GPU による相互相関 PIV の高速化に関する研究

47096716 田良島周平
指導教員　岡本 孝司 教授

This study aims to develop a fast Particle Image Velocimetry(PIV) software by Graphics Processing Unit(GPU). This summary consists of 3 parts. In the first part, the methods to accelerate recursive direct/FFT-based cross-correlation PIV by GPU are discussed. After the comparison, GPU accelerated cross-correlation PIVs are implemented on GPU and about 800 times faster processing is achieved compared to using single CPU. In the second part, the methods to accelerate these PIVs by multiple GPUs(multi-GPU) are also discussed. By parallelizing single GPU accelerated PIV by multi-GPU, about 85% of the theoretical performance can be obtained compared to using single GPU. In the third part, recursive hybrid cross-correlation PIV scheme is proposed, which uses FFT-based method in the former step and use direct method in the latter step. Real PIV images are used to validate hybrid cross-correlation PIV and the advantages in terms of its range, precision, and time cost are shown.

Key words: Direct cross-correlation PIV, FFT-based cross correlation PIV, Hybrid cross-correlation PIV, GPGPU, Multi-GPU

1　緒言

PIV(Particle Image Velocimetry: 粒子画像流速測定法) は，流れ場の瞬時・多点の速度計測をおこなう手法として広く用いられている．現在に至るまでに PIV の実験手法および解析アルゴリズムに関する様々な研究がなされ，それらが計測機器の発達と相まった結果，PIV により観測困難な様々な流体現象の可視化および解明がおこなわれてきた．一方で PIV 解析の計算負荷は増大傾向にあり，たとえば高速度カメラの時間解像度と比較すればその時間コストは非常に高い．

本研究では高速かつ実用に足る PIV 画像処理システムの開発を目標とし，その手段として Graphics Processing Unit(GPU) の有用性を検証する．本研究は具体的に以下の項目について検討評価をおこなう．

- シングル GPU を用いた直接/FFT 相互相互相関 PIV の高速化手法の検討
- マルチ GPU を用いた相互相関 PIV の高速化手法の検討
- 直接法と FFT 法を融合したハイブリッド相互相関 PIV の提案とその評価

本要旨では，高速化対象の相互相関 PIV と高速化手段の GPU について概説したあと，各項目における研究の概要と結果について簡潔に示す．

2　再帰的相互相関 PIV

相互相関 PIV は，局所的な輝度値パターンの類似度を相互相関の値で評価する手法である．相互相関 PIV の計算方法には，定義式に基づき計算をおこなう直接相互相関法 (直接法) と，FFT を利用して相互相関を計算する FFT 相互相関法 (FFT 法) が考えられる．本研究で高速化を検討する直接法は (1) に示す相互相関係数の式を，FFT 法は (2) に示す規格化なし相互相関関数の式を用いる．式 (1),(2) に関して，$f(x)$, $g(x)$ は 1 枚目および 2 枚目画像内の位置 x における輝度値，f_m，g_m は検査領域内の緯度値平均である．

PIV 処理では，導出するベクトルの数だけ画像中に検査領域 (Interrogation window) を設定し，各領域における相互相関のピーク位置および値を検出することで速度場を得る．速度場はガウス補間によりサブピクセルオーダーまで求めている．

$$C_{Direct} = \frac{\sum_{m}^{IW}\{f(\boldsymbol{m}) - f_m\}\{g(\boldsymbol{m} + \boldsymbol{u}(\boldsymbol{k})) - g_m\}}{\sqrt{\sum_{m}^{IW}\{f(\boldsymbol{m}) - f_m\}^2 \sum_{m}^{IW}\{g(\boldsymbol{m} + \boldsymbol{u}(\boldsymbol{k})) - g_m\}^2}} \tag{1}$$

$$C_{FFT} = \sum_{m}^{IW} f(\boldsymbol{m}) g(\boldsymbol{m} + \boldsymbol{u}(\boldsymbol{k})) \tag{2}$$

本研究では，高ベクトル解像度かつ高精度な PIV を実現する手段として用いられる，再帰的相互相関 PIV [1] の GPU への実装をおこなう．この方法では相互相関 PIV は複数ステップ繰り返される．前のステップの結果から検査領域の移動先を限定し，検査領域はステップが進むたびに小さくすることで高精度かつ高解像度な PIV を実現している．

3　GPU

最新の GPU では 1000 以上のプロセッサコアを搭載したマシンも存在し，これらが並列に動作することで GPU は高い性能を示す．GPU の性能を計算処理に活用するためには，並列処理への効果的な実装方法を検討する必要がある．本研究では 480 コアを搭載する NVIDIA 社製 GPU の GTX 480 を用いてシングル GPU 並列計算の実装をおこなう．

シングル GPU による計算に加え，マルチ GPU による並列計算も考えられる．マルチ GPU では GPU 単体の更に数倍のプロセッサコアを利用することが可能となるが，GPU 間の直接データ通信ができないといったマルチ GPU 固有の制約も存在する．シングル GPU での並列化手法がそのまま適用できるわけではないため，マルチ GPU の特徴に適した並列化を検討する必要がある．本研究では 240 コアを搭載する GTX 285 を最大 4 基用いてマルチ GPU 並列計算への実装をおこなう．なお開発環境としては NVIDIA 社が提供する CUDA を用いている．

4　シングル GPU による相互相関 PIV の高速化

本研究が対象とする再帰的相互相関 PIV は，相互相関の計算・相関のピーク検出およびサブピクセル計算・統計処理による誤ベクトル除去・ベクトルの補間といった要素から構成される．本研究では CPU ⇔ GPU 間のデータ転送を極力減らすため，これらの要素を全て GPU で計算する．以下では，各要素について GPU 並列計算への適用方法を述べる．

4.1　直接相互相関法

直接法で計算される各相互相関は計算上すべて独立である．この特徴に着目した並列化の手法として，GPU の並列処理単位であるスレッドひとつに 1 相互相関を対応させる方法が考えられる．この方法ではスレッド間データ通信を考慮する必要がなく，また待機状態となるスレッドも発生しないことがメリットであるが，一方で相互相関計算時に必要となる条件分岐命令の負荷がすべてのスレッドにかかり，これが本手法のパフォーマンスを下げる要因となる．条件分岐を回避する方法として，複数のスレッドを用いて 1 つの相互相関計算

をおこなう方法も考えられる．この方法では待機状態となるスレッドが発生するものの，条件分岐命令を使用することなく相関の計算をおこなうことができる．

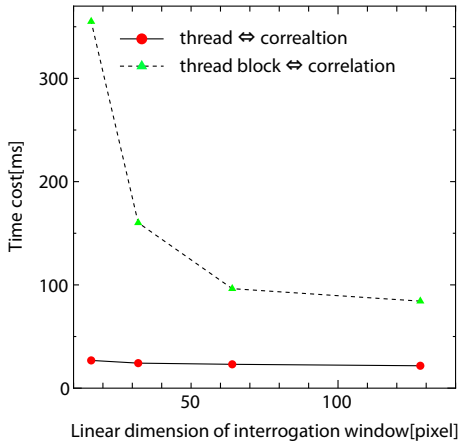


Figure 1: Time cost of direct cross-correlation method as a fuction of interrogation window size

Fig. 1 に，各検領域サイズを変化させたときの両手法の処理時間を示す．この結果は，1024 × 1024[pixel] の PIV 画像に対し検査領域を 50%ずつオーバーラップするよう設置し，各検査領域に 20 × 20[pixel] の探査領域を設定して得られたものである．少なくとも本研究の計算環境では，検査領域サイズによらず 1 スレッドに 1 相互相関を割り当てた手法の性能がもう一方に比べ圧倒的に高いことが確認された．その性能差は検査領域サイズが小さくなるにつれ大きくなっているが，これは領域サイズが小さくなることによりスレッドブロックあたりの計算負荷が下がり，待機状態になるスレッドの割合が相対的に高くなることに起因すると考えられる．検査領域サイズが大きくなるにつれ手法間の差は縮まるが，それでも最低 4 倍以上の差があることを考えると，他の計算環境においても待機状態のプロセッサが発生しない 1 スレッドに 1 相互相関の計算を割り当てた手法のほうが効果的な高速化を実現できる可能性が高いと考えられる．

4.2 FFT 相互相関法

FFT 相互相関法による相互相関の計算は，両画像に設定された各検査領域に対する 2 次元 FFT，対応する検査領域ごとの複素要素積とその結果に対する 2 次元逆 FFT から構成される．本研究では 2 次元 FFT の計算方法には Row-Column 法を用い，その際計算される 1 次元 FFT の計算には並列化が容易な Stockham の FFT アルゴリズムを用いて実装をおこなった．ここで開発された 1 次元 FFT の性能は最高で 150GFLOPS 弱であり，これは本研究で使用した GPU の理論性能の 10%程度にあたる．

FFT 相互相関法を構成する各要素ごとに GPU 処理単位であるカーネルを構成することで，GPU 処理としての最適化は実現されるが，一方で低速なチップ外メモリを介したデータ通信が多く発生し，複数カーネルの呼び出しに伴うオーバーヘッドも増大する．これらの問題を回避するためには，複数の要素をひとつのカーネルとしてまとめる方法が考えられるが，その場合まとめた処理内での並列度は統一されてしまい，要素ごとに最もパフォーマンスが上がる実装方法を採用できない可能性がある．本研究では，FFT 相互相関法を構成する要素処理ごとに GPU 処理をおこなう方法 (計 10 カーネル) と，2 次元 FFT の計算をひとつの GPU 処理としてまとめた方法 (計 4 カーネル) とを実装し，その処理速度の比較をおこなった．Fig. 2 にその結果を示す．Fig. 2 は，各検査領域サイズでの FFT 法による相関計算の処理時間を示している．この結果は 1024 × 1024[pixel] の PIV 画像に検査領域を 50%ずつオーバーラップするように設定したときに得られたものである．

大きな検査領域では処理要素ごとにカーネルを構成した方法の処理速度が他方を上回っているが，検査領域が小さくなるとその性能差が逆転していることがわかる．これは検査領域が小さくなるにつれて，低速なメモリを介したデータ転送の負荷が計算量に対して相対的に増大するためだと考えられる．したがって他の計算環境での実装にあたっては，設定する検査領域サイズに則して手法を選択することが効果的な高速化につながるといえる．なお GPU 処理の呼び出しに伴うオーバヘッドは 10 カーネルで約 0.2ms，4 カーネルで約 0.05ms と処理コスト全体に対して無視できるほど小さく，結果にはほぼ影響していない．しかし小さな画像サイズを用いたケースでは全体の計算負荷が低いため，オーバーヘッドの影響が相対的に大きくなることが確認された．

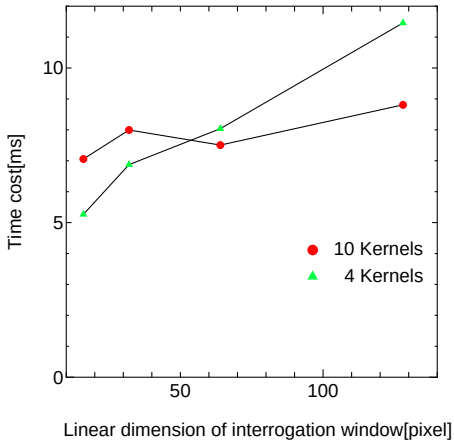


Figure 2: Time cost of FFT-based cross-correlatiaon computing as a function of interrogation window size

4.3 その他の要素の並列化

相互相関のピーク検出とサブピクセル計算はひとつの GPU 処理にまとめ，複数スレッドで構成される同期可能な 1 スレッドブロックに 1 つの検査領域に関する処理を割り当てて並列化する．再帰的手法で用いられるベクトル場の補間もまた，補間される 1 ベクトルを 1 スレッドブロックに割り当てて並列化する．統計処理に基づく誤ベクトル除去は，誤ベクトルの判定が近傍のベクトルに依存するため並列化することが難しい．本研究ではスレッドブロックを 1 つだけ使用し，並列化された行ごとの誤ベクトル除去を，列要素の数だけ繰り返している．

4.4 PIV の処理速度

以上の検討をふまえた上で，現在広く使用されている再帰的相互相関 PIV の構築をおこなった．シングル CPU での処理同様，PIV の処理時間の大部分を占めるのは相互相関の計算部分であり，再帰を繰り返すほどにその処理時間は長くなる．再帰の回数や検査領域サイズ，直接法の場合には探査領域のサイズなど，PIV の処理時間を決める要素は多々あり，一概に評価することは難しい．ここでは一例として，堀田 (2011) がおこなった 2 円柱の流体励起振動の実験で得られた PIV 画像に，本研究で開発した再帰的相互相関 PIV による処理をほどこした結果を示す．画像サイズは 512 × 1024[pixel] である．

Fig. 5 の (a) および (b) が，この画像に直接相互相関 PIV および FFT 相互相関 PIV をほどこした解析結果であり，その処理時間は Table 1 に示されている．ここで示す処理時間は，GPU のメモリへ画像データの転送を開始した瞬間から，

ベクトルのデータを CPU のメモリに確保するまでの時間と定義している．両手法とも再帰は二回おこなわれ，検査領域は $128 \times 128 \rightarrow 64 \times 64 \rightarrow 32 \times 32$[pixel] と変化する．直接法における探査領域は $20 \times 20 \rightarrow 10 \times 10 \rightarrow 6 \times 6$[pixel] と設定した．両手法とも処理時間は 20ms を下回っており，したがってこの条件であれば 1 秒で 50 画像ペアに PIV 処理が可能なことがわかる．参考としてシングル CPU を用いた同条件下での直接相互相関 PIV による結果も Table 1 に示しているが，GPU による直接相互相関 PIV 処理ではこの 800 倍弱，FFT 相互相関法では 1100 倍弱の高速化が実現されていることがわかる．

ここで示した条件下での FFT 相互相関 PIV の処理速度は，直接法を上回っている．しかしその解析結果である Fig. 5 を見ると，FFT 法による PIV 処理では現象のおおまかな解析はできているものの，誤ベクトルが多く発生しておりこれが全体の精度を低下させている．一方直接法による PIV 処理では目立った誤ベクトルの発生もなく現象が可視化されており，したがってこれらの結果は FFT 法と比較した直接法の精度がよりロバストであることを示した一例であるといえる．なおこの可視化結果は誤ベクトル除去を加えたうえでの結果であるため，FFT 法によるこの解析結果から誤ベクトルを除去するためには画像変形などの付加的な処理を検討する必要があるが，ここではこれ以上の議論はおこなわない．

5　マルチ GPU を用いた PIV の高速化手法の検討

シングル GPU を用いた検討に続き，本研究ではマルチ GPU を用いた相互相関 PIV の並列化手法について検討をおこなう．マルチ GPU 並列計算では GPU 間データ転送に非常に時間がかかるため，極力データ転送をおこなわないかたちで実装をおこなう必要がある．通常 PIV 処理は，時系列画像データの複数ペアに対して時系列に沿って実行されるため，マルチ GPU を用いた高速化の方針は，1.)PIV 処理単位を高速化する方針 と，2.) 前章で示したシングル GPU による PIV 処理を複数 GPU で並列に計算する方針 の二通りが考えられる．

本研究では方針 1.) の具体的な手法として，使用する GPU の数で画像を分割しそれら小画像にたいして PIV 処理をおこなう方法を検討した．この方法では GPU 間のデータ通信は発生させずに複数 GPU に処理を分散させることができる．しかし本手法の開発評価をおこなったところ，シングル GPU 使用時の時間コストが高い場合をのぞきその性能はシングル GPU による処理を下回ってしまった．これは本研究で用いたマルチ GPU 計算環境ではミリセカンドオーダーの並列性で各 GPU へデータを転送することができず，実質並列処理がおこなわれなかったことに起因する．

一方で方針 2.) では各 GPU の同期をとることなく処理すること自体は可能であるものの，その場合処理順序は実際に実行するまで不明であり，時系列順に処理がおこなうことができない．そこで本研究では，各 GPU と対応している CPU スレッド間での同期をとることでマルチ GPU を用いながら時系列処理が可能なシステムを開発し，シングル GPU による結果と比較したときの本システムの性能を計測した．Fig. 3 がその結果である．処理される PIV 画像ペア (1024×512[pixel]) の数を変化させたときの処理時間の推移と，シングル GPU 使用時 (ここでは 1 回の PIV 処理に約 80ms) と比較したときの本システムの性能向上度について，2GPU〜4GPU を使用した際の結果を示している．

Fig. 3 から，たとえば 4GPU を用いた PIV 画像 600 ペア (600MB) の処理は 15 秒足らずで実行可能であることがわかる．使用した GPU の数によらず本システムの性能がシングル GPU より高いことも確認できるが，その性能向上は最高で理論性能の 85%程度にとどまった．これは CPU ⇔ GPU 間のデータ転送がボトルネックとなっているためだと考えられる．このデータ転送の影響は PIV 処理単位の時間コストが減少するにつれ大きくなり，シングル GPU での処理が既に高速なケースではマルチ GPU の効果は低下してしまうことも確認された．

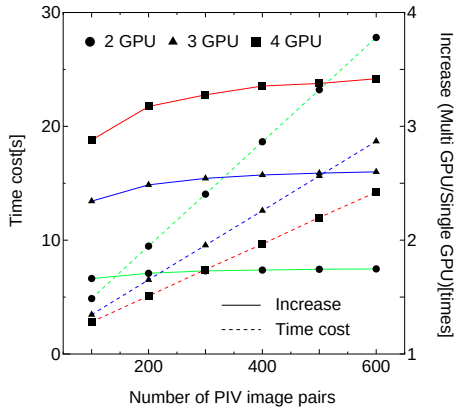


Figure 3: Time cost and performance improvement of multi-GPU accelerated cross-correlation PIV

6　直接法と FFT 法を併用するハイブリッド相互相関 PIV の提案と評価

相互相関 PIV の計算手法である直接法と FFT 法の違いを比較したとき，その最大の違いは各検査領域あたりに計算される相互相関の数である．FFT 法で検査領域あたりに算出される相互相関の数は検査領域サイズのみに依存するが，直接法において計算される相互相関の数は別途設定される探査領域 (Search region) によって決まり，この探査領域サイズは基本的に検査領域サイズによる制約を受けない．一般的には FFT 法の計算量は直接法に比べ少なくよって高速であるとされるが，こと PIV についていえばこれは必ずしも正しくない．実際には設定された探査領域サイズによって直接法の計算コストは大きく変わるため，直接法が FFT 法よりも高速に計算可能であるケースが考えられる．(1) 式および (2) 式をもとに計算量の比較をおこなうと，そのような探査領域サイズがいずれの検査領域サイズでも存在しうることが確認できる．本研究ではこのような，直接法の計算負荷が FFT 法より小さくなる最大の探査領域を臨界探査領域 (Critical seacrh region) と定義する．

本研究で開発した直接相互相関法と FFT 相互相関法との速度を比較し，各検査領域，各画像サイズにおける臨界探査領域をプロットしたものが Fig. 4 である．本研究の実装結果では，計算量のみの考察から算出した臨界探査領域よりも，実装結果から得られた臨界探査領域のほうが高いことがわかる．探査領域として 10×10[pixel] 以下のサイズを設定する限り，多くのケースで直接法の計算速度は FFT 法を上回っていることがわかる．

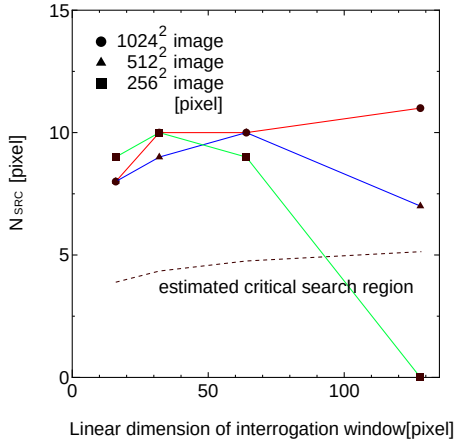


Figure 4: Critical search region size as a function of interrogation window size

臨界探査領域サイズを下回る探査領域を設定した場合，直接法の処理速度は FFT 法を上回る．FFT 法に対する直接法の精度のロバスト性の高さは Fig. 5 が端的に示している．探査領域を小さく設定することは直接相互相関 PIV のダイナミックレンジを低下させることにつながるが，たとえば再帰的手法の初回以降のステップではピーク位置が探査領域中央近傍に存在していることが前提であり，これらのステップにおいて小さな検査領域を設定することはダイナミックレンジの低下にはほぼ影響を与えない．これらの考察をふまえると，直接法と FFT 法とを融合したハイブリッド相互相関 PIV(Hybrid cross-correlation PIV，ハイブリッド法) の有効性が推察できる．すなわち再帰的相互相関 PIV において，最初のステップではダイナミックレンジが広くかつ高速な FFT 法を利用して PIV 処理をおこない，以降のステップでは臨界探査領域より小さな探査領域を設定した直接法による PIV をおこなうことで，直接法よりも高ダイナミックレンジで，FFT 法よりもロバスト性が高く，両手法よりも更に高速な PIV が実現する可能性がある．直接法および FFT 法が開発できている状況ならば，このハイブリッド法を実現することは難しくない．しかしこのハイブリッド法を処理速度の点から最適化するためには，初回以降で用いられる直接相互相関法における探査領域の設定が重要な意味をもつ．すなわちハイブリッド相互相関 PIV を開発するにあたり，使用する計算環境における臨界探査領域サイズが重要な意味をもつということになる．

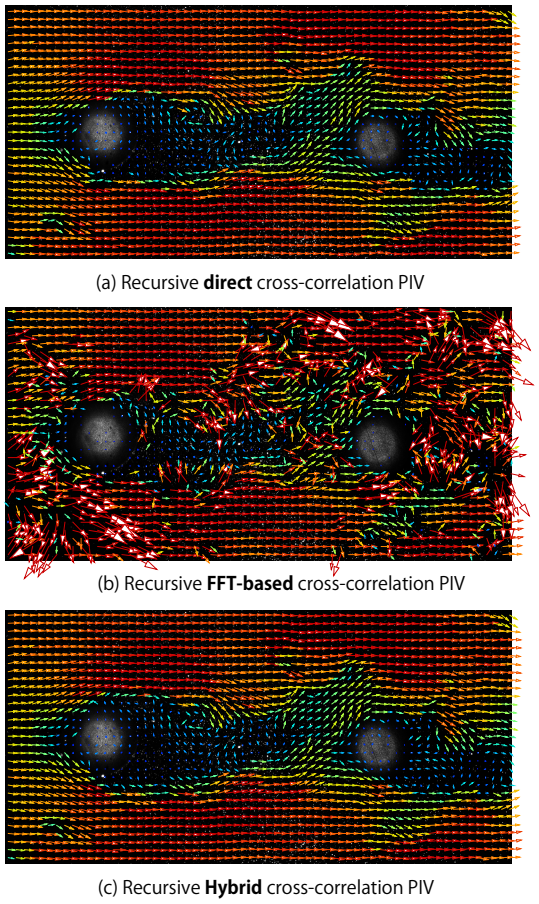
ハイブリッド相互相関 PIV 処理の結果の一例として，ここでも堀田 (2011)[3] の例を取り上げる．Fig. 5(c) および Table 1 にハイブリッド相互相関 PIV 処理の可視化結果および処理時間を示す．再帰的 PIV が用いられその検査領域は Fig. 5(a)，(b) と等しい．初回のステップは FFT 法が用いられ，続く 2 ステップには直接法を用いる．直接法における探査領域は $[8 \times 8, 6 \times 6]$[pixel] と設定した．FFT 法では誤ベクトルの発生が目立っているが，ハイブリッド法では最終ステップは直接法を用いているため，直接法のみで得られた結果 (a) に近い結果を得ることができている．一方でその処理時間は，直接法だけでなく FFT よりも高速な処理を実現しており，その高速化はシングル CPU 比約 1300 倍を記録した．この結果から，ハイブリッド相互相関 PIV が精度・速度の両者の観点から有用であることを示せたといえる．

Table 1: Time cost of cross-correlation PIV developed in this study

	Time cost [ms]	CPU/GPU [times]
Direct(GPU)	19.8	784.3
FFT-based(GPU)	14.2	1093.7
Hybrid(GPU)	12.2	1273.0
Direct(CPU)	15530	

7 結論

本研究ではまず直接相互相関 PIV と FFT 相互相関 PIV をそれぞれシングル GPU で高速化する手法について検討し，実装を踏まえた評価により効果的と考えられる実装方法を示した．その検討を踏まえた再帰的相互相関 PIV を開発した結果，処理速度はシングル CPU と比較して大幅に高速化されることを確認した．本要旨で示した流体励起振動を観測した PIV 画像の処理速度について言えば，GPU の処理は CPU の約 800 倍弱高速な処理が可能であることが確認できた．FFT 相互相関 PIV についても，直接相互相関法と同等の高速処理が可能であることを確認した．



(a) Recursive **direct** cross-correlation PIV

(b) Recursive **FFT-based** cross-correlation PIV

(c) Recursive **Hybrid** cross-correlation PIV

Figure 5: PIV result

次に本研究では，マルチ GPU 並列計算による相互相関 PIV の高速化手法について検討評価をおこなった．時系列画像データに対する複数 PIV 処理に対しマルチ GPU を適用することでシングル GPU と比較してさらに高速なシステムが構築可能であることが確認できたものの，その効果は限定的であり，シングル GPU での処理が既に高速化処理を更に高速化する手段としては，マルチ GPU はあまり有効ではないことを明らかにした．

さらに本研究では，GPU に実装した直接相互相関法と FFT 相互相関法との処理速度の比較をおこなった．一般により計算コストが高いとされる直接相互相関法であっても，設定する探査領域領域サイズによっては FFT 相互相関 PIV 以上の高速処理が可能なケースが存在することを明らかにし，この条件を満たす最大の探査領域を臨界探査領域と定義し明確化した．そしてこの結果をふまえたうえで本研究では直接法と FFT 法を併用するハイブリッド相互相関 PIV を GPU に実装し，その評価をおこなった．ハイブリッド法は直接法と同等の精度を有し，臨界探査領域よりも小さな探査領域を設定する限り直接法・FFT 法どちらの手法よりも高速に処理が可能である．実画像の解析を通じて，実際にハイブリット法が直接法と同等にロバストであり，直接法・FFT 法どちらよりも高速に処理が可能であることを確認した．

参考文献

[1] Douglas P. Hart. Super-resolution piv by recursive local-correlation. *Journal of Visualization*, Vol. 10, , 1999.

[2] Oliver Pust. Piv: Direct cross-correlation compared with fft-based cross-correlation. In *the 10th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, 2000.

[3] 堀田周作. 東京大学大学院修士論文. 2011.