

FPGAによるブレ補正機構に関する研究

A study of an FPGA-based video stabilizing system

丸山 裕士† 山口 佳樹‡ 児玉 祐悦†
 Hiroshi MARUYAMA Yoshiki YAMAGUCHI Yuetsu KODAMA

1. はじめに

小型機器を用いた動画の撮影には、撮影時のブレによって映像の品質が劣化してしまう問題がある。現在、それらのブレを補正するための方法は、光学式補正と電子式補正の2つに大別される。

光学式補正は、発生したブレを加速度センサ等によって検出し、機械的機構を用いてレンズの光軸を物理的に移動させる方法である。これまでに、インナーレンズシフト方式 [日下, 土田, 下島, 2002] や可変長角プリズム方式 [佐藤, 石塚, 二神, 佐藤, 1993] などによる補正が提案されている。これらは静止画の撮影に対して高品質な補正が提供できる一方、その機械的機構に多くの部品を必要とするため、小型機器への応用が難しい。

これに対して、電子式補正は撮像素子から入力される映像信号よりフレームの移動量を計算し、発生したブレを打ち消す方向に画面の表示位置を移動させることで補正を行う方法である。これらには周波数フィルタリングに基づく方法 [Fergus, Singh, Hertzmann, Roweis, Freeman, 2006] [Litvin, Konrad, Karl, 2003] や、移動体追跡を応用した方法 [Hu, Tan, Wang, Maybank, 2004] [Yilmaz, Javed, Shah, 2006] などが提案されている。実行には機械的な機構が不要であるため、撮像システムの小型化や高信頼化が期待できる一方、補正処理の実行には大量の演算が必要になる。このため、それらを実行するためのシステムには高い演算性能が求められるが、近年のプロセッサの性能向上によってこれらは組み込み機器向けの小型プロセッサでも達成されるようになった [MovieSolid 4.0, 2013]。

しかし、汎用プロセッサでの演算による補正処理の実行は、フレームのデータを一旦システム内部に保持し、それに対して大量のアクセスを行う事によって行われる。このため、補正後の画面を出力するまでのレイテンシが増加してしまう。また、今後の撮像素子の高精細化、あるいはフレームレートの向上などにより、単位時間あたりにアクセスが必要なデータの量が増加すると、メモリのスループットがボトルネックとなることが予想される。つまり、これらの方法には 4K や 8K といった次世代の映像規格をサポートする撮像素子に対して拡張性が乏しい欠点があった。

本稿では撮像システムの小型化と高信頼化の観点から、小型機器への応用を想定した電子式ブレ補正システムを提案する。これらのシステムは FPGA を用いて一つの集積回路として実装を行った。処理の実行の際に用いる映像信号として、カメラから送信される画素のストリーミングデータのみを使用する事で、フレーム全体へのアクセスをなくすことができ、上述の電子式補正の欠点を克服した。また、補正処理において、各フレームの移動量を求める際の精度向上手法について提案し、それらの実装と評価を行った。

2. オプティカルフロー検出とブレ補正手法

映像中に存在する物体を検出し、その移動ベクトルを求める処理のことをオプティカルフロー検出と呼ぶ [Fleet Weiss, 2005]。ブレ補正の実現にはこのオプティカルフロー検出が要求される。この理由は、[1] 目標物(被写体)の移動によるブレと、[2] 撮像素子(カメラ)の移動によるブレの大きく二つが実世界で想定されるが、いずれも目標物と撮像素子の相対関係(幾何学的制約)によりその運動を推定できるからである。

ここで、オプティカルフロー検出と本研究で対象としているブレ補正との違いについて考える。図1にフレーム内オプティカルフローとブレ補正の違いを示す。

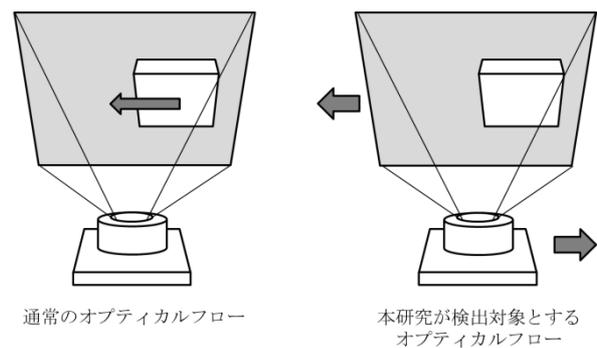


図1 フレーム内オプティカルフローとブレ補正の違い

二つの違いは、フレーム内の一部の画素を目標物とするか(図1左)、フレーム全体を目標物とするか(図1右)である。フレーム内の一部を目標物とする場合、例えば、フレームそのものを絶対座標系として定義し移動量を検出することが可能だが、フレーム全体が移動物となる場合は同様に処理することはできない。そこで本論文では、システム上に別の座標を設定し、その座標上のフレーム移動量を検出することで、ブレ補正を実現している。

2.1 フレーム移動量の検出手法

ブレ補正を行うためにはまずフレーム移動量を算出する必要があるが、移動量計算には時間的に連続した2枚以上のフレームが必要となる。ここで、実システムへの応用を考えると、4K や 8K といった次世代において計算資源を抑えつつも十分な推定を行える方法が望まれる。そこで本研究では、フレームの画素全てを使用するのではなく、システム内部に保存するデータ量を最小限にするためにフレーム内から複数の部分空間を切り出し、それらの部分演算を統合的に評価することによって高精度のブレ補正を実現した。図2にこの概要を示す。

†筑波大学大学院 システム情報工学研究科

‡筑波大学計算科学研究センター

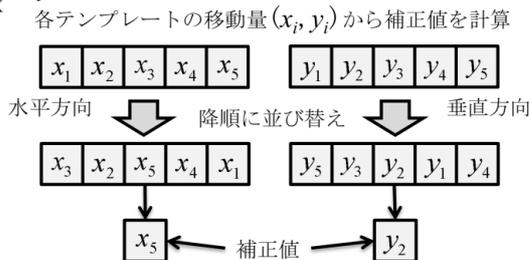
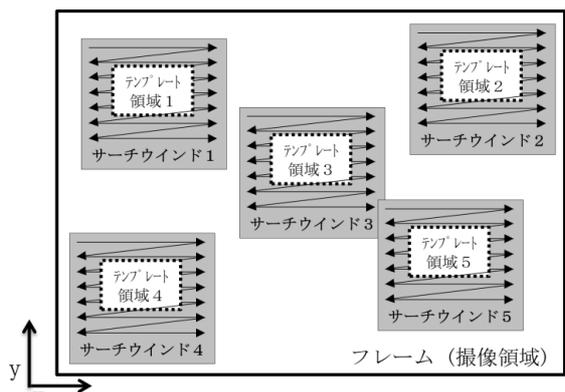


図 2 フレーム移動量の推定方法

図 2 において、フレームより切り出された一定サイズの領域をテンプレート領域、フレーム移動量検出のためにテンプレートを比較させる範囲をサーチウインドと呼ぶ。これらの領域は相互に重なっていても構わない。この手法を導入することで一時記憶を大幅に削減できる。また、各テンプレートは独立なので並列計算を導入できる。

次に、テンプレートの移動量からフレーム全体のオプティカルフローを算出する。ここで、例えば交差点における車の移動など、テンプレートとして指定された領域がカメラのブレとは独立に移動量を持つことが考えられる。つまり、計測されたテンプレート移動量はブレによって発生する移動量とテンプレート内の目標物自体の移動量の 2 種類存在する。後者を無視してフレームのブレを特定するために、本論文では複数のテンプレート領域を設定し、それぞれの領域で個別に得られた移動量の中央値をフレーム全体の移動量として推定することにした。

図 2 を例に説明すると、初めに五つのテンプレートとそれぞれのサーチウインドから各テンプレートの移動量を計算する。次に、これらの移動量を水平成分および垂直成分のそれぞれで並び替えて中央値を決定し、それをフレーム移動量として採用する。しかし、この方法のみで推定は難しく、例えばフレーム内の円運動（風車の撮影）などでは、各テンプレートの移動量が大きく異なり中央値が代表値として適切ではない場合も考えられる。そこで、テンプレートの移動量に関して統計的処理を適用し、一定数のテンプレート移動量が中央値付近に存在しない場合、オプティカルフロー検出が失敗した、と考え中央値を棄却する(補正しない)ようにした。

2.2 テンプレート移動量の演算方法

本節では、2.1 節で述べたテンプレート移動量の演算方法について説明する [Schweitzer, Bell, Wu, 2002] [Lewis, 1995]. 図 2 において、時間(t-1)で読み込まれた各テンプレートは、時間(t)で各テンプレートの周辺に定義されるサーチウインド内で比較処理を行い、その移動量が算出される。ここで、時間(t-1)のテンプレート情報を $T_{i,j}$ 、時間(t)のサー

チウインド内のフレームの一部を $I_{i,j}$ 、それぞれを水平 M 画素及び垂直 N 画素の領域と仮定すると、

$$\phi_{i,j} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} T_{i,j}}{NM}$$

$$\psi_{i,j} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} I_{i,j}}{NM}$$

として、各ベクトルを零平均化し、

$$R_c = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \phi_{i,j} \psi_{i,j}}{\sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \phi_{i,j} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \psi_{i,j}}}$$

で示される正規化相互相関 R_c を最大にするサーチウインドの部分を見つけるのが一般的である [J., 1995] [Schweitzer, Bell, Wu, 2002]. しかし、この方式には乗算や逆平方根演算が含まれ、ハードウェアでの直接計算に向いているとは言いにくい。

そこで本研究では、時間的に連続しているフレーム間の明るさの変化は相関の変化に対して十分小さいと仮定し、 R_c が最大となる箇所を求める代わりに残差 R が最小になる箇所を検出対象とした。

$$R = \|T - I\|_1 = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |T_{i,j} - I_{i,j}|$$

2.3 移動量検出における精度の向上について

残差 R を各画素の差分により求める場合、検出可能な最小移動量は 1 画素である。そこで本研究では、最小検出幅が 1 画素である移動量検出により補正システムの実装と評価を行った。このとき実環境では 1 画素未満の幅でブレが生じていたため、フレームの移動方向は正しく検出できたものの、細かい誤差が蓄積し長時間補正を続けると正しく補正ができないことがわかった。そこで、1 画素未満のブレに対しても正確に補正を続けるために、取り扱う画素情報を仮想的に増やすことを試みた。図 3 にこの概要を示す。

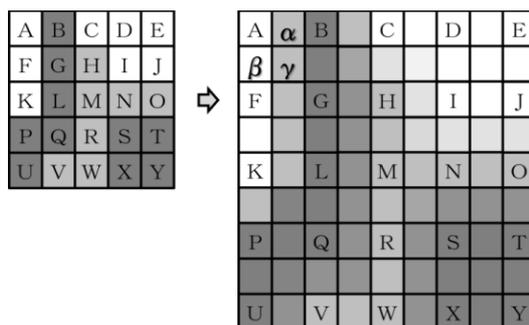


図 3 画素間の値の補間 (左: 補間前 右: 補間後)

図 3 に示すように、本研究では、移動量の精度向上のため 0.5 画素単位でのオプティカルフローを検出する処理を実装した。これは、比較対象となっている領域について、画素間の値を近傍画素の平均値によって補間し、新たな仮想画素を生成することで実現している。図 3 を例に示すと α 、 β 、 γ 、の各画素値が $\alpha=(A+B)/2$ 、 $\beta=(A+F)/2$ 、 $\gamma=(A+B+F+G)/4$ 、で計算される。比較に必要な演算量が 1 画素単位の補正に対して 4 倍となってしまうが、この補間処理によって 0.5 画素単位での移動が検出可能になり、移動量検出の精度を向上させることが可能となる。

2.4 映像信号のノイズ軽減手法

撮像素子を用いて動画の撮影を行う際に問題となるのが、撮像素子からのノイズである。これに対しては、フレーム内の微小な領域について画素の値を平均化することで、ノイズ成分が多く含まれる信号の高周波成分を除去するフィルタを実現した。本研究では、この処理をモザイク化と呼ぶ。モザイク化データ M_{ij} は、モザイク化に必要な近傍画素の数を n^2 とすると、

$$M_{ij} = \frac{\sum_{k=i}^{i+n-1} \sum_{l=j}^{j+n-1} I_{kl}}{n^2}$$

として与えられる。 $n=2$ の際の例を図4に示す。

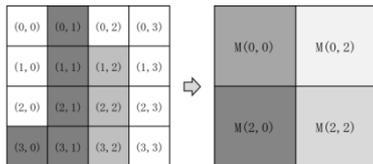


図4 モザイク化処理 (n=2)

この処理の実行によって、ノイズ除去フィルタの効果以外にも、類似領域探索の際に差分を求める画素数の削減が可能であり、これは回路規模の縮小にも貢献している。

しかし、この際にフレームを単純にモザイク化するだけではフレームの解像度が低下してしまい、現フレームで入力されたモザイクが前フレームのモザイクの境界を跨ぐようにフレームの移動が発生した場合に、正しく移動を補正できない問題がある。このため、2.3節で補間した画素を利用して0.5画素ずつずらした領域に対してもモザイク化処理を実行することで、この問題を回避した。この様子を図5に示す。

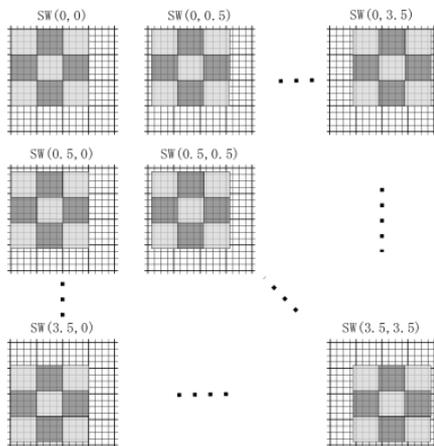


図5 モザイク化サーチウインド(SW)の例 (n=4)

3. システムの実現方法

補正を行う回路全体の構成を図6に示す。補正值算出には、撮像素子から入力される8bitのグレースケールの輝度情報を使用する。また、算出処理は全てパイプライン化され、画素の受信中に常時動作する。このため1フレーム分の画素の情報がシステムに対して入力され終わる際には、前フレームに対する現フレームの移動量の計算が完了し、補正值が算出されている。本章では、これらの処理を実現するための回路の構成法について述べる。なお、本システ

ムで入力される画素情報はプログレッシブ方式で入力されるものとする。

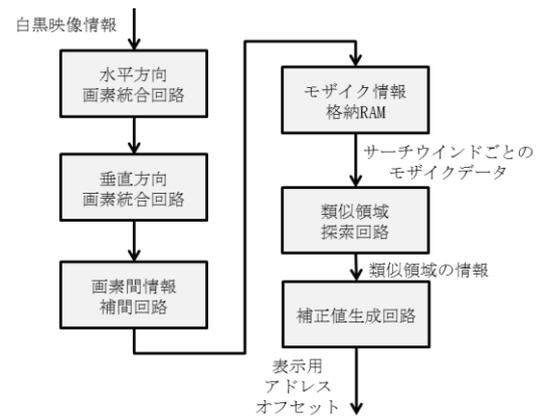


図6 補正值算出回路の全体構成

3.1 画素統合回路

フレームのモザイク化を実行するためには、水平方向と垂直方向に画素の輝度を合計する必要がある。

水平方向の画素の合計は、シフトレジスタを用いて、連続して入力される画素の情報を保存し、それらの値を足し合わせることで求める。この構造を図7に示す。

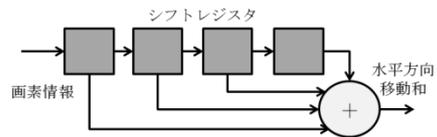


図7 水平方向画素統合回路

これに対して、垂直方向に隣り合う画素の情報はシステムに対して連続に入力される訳ではない。このため、画素情報は回路内のRAM領域に一旦保持し、処理が必要になり次第レジスタへと読み出す必要がある。この保持のため、回路は水平1ライン分のバッファを、モザイクの辺の長さに応じて保持する必要がある。この構造を図8に示す。

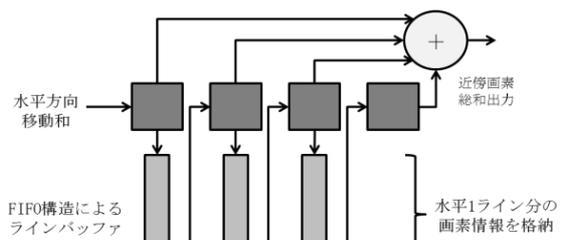


図8 垂直方向画素統合回路

3.2 画素間情報補間回路

0.5画素単位の補正の実現のため、画素間の値の補間を行う。補間は隣り合う画素の合計同士の平均を用いるため、水平方向と垂直方向及び、斜め下方向の画素の合計の情報を同時にレジスタ上に呼び出し、それぞれに平均を求める。この時、画素統合の際と同様に水平方向に連続する情報は、時間に連続して生成されるが、垂直方向に隣り合う情報をレジスタに保持するためには、水平方向に1ライン分のバッファが必要になる。この構造を図9に示す。

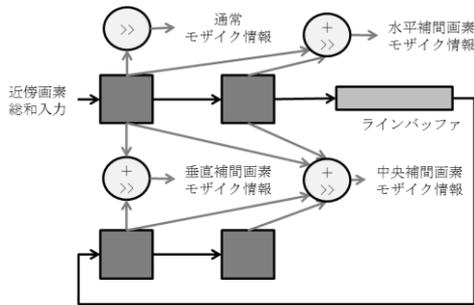


図 9 画素間情報補間回路

3.3 モザイク情報格納 RAM

生成されたサーチウインド内のモザイク情報は、演算時に FPGA 上の内部 RAM に格納する必要がある。これは、サーチウインド内のデータを高効率に読み出すことで、後段の類似領域探索回路でパイプラインストールを発生させないようにするためである。以上より演算精度と効率が向上する一方、入力情報と水平・垂直・斜め方向の補間情報を保存する内部メモリおよびその 4 倍の演算回路サイズがボトルネックとなる。そこで、RAM 内に保持される画素情報を 4 分の 1 ずつ後段に出力することで、処理の実時間性能を保ちつつ、より小規模な回路で探索処理を可能にしている。図 10 にこの構造を示す。

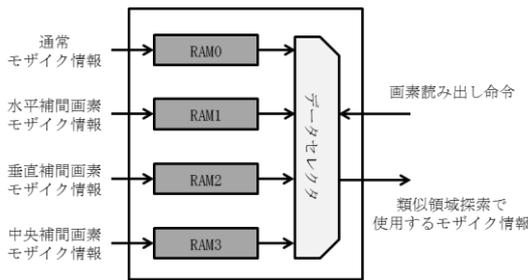


図 10 モザイク情報格納 RAM

この RAM モジュールはサーチウインドの個数だけ存在し、それぞれに一つのサーチウインド内の全てのモザイク情報を格納している。

3.4 類似領域探索回路

2.2 節で述べた残差 R を求めて、最も類似度の高い領域を出力する回路である。これらの実行には、シフトレジスタとラインバッファで構成される図 11 のような構造を利用する。

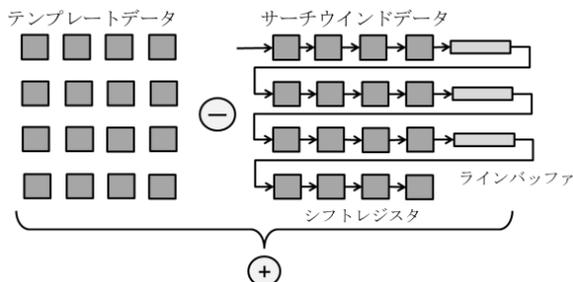


図 11 類似領域探索のための残差の計算の構造

まず、処理の開始前にテンプレートデータをレジスタ上に用意しておく、次に、サーチウインドのモザイク情報を 1 モザイクずつシフトレジスタに対して入力していく。こ

の時、ラインバッファは図 8 の垂直方向統合回路と同じ要領で、垂直方向のモザイク情報の整列のために使用される。最初に入力されたモザイク情報が、最終ラインにおける最後のシフトレジスタに到達した時、シフトレジスタ中には、まさにサーチウインドを切り出した部分が格納されており、テンプレート領域に対する残差の計算が可能になる。

また、この後に求められた各レジスタの差から、絶対値を求める処理と、それらを合計する処理をパイプライン化することで、実質的に 1 クロックで一回の残差の計算を完了することが可能である。

3.5 補正值生成回路

5 箇所の類似領域の探索結果から、補正值としての表示アドレスのオフセットを出力するための回路である。この回路では、まず水平及び垂直方向についてそれぞれの移動量の中央値を求める。これらは、一旦値を降順に並び替えた後、第 3 番目の値を出力する回路である。また、中央値周辺の値の分散を計算し、算出した値がフレーム全体の移動として妥当かどうかを確認し、その結果を出力する。ここで得られた値は、前フレームと現フレームの間の移動量なので、現在の画面の表示位置にこれらの移動量を加算することで、実際にフレームの読み出しの際に使用される表示アドレスのオフセットを計算する。

4. 撮像システムの構成

本研究において作成した撮像システムの全体構成を図 12 に示す。

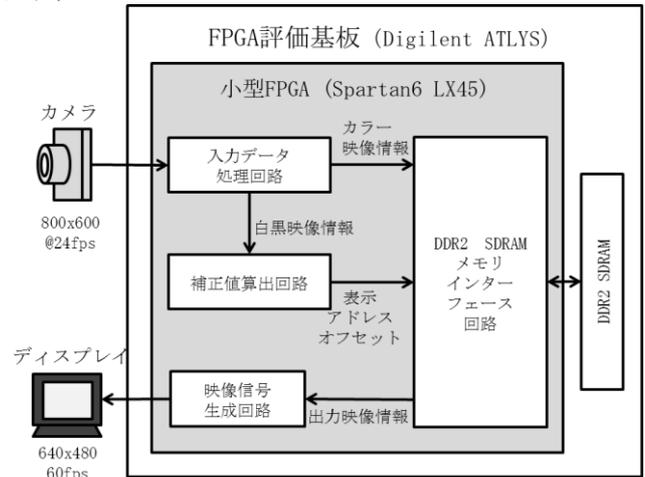


図 12 撮像システムの全体構成

システムに対する映像信号の入力には、Digilent 社の VmodCAM を使用した。これは最大 UXGA(水平解像度 1600[pix], 垂直解像度 1200[pix])で 15fps での撮影が可能な撮像素子を搭載しているカメラモジュールである。

システムは Digilent 社の Atlys ボードに実装され、使用された FPGA は Xilinx 社の XC6SLX45 (Digilent 社の Atlys ボードに実装) である。比較的小型のデバイスであるが、著者らは提案する方式に関し、SVGA(水平解像度 800[pix], 垂直解像度 600[pix])であれば 120[fps], SXGA (水平解像度 1280[pix], 垂直解像度 1024[pix])であれば 60[fps]などでリアルタイムに補正できることを確認している。但し、本論文では測定系(撮像素子および定量的評価ソフトウェア)の制約から SVGA の 24fps で定量的評価を行った。また、

評価における画素伝送のクロック周波数は 80MHz とし、確認用の映像出力には HDMI 出力を使用している。

回路の開発環境としては、回路記述には VerilogHDL を使用し、論理合成ツール及びシミュレーションツールとして、Xilinx 社提供の ISE Design Suite 13.3 を使用した。また、映像出力のためのフレームバッファとして、ATLYS 上に搭載されている Micron 社の DDR2 SDRAM を使用した。このメモリのインターフェイスについては Xilinx 社の Memory Interface Generator で生成される Memory Controller Block を使用している。表 1 に 1.0 画素精度補正および 0.5 画素精度補正の回路規模を示す。

表 1 回路使用量の比較 (XILINX 社 XC6SLX45 使用時)

補正方式	LUT 数		内部 RAM(9kb)数	
	使用個数	使用率(%)	使用個数	使用率(%)
1.0 補正	7866	28.8	39	16.8
0.5 補正	8990	32.9	116	50.0

1 画素単位の補正精度で補正処理を行った報告は既にあるが [Tohru, Yoshiki, Yuetsu, 2013], 本論文では同一手法での実装および同一環境で評価するために何れもスクラッチから設計した。0.5 画素精度補正は 1.0 画素精度補正の約 4 倍のデータ量を取り扱うため内部 RAM の使用量が増加する。この使用量の削減は手法的に難しいが、演算回路についてはラインバッファ等を高効率に利用することで、高々 15% 程度の増加に留めることができた。

4. 評価と検証

本論文の提案およびその実装を検証するため、得られた補正值について定量的な評価を行った。この際、本来であればカメラモジュールを直接振動させることで評価を行うことが望ましい、また、これを行う方法としては手動によりカメラモジュールに直接振動を与える方法が考えられるが、この方法では振動の量を定量的に測定することが困難であり、システムの検証には使用不可能である。

そこで本研究においては、動作の定量的な評価のため、カメラ側を固定した状態でフレームの移動量があらかじめ測定されている映像をカメラに入力し、それに対して補正を行った。また、得られた補正值に対して、それらをブレの理論値と比較することによりシステムの評価を行った。

4.1 評価手法

評価の際に用いた装置の配置を図 13 に示す。ディスプレイに表示された水平解像度 800[pix]、垂直解像度 600[pix]の画像が、使用する VmodCAM の撮像範囲全体に対応するように配置等を調整した。本環境では、ディスプレイより 235mm 離れた位置にカメラを配置している。

この時、テンプレートは画面内に 5 箇所設定し、それぞれは画面の中心及び、画面の中心から撮像範囲の頂点に向けて引いた線分のおよそ中心となる位置に配置した。モザイクは正方形の形状で、一辺は 4[pix]とした。テンプレートも同様に正方形とし、一辺が 8[モザイク](32[pix]相当)の領域とした。また、サーチウインドはテンプレートの移動を ± 18 [pix]で検出するために、一辺が 69[pix]の正方領域とした。

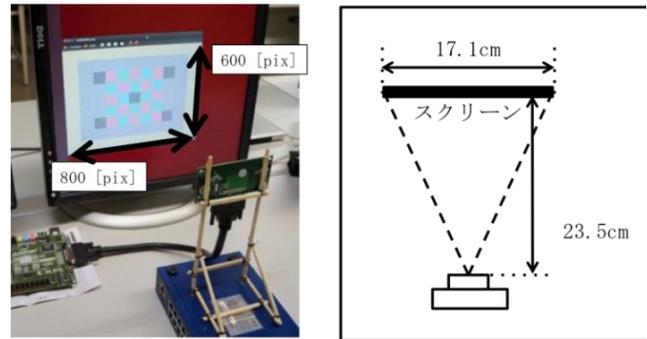


図 13 評価時のカメラとディスプレイの配置

ディスプレイに表示される映像は著者らが作成したソフトウェアによりブレ幅や速度を設定できるようになっている。よって、この計測系の流れは、表示された映像を VmodCAM で撮影し、FPGA はその映像からテンプレートマッチングを通し二次元のベクトルで表される補正值を求める。著者らはこの補正值を PC に適宜フィードバックし、ソフトウェアの設定ブレ幅と FPGA の補正值を比較しアルゴリズムの有意性について検証を行った。

4.2 入力した映像情報

システムに入力する映像には、図 14 に示す 14 種類の自然画像を垂直方向と水平方向のうちの一方方向について、同じ速さで周期的に振動させたものを用いた。画像には通常の風景画像の他にもタイルや芝生等のような一定のパターンを持ったものを使用した。

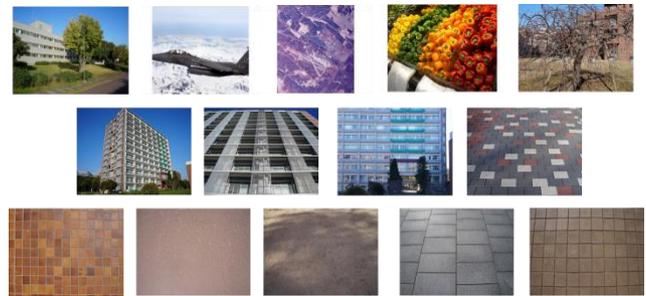


図 14 移動を与えた画像の例 (14 種)

フレームの移動速度については、1 フレーム間で 9.5 画素のものと、13.5 画素のもの(図 15)を作成し、これらを 14 秒間移動させた。この評価を垂直及び水平方向への移動の双方に対して行う事で、事実上は全ての方向に対して補正が実現できることを示すものとした。

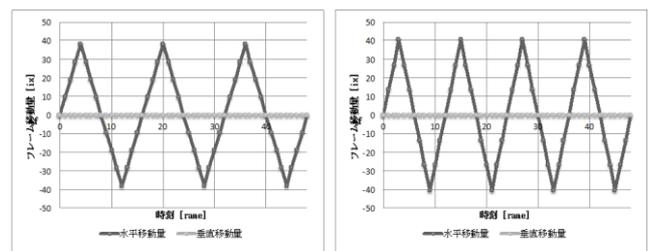


図 15 水平方向に与えたブレの理想移動量 (左: 9.5 [pix/frame] 右: 13.5 [pix/frame])

4.3 評価結果

評価の実行結果について、図16に示す。

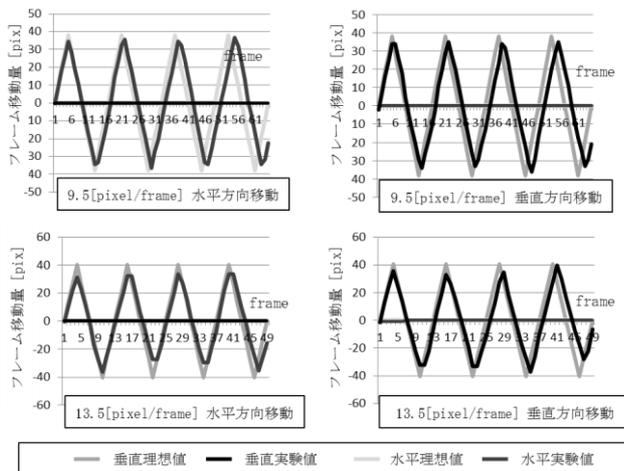


図16 短時間(50フレーム)における補正結果
(0.5画素精度および1.0画素精度)

これにより、フレーム間のブレが9.5画素及び13.5画素の場合について、短時間(例えば静止画に近いまたは携帯電話の動画など)であれば、0.5画素精度/1.0画素精度に関わらずブレを良く補正できたと結論づけることができる。

また、このブレ補正による小型携帯機器での信頼性の向上効果について検証を行った。解析対象映像は筑波大学構内にてSony HDR-TD20を自転車で固定して十分に走行して作成された。この時のブレ量の解析結果を表2に示す。

表2 筑波大学構内走行時のブレ量(19451frames @24fps)

	絶対値平均	標準偏差
水平方向	6.99	7.15
垂直方向	7.51	8.53

以上から水平・垂直方向に関して0.5画素精度により、実応用においても十分な補正範囲を提供出来ると考える。

次に、より長い時間(300フレーム超)における誤差累積について図17に示す。

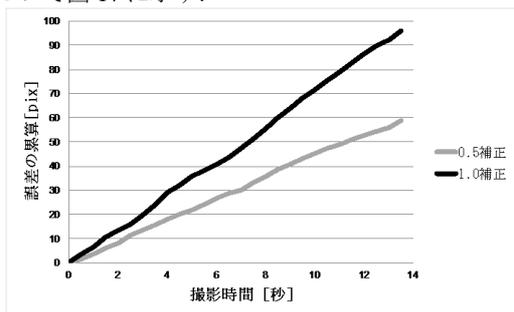


図17 撮影時間に対する誤差累積の比較

図17において、0.5画素精度の補正に比べて1.0画素精度補正において誤差が大きいことが分かる。これは、1.0画素未満の誤差が頻繁に生ずるケース(小規模な振動が継続してするような場合)では無視できず例えば車載システムなどでは大変大きな問題となる。図17の実験においては、0.5画素精度の導入により、14秒で37画素(誤差全体の約4割)の誤差を削減できた。一時記憶については手法上削減が困難だが、演算回路の増加とのトレードオフを考

えると0.5画素精度は実システムへの応用において有効であると言える。

5. まとめ

水平800[pix]垂直600[pix]の解像度を持ち、24[frame/s]のフレームレートをもつ動画像に対して、電子式ブレ補正を実行するシステムをFPGA上に構成することで、これらが一つの集積回路として実現できることを示した。また補正精度向上手法として、0.5画素単位のブレに対する補正処理の実装と評価を行い、この処理によって誤差の蓄積が緩和される様子が確認できた。なお、本研究は特定の対象物を追尾するのではなく、フレーム全体の移動量の補正を提案するものであり、画像の拡大や0.5画素未満の微小なブレをわざと補正しない(あそびを設ける)等、実応用を考慮した設計がなされている。1画素未満の変位を正確に補正する方法として、[Shimizu, 2005]なども提案されているが、以上の理由から本研究の目的と合致しないため、採用しなかった。

これらのシステムは全て撮像素子からの画素のストリーミングデータのみを解析することによって実行されており、システムは撮像素子に同期して動作可能である。

参考文献

- FergusRob et al. (2006). Removing camera shake from a single photograph. ACM Transactions on Graphics, 25(3), 787-794.
- FleetJ.David and WeissYair. (2005). Handbook of Mathematical Models in Computer Vision (chapter 15). (ParagiosNikos, ChenYunmei, FaugerasD.Olivier, 共同編集) Springer.
- HuWeiming et al. (2004). A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man, and Cybernetics (Part C: Applications and Reviews), 34(3), 334-352.
- J.P.Lewis. (1995). Fast Template Matching. Vision interface, 10 (1), ページ: 120-123.
- LewisP.J. (1995). Fast Template Matching. Vision interface, 95(120123), 15-19.
- LitvinAndrew et al. (2003). Probabilistic video stabilization using kalman filtering andmosaicking. Electronic Imaging, Image and Video Communications, (ページ: 663-674).
- MovieSolid 4.0. (2013年2月22日). (Morpho, Inc.) 参照先: <http://www.morphoinc.com/>
- SchweitzerHaim et al. (2002). Very Fast Template Matching. the 7th European Conference on Computer Vision-Part IV.
- TohruYabuki et al. (2013). Real-time Video Stabilization on an FPGA. Smart Structures and Systems, (ページ: 114-119).
- YilmazAlper et al.(2006). Object tracking: A survey. ACM Computing Surveys (CSUR), 38(4), Article 13 (1-45).
- 佐藤満ら. (1993). 光学式手振れ補正システム. テレビジョン学会技術報告, 17(5), 15-20.
- 日下博也ら. (2002). 光学式手振れ補正のための制御システム開発. 映像情報メディア学会誌, 56(3), 411-416.
- M.Shimizu and M. Okutomi, "Sub-pixel estimation error cancellation on area-based matching", International Journal of Computer Vision, Vol.63, No.3, pp.207-224 (2005).