

FPGA によるホログラム高速イメージングシステムの開発 Development of high-speed holographic imaging system on FPGA

谷口 甲斐[†] 増田 信之[†]
Kai Taniguchi Nobuyuki Masuda

1. はじめに

デジタルホログラフィは光波面を巧みに記録することで、2 次元画像であるホログラムから 3 次元の空間情報をコンピュータによって再生する技術である。この時、フレネル・キルヒホッフの回折積分に代表されるような、ある面からある面への光の伝搬を複数回実行する必要がある。一般的に回折計算の実行には、計算量の観点から FFT (高速フーリエ変換) を使用した計算方法が用いられる。

FFT は非常に優れたアルゴリズムではあるものの、デジタルホログラフィでは、十分な計算速度が得られているとは言えないのが現状である。また、本研究ではこれら課題に対して FPGA による専用計算機の作成を通じて取り組むが、FPGA での FFT の実装はデータの依存性から、データを 1 画素ずつ計算回路に供給し、内部でキャッシングし処理するのが一般的である。しかし、このような仕様は専用計算機での効率的な処理を目指す上では解決すべき課題である[1-3]。

本研究は上記のような課題に対して、複数ピクセルを同時に計算回路に供給し処理可能な専用計算機を、FPGA を用いて設計し、その有用性について検討する。また、FPGA の実装効率を高めるため、基数 4 の Vector-Radix 型 FFT を用いて計算回路を構成し、最適化を施した。

2. フレネル回折

光波面 $u_1(x_1, y_1)$ から光波面 $u_2(x_2, y_2)$ への光の伝搬は、レイリー・ゾーマーフエルトの回折積分ないしはフレネル・キルヒホッフ回折積分式によって計算される。ここで、 $u_1(x_1, y_1)$ の任意の位置から $u_2(x_2, y_2)$ の任意の位置への距離に近似解を与えたものがフレネル回折である。次の式 (1) にフレネル回折の計算式を示す。

$$u_2(x_2, y_2) = \frac{\exp\left(i\frac{2\pi}{\lambda}z\right)}{i\lambda z} \iint_{u_1} u_1(x_1, y_1) h(x_2 - x_1, y_2 - y_1) dx_1 dy_1 \quad (1)$$

ただし

$$h(x_2 - x_1, y_2 - y_1) = \exp\left(i\frac{\pi}{\lambda z}\{(x_2 - x_1)^2 + (y_2 - y_1)^2\}\right) \quad (2)$$

である。また、 i は虚数単位、 λ は光波長、 z は光波面に垂直な軸上の位置である。ここで (1) 式は畳み込み定理より、フーリエ変換を利用した形に変形可能であり、FFT を使用可能である。また係数項は無視しても数値計算上の問題はない。よって $H(f_x, f_y) = \mathcal{F}[h(x_2 - x_1, y_2 - y_1)]$ とすると、式 (1) は式 (3) のように変形できる。

$$u_2(x_2, y_2) = \mathcal{F}^{-1}[\mathcal{F}[u_1(x_1, y_1)]H(f_x, f_y)] \quad (3)$$

ここで、 $H(f_x, f_y)$ は解析解が求めることができるため、フーリエ変換を使用せず直接計算が可能である。コンピュータでは、この (3) 式を離散化し計算する。

3. 専用計算機の構成

本研究で開発した専用計算機は、FFT 計算回路へのデータの並列供給を行う。この時、画像データ先頭から順番に FPGA 内部 RAM に格納していくと、計算回路で必要とされるデータを必要なタイミングで取り出すことができない。そこで本専用計算機では、FPGA 内部 RAM への格納方法を工夫し、必要なタイミングでのデータ取り出しを可能にしている。以下、図 1 に画像データと、格納を行う内部 RAM の番号関係を示す。ただし、ホログラム画像は 256×256 画素、各画素 32bit (実部 16bit, 虚部 16bit) であり、これを 32bit 長、深さ 256 の 0~255 番までの 256 個の内部 RAM に格納している。

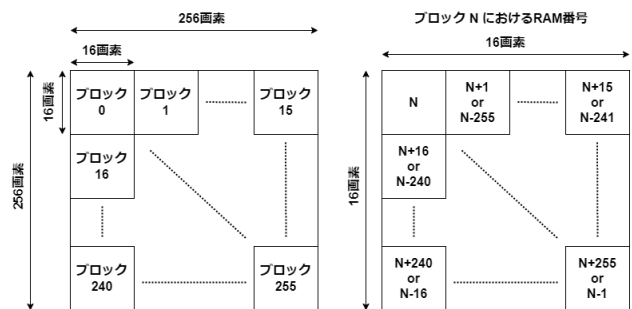


図 1 画像データの内部 RAM への格納方法

図 1 には内部 RAM の番号のみ示したが、図 1 の各 16×16 ブロックごとに、左上端から、RAM アドレス 0 番地を割り当てていけば、画像はうまく格納できる。つまり、画像左上端は 0 番の RAM のアドレス 0 番地に格納し、その右の画素は 1 番の RAM のアドレス 1 番地に格納する。

ここまで、画像の RAM への割り当てを示したが、次に具体的な処理過程について示していく。本専用計算機では、 16×16 画素のデータを常に流し、これらを並列処理していく。例えば、FFT は離散フーリエ変換 (DFT) をより小さい DFT に分解するアルゴリズムであるが、計算回路では 1 ステップで 256×256 の DFT を 16×16 に分解する。この分解は、 16×16 画素ごとに行われるので、256 クロックで完了する。FFT はこれを 2 ステップ行うので、512 クロックで FFT の分解が行われ、この後データのビット反転整理を実行し、FFT は 768 クロックで完了する。本専用計算機は、768 クロックでの FFT、256 クロックでの複素乗算、768 クロックでの IFFT を行うため、パイプライン遅延時間を除けば、1,792 クロックで計算が完了する。

[†] 東京理科大学 Tokyo University of Science

ここで、ここまで説明した仕様を満たせる、実際に設計した計算部の構成を次の図 2 に示す。

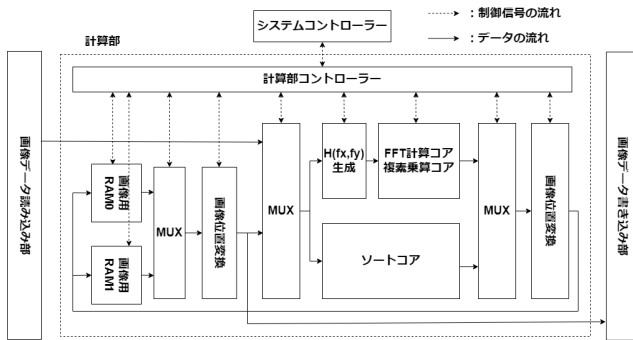


図 2 専用計算機の計算部の構成

図 2 では実線でデータの流れる、破線で制御の流れを示している。データについては、 $H(f_x, f_y)$ 生成ブロックから FFT 計算コアの間以外は、32bit 長 16×16 画素でデータが流れている。画像位置変換のブロックは、内部 RAM から取り出されたデータと計算回路の画素供給の位置関係を調整するためのモジュールである。

4. 実装結果

本専用計算機の実装には、Xilinx 社が提供する Alveo U250 アクセラレータカードを使用した。この U250 は UltraScale+ XCU250-2LFIGD2104E という FPGA が搭載されている。また、専用計算機の開発には、Vitis2020.2 の RTL カーネルによる実装および、Vivado2020.2 の bitstream 生成によるカスタム実装の 2 つの手法を使用した。本専用計算機は 50 MHz で動作する。次の表 1 に時間計測の結果を示す。ただし、時間計測は RTL カーネルの場合には `clGetProfilingInfo()` を用いて測定し、カスタムプラットフォームでは、`clock_gettime()` を使用して測定した。また、計算時間は、専用計算機の計算部にデータが供給される準備が整った段階から、計算部をスタートさせて、終了を CPU 側が検出するまでの時間になっている。

表 1 専用計算機の計算時間

実装方法	計算時間 [us]
カスタム (Vivado フロー)	42.8
RTLカーネル (Vitis フロー)	123.2

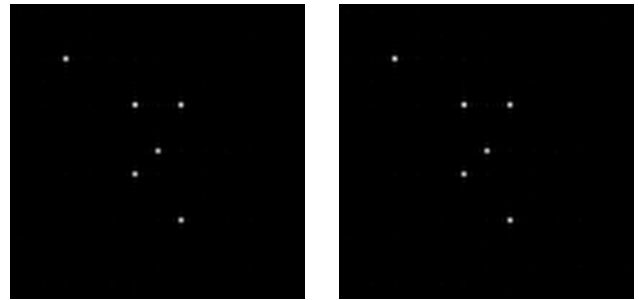
専用計算機での計算は前述したように、パイプライン遅延時間を除けば 1,792 クロックで動作する。ここにパイプライン遅延時間や制御時間を加えれば、概ね 1,900~2,000 クロックで動作するため、理論上は 40 us 程度の結果が期待される。カスタム実装では概ね想定される結果となったが、RTL カーネルによる実装では予期しない時間が含まれている結果となった。

次の表 2 に専用計算機の配置配線後のリソース使用状況を示す。表 2 を見ると、概ねリソース使用状況は、2 つの実装手法の間で変わらないことがわかる。ここで一点留意すべきは、カスタム設計では、PCIe 経由で転送されたホログラムを、そのまま FPGA の内部 RAM に格納できるように、BRAM を用意しているのに対して、RTL カーネルでは U250 搭載の DDR メモリを介す仕様となっている点である。

表 2 専用計算機のリソース使用率

リソース名	実装方法	使用数	使用率 [%]
LUT	カスタム	364,576	21.10
	RTLカーネル	376,470	21.81
BRAM	カスタム	892	33.18
	RTLカーネル	837	31.14
DSP	カスタム	2,240	18.23
	RTLカーネル	2,244	18.26

最後に、専用計算機で実際にサンプルホログラムをイメージングした結果を、次の図 3 に示す。サンプルのホログラムには、6 点の物体点より作成した位相型コンピュータ合成ホログラム (CGH) を使用した。



(a) 倍精度演算の結果 (b) 専用計算機の結果
図 3 ホログラムのイメージング結果

図 3 の結果を比較すると、イメージングという観点においては、専用計算機の精度は十分であろうことがわかる。

5. まとめと今後の課題

本研究で作成された専用計算機は、256 並列で FFT, IFFT を実行し、期待された計算結果を得ることができた。

その一方で、その動作周波数は 50 MHz であり、これ以上の周波数では、Vivado および Vitis がタイミングを満たせず配置配線できていない。表 2 を見ると、全体のリソース使用率はそれほど高くない。それにも関わらず、このような低周波の動作となるのは、配置配線の局所性が原因であろうことが、Vivado のレポートより判明している。本研究の FPGA チップは、4 つの SLR と呼ばれる領域を相互に接続し構成されている。ここで、この SLR 間の配線上限数が、計算部を FPGA 全体に分散させるほどの余裕がなく、1 つの SLR に計算部が集中的に配置されていることを確認している。今後の研究では、この SLR 間配線の問題を改善できるよう、pblock 制約などの対策を講じていく必要がある。また、本専用計算機は、当初 RTL カーネル実装を前提として設計されたため、Vivado によるカスタム実装では、無駄な内部 RAM を使用しているが、これらを削減することが間接的に上記問題の改善に繋がる可能性もある。

参考文献

- [1] Y.Yamamoto, S.Namba, T.Kakue, T.Shimobaba, T.Ito, N. Masuda, "Special-purpose computer for digital holographic high-speed three-dimensional imaging", *Optical Engineering*, Vol. 59, No. 5 (2020).
- [2] Y.Abe, N.Masuda, H.Wakabayashi, Y.Kazo, T.Ito, S.Satake, T.Kunugi, K.Sato, "Special purpose computer system for flow visualization using holography technology", *Optics Express*, Vol. 16, No. 11 (2008).
- [3] Xilinx, "Fast Fourier Transform v9.1 LogiCORE IP Product Guide", <https://docs.xilinx.com/r/en-US/pg109-xfst> (2022)