合、デルタ回転よりも連続回転の方が高いMPI描画精度であることが示された。最後に、生成されたMPIを利用した3つのアプリケーションを提示した。

本研究で議論が残るのが、フォーカスタック撮影において我々がシミュレーションした誤差の妥当性である。手法の限界としては、フォーカスタックの位置補正が局所的なものでなく物体が移動する動的なシーンや激しい手ぶれに対応していない点が挙げられる。今後の課題としては、実際にカメラのフォーカスリングを操作する様々な参加者を募るユーザースタディを行うことが考えられる。また、画素単位のフォーカスタックの位置合わせや、MPI推論を高速化するための効率的なネットワークの設計も進める。

## 謝辞

本研究は、JST次世代研究者挑戦的研究プログラムJPMJSP2123の支援、JSPS科研費JP23H03422とAustrian Science Fund FWF (No. P33634)の助成を受けたものである。

## 参考文献

- [1] F. Pérez, A. Pérez, M. Rodríguez, and E. Magdaleno, "Lightfield recovery from its focal stack," *Journal of Mathematical Imaging and Vision*, vol. 56, no. 3, pp. 573–590, 2016.
- [2] K. Takahashi, Y. Kobayashi, and T. Fujii, "From focal stack to tensor light-field display," *IEEE Trans. on Image Processing*, vol. 27, no. 9, pp. 4571–4584, 2018.
- [3] H. Kim, C. Richardt, and C. Theobalt, "Video depth-from-defocus," in *Proc. Int. Conf. on 3D Vision*, 2016, pp. 370–379.
- [4] R. Ishikawa, H. Saito, D. Kalkofen, and S. Mori, "Multi-layer scene representation from composed focal stacks," *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, vol. 29, no. 11, pp. 4719–4729, 2023.
- [5] Z. Wu, X. Li, J. Peng, H. Lu, Z. Cao, and W. Zhong, "Dof-nerf: Depth-of-field meets neural radiance fields," in *ACM Int. Conf. on Multimedia*, 2022, pp. 1718–1729.
- [6] A. Abuolaim, M. Delbraccio, D. Kelly, M. S. Brown, and P. Milanfar, "Learning to reduce defocus blur by realistically modeling dual-pixel data," in *Proc. Int. Conf. on*

*Computer Vision (ICCV)*, 2021, pp. 2289–2298.

- [7] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy, “Using plane+ parallax for calibrating dense camera arrays,” in *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2004, pp. 1–1.
- [8] M. Subbarao and T. Choi, “Accurate recovery of three-dimensional shape from image focus,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 17, no. 3, pp. 266–274, 1995.
- [9] C. Ebner, S. Mori, P. Mohr, Y. Peng, D. Schmalstieg, G. Wetzstein, and D. Kalkofen, “Video see-through mixed reality with focus cues,” *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, vol. 28, no. 5, pp. 2256–2266, 2022.
- [10] S. Kuthirummal, H. Nagahara, C. Zhou, and S. K. Nayar, “Flexible depth of field photography,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 1, pp. 58–71, 2011.
- [11] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *Int. J. of Computer Vision (IJCV)*, vol. 56, pp. 221–255, 2004.
- [12] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patchmatch: A randomized correspondence algorithm for structural image editing,” *ACM Trans. on Graphics (TOG)*, vol. 28, no. 3, p. 24, 2009.
- [13] T. Porter and T. Duff, “Compositing digital images,” in *Proc. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, ser. SIGGRAPH ’84. New York, NY, USA: Association for Computing Machinery, 1984, p. 253–259. [Online]. Available: <https://doi.org/10.1145/800031.808606>
- [14] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in *Proc. Asian Conf. on Pattern Recognition (ACPR)*, 2015, pp. 730–734.
- [15] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: <http://www.blender.org>
- [16] Q. Zhou and A. Jacobson, “Thing10k: A dataset of 10, 000 3d-printing models,” *CoRR*, vol. abs/1605.04797, 2016. [Online]. Available: <http://arxiv.org/abs/1605.04797>
- [17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>