

3次元焦点ぼけ構造に基づく実画像からのFPGAを用いた自由視点画像生成 FPGA-based Free Viewpoint Image Reconstruction by Using Real 3-D Multi-focus Imaging Sequences

伊澤逸平太[†] 野々下俊[†] 児玉和也[‡] 浜本隆之[†]
Ippeita IZAWA Shun NONOSHITA Kazuya KODAMA Takayuki HAMAMOTO

1. まえがき

筆者らは、単一のレンズ系のみを用い、精細な3次元画像情報を取得する手法を提案している。本手法では、適切に焦点合わせを変化させながら撮像した画像群を3次元信号とみなし、その3次元情報を2次元情報へ統合、周波数領域上で線形の2次元フィルタ処理を施すことで、多数枚の焦点ぼけ画像群から、対象シーンの構造推定を介することなく、直接に所望の自由視点画像等を生成する [1]。

本手法は単純な信号処理で構成され、FPGAを用いた高速化が期待できる。しかし、FPGA上の内部メモリ(BlockRAM)の容量のみでは、例えば64×64画素程度までの解像度にしか対応できない [2]。本稿では、FPGA評価ボード上の外部メモリ(DDR SDRAM)を活用し、画像分割された64×64画素単位で繰り返し処理を適用することで、自由視点画像生成システムの高解像度化を検討する。実画像を用いたシミュレーションにより、その評価を行う。

2. 自由視点画像生成手法

まず、図1に示す様に、適切に焦点合わせを変化させながら撮像した N 枚の画像群を構造をもった3次元情報 $g(x, y, z)$ として統合する。また、各撮像画像について、対応する撮像面で合焦する対象奥行き平面のみ注目し、その奥行き平面から得られた輝度値成分だけを取り出した画像群 $f(x, y, z)$ を定義する。これらは3次元ぼけフィルタ $h(x, y, z)$ を用いた畳み込み演算により、図1の様に空間位置不変の恒等式で結び付けることができる [1]。

ここで、下式また図2の様に、 z 軸または z' 軸(視差 (s, t) により傾きが変化する)に沿った平行な直線群にそった線積分を用いて2次元情報を新たに定義する。

$$\begin{aligned} a(x, y; s, t) &= \int f(x, y, z) dz', \\ b(x, y; s, t) &= \int g(x, y, z) dz', \\ c(x, y) &= \int h(x, y, z) dz. \end{aligned}$$

このとき、図1の空間位置不変の恒等式は以下のように変形できる。

$$\begin{aligned} b(x, y; s, t) &= c(x, y) * a(x, y; s, t), \\ B(u, v; s, t) &= C(u, v)A(u, v; s, t). \end{aligned}$$

ここで、 $C^{-1}(u, v)$ は高周波成分の強調特性を持たないので、以下の式により所望の自由視点画像 $a(x, y)$ を安定に再構成できる。

$$A(u, v; s, t) = C^{-1}(u, v)B(u, v; s, t).$$

[†]東京理科大学大学院 工学研究科

[‡]情報・システム研究機構 国立情報学研究所

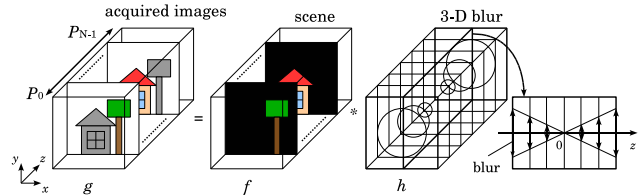


図1: A 3-D space-invariant equation.

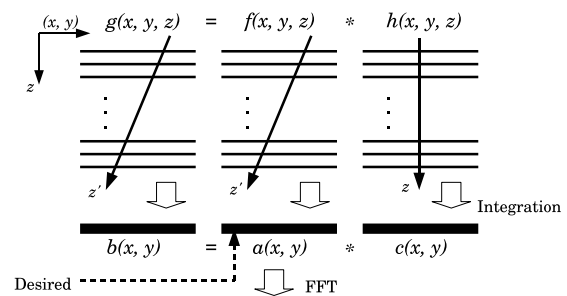


図2: Dimension reduction.
 $B(u, v) = A(u, v)C(u, v)$

結局、(1)視点に対応した方向への次元削減(各撮影画像の周辺4画素を用いた双一次内挿と全画像の加算)、(2)FFT、(3) C^{-1} によるフィルタリング(複素乗算)、(4)逆FFT、の順序により所望の自由視点画像が生成される。

なお、画像中心位置の補正 [1] のため、視差 (s, t) に対し、 k をカメラパラメータによって決まる正の値として、生成した自由視点画像に $-k(s, t)$ の平行移動を施す。これは上記の順序(1)に簡単に組み込むことができる。

3.FPGAによる自由視点画像生成

筆者らは既に、Xilinx社製のFPGAであるVirtex-4を搭載したML402評価ボード上に、本手法を実装した。しかし、FPGA上の内部メモリであるBlockRAMの容量の制約から、最大解像度は64×64画素となっている [2]。そこで、より大きな容量(32MB)を有するML402評価ボード上の外部メモリであるDDR SDRAMを活用することで、本システムの高解像度化をはかる。BlockRAMと外部メモリとの間で64×64画素の入出力を実装した結果、それぞれ0.013ms、0.012msを要するが、これは従来の処理速度である0.594msと比較して十分に小さい。

本実装のみでなく一般に、デバイスのリソースの制約の範囲で、取り扱う画像の解像度や撮像枚数、演算の精度を調整することになる。それぞれのパラメータの組み合わせの数だけ最適化された回路の検討を行う

ことは現実的に難しく、ある程度柔軟なスケラビリティが強く望まれる。

そこで本稿では、各処理が必ずしも画像全体を必要としないという提案手法の特徴を活かし、図3に示すように対象画像を64×64画素にブロック分割し、それぞれの分割画像を最適化された既存の回路によって処理を行い、その結果を合成して最終的な高解像度画像を出力する実装を採用した。

また、処理時間を削減するため、図4に示すような二段階のパイプラインを採用した。一段目はSDRAMからBlockRAMへの焦点ぼけ画像16枚の読み出しと次元削減処理、二段目は2D-FFT、フィルタリング、2D-IFFTで構成される。それぞれの処理のパイプラインピッチはほぼ等しいので効果的な処理時間の削減を期待できる。

4. 実画像を用いたシミュレーション

図5(a)に示すような、120×120画素の16枚の実画像からなる焦点ぼけ画像群を用いて実験を行った。今回の実験では3次元ぼけパラメータは既知として $r = 1.33$ を用いた。

再構成した自由視点画像を図5(b)に示す。実際に、120×120画素の焦点ぼけ画像群を64×64画素の単位ブロックに分割し、既存の自由視点画像生成回路を繰り返し適用することで、実装上のスケラビリティを保持しつつ、より高解像度の自由視点画像を2.4msと高速に生成できた。背景の模様(二つの小さい丸点)に対して、手前の物体の視差が大きく生成されており、水平視点位置 s に応じた適切な視差が付与されていることがみてとれる。このFPGAを用いた自由視点画像生成システムによって、256×256画素程度の自由視点画像も10ms以内と十分実時間で再構成することが期待できる。

5. まとめ

焦点ぼけ画像群からの実時間での自由視点画像生成を目的として、FPGAへの実装に基づく高速化の検討を行った。本稿では特に、FPGAのリソースの制限を考慮した場合でも、画像をブロック分割してそれぞれに本手法を適用することで、実時間で所望の画像生成が実現できることを明らかにし、実画像を用いてその検証を行った。

検討すべき課題として、オーバーラップ領域の合成処理がある。境界領域は各分割画像の周辺画素を切り落として接続しただけであるので、接続境界で不連続性を有し、視覚的な劣化が大きい。これを抑制するため、同領域においては、複数の分割画像の生成結果を重み付け合成し、分割画像間を滑らかに接続することが考えられる。画像周縁部の対策も同様に必要である。

今後の目標として、解像度がQVGA程度のカラー画像を対象とした実時間自由視点画像生成システムを試作したい。

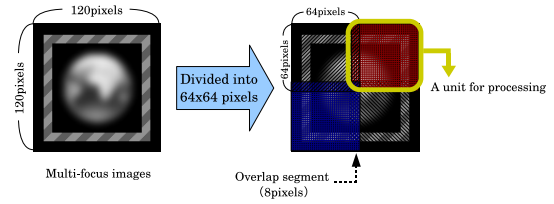


図3: Dividing images into blocks of 64×64 pixels.

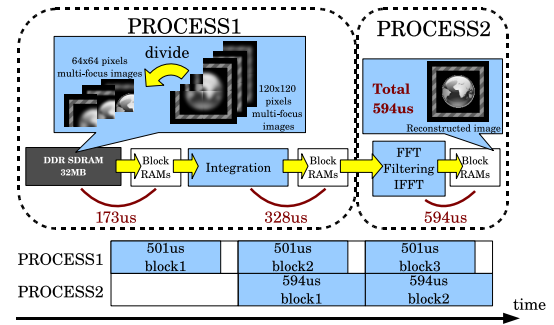
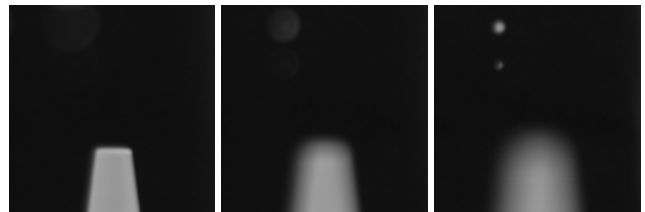
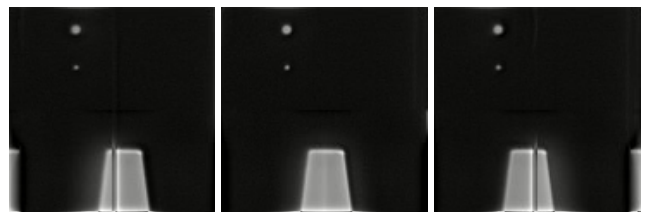


図4: Our pipeline structure.



(a) Acquired (input) $g(x, y, z)$ ($z = 4, 8, 12$)



(b) Generated (output)
 $a(x, y; s, t)$ ($(s, t) = (-0.5, 0), (0, 0), (0.5, 0)$)

図5: Free viewpoint image reconstruction.

参考文献

- [1] 欧他: “3次元焦点ぼけ構造からの次元削減に基づく自由視点画像生成”, 映像メディア処理シンポジウム (IMPS 2007), I-4.19, pp.109-110, 2007
- [2] I.Izawa et al., “Free viewpoint image reconstruction from 3-D multi-focus imaging sequences and its implementation by FPGA-based processing,” in *Proc. of IEEE International Conference on Signal and Image Processing Applications 2009*, pp.415-418, Nov. 2009