Vol.2022-ARC-248 No.34
Vol.2022-SLDM-198 No.34
Vol.2022-EMB-59 No.34
2022/3/10

多入力画像解析システムにおけるリマップ処理のメモリ帯域低減

一場 利幸[1,a)]　貞末 多聞[2]　吉川 隆英[1]

概要：カメラの普及により，画像解析の重要性が増している．我々は，多入力の画像を一拠点に集約して解析することを想定し，ネットワーク処理・デコード・画像処理の全てを FPGA で実行するシステムの開発を行っている．FPGA で実行する画像処理が外部メモリを使う場合は，メモリ帯域が性能のボトルネックになる可能性がある．本論文では，多入力の画像についてリマップ処理を行う際に，リマップ処理で利用するパラメータについて画像間で共有するベース値と各画像の個体差を表現する差分値に分けることで，メモリ帯域を低減する手法を提案する．提案手法の効果を見積もった結果，メモリ帯域を最大 50%低減できることがわかった．

1. はじめに

近年，IoT 機器や自動車などにカメラが搭載されることが増えてきており，その画像を解析して，例えば道路状況を動的に地図に反映するダイナミック・マップや，建屋内の人流可視化などに基づいた危険回避誘導，安全予防検知などのサービスが提供され始めてきている．そして，これらのサービス提供には複数のカメラからの画像を集約して，一括で AI などによる画像認識を行う必要がある．また，実用的なサービスを提供するためには，画像認識を含む画像の解析処理にはリアルタイム性が求められると考えられる．

複数のカメラからの入力画像は，レンズの歪み，解像度，角度，明るさなどがカメラや時間によって変化する．レンズの歪みなどによって，画像認識の精度に影響がある [1] ため，状況に応じて様々な画像処理を行う必要がある．CPU のみで画像処理を実行すると，遅延によるフレーム落ちなどが発生し，リアルタイム性を損なうおそれがある．そのためアクセラレータが必要であり，また，アクセラレータが画像処理を行うことでサーバ 1 台で処理可能な画像数を増やすことができる．多数の画像を処理するアクセラレータの例として，移動物体の抽出と追跡をカメラ 64 台の画像に対し実現する FPGA アクセラレータが提案されている [2]．しかし，カメラ毎に存在する個体差や入力画像の性質の変化（ある箇所で照明が消えたなど）に対応するための画像処理を FPGA で実現していない．このようなシステムでは，CPU や GPU で画像処理を行うために，FPGA と CPU/GPU 間でデータ転送を行う必要がある．デバイス間のデータ転送も，リアルタイム性を損なう要因である．

リアルタイム性を持つ画像解析を実現するため，我々は，複数の画像についてネットワークから受信する処理，デコード処理，画像処理の全てを FPGA で実行するシステムの開発を行っている．画像処理は，入力の変化に柔軟に対応出来るよう動的に設定パラメータを調整可能である．画像処理を FPGA で実現するにあたって，FPGA 内のリソース量と FPGA 外のメモリ帯域が制約になる．文献 [3] によると 2008 年から 2017 年の期間で，FPGA 内のリソースの 1 つである DSP は 6 倍に増えたが，メモリ帯域の増加は 2 倍未満である（図 1）．このように，FPGA に出来る限り多数の画像を入力して処理を行おうとすると，搭載可能なリソース量の制約よりも，メモリ帯域の制約の方が厳しくなってしまう傾向にある．

そこで本論文では，画像処理のなかでも重要なレンズ歪み補正を実現するリマップ処理について，FPGA 外のメモリ帯域を低減する方法を提案する．これは，画像間でリマップ処理のパラメータ共有を行うもので，類似の研究は見当たらない．

本論文の構成は次の通りである．第 2 章では，我々が開発している多入力画像解析システムの概要を紹介する．第 3 章では，レンズ歪み補正を行うリマップ処理の概要と実装について述べ，リマップ処理に必要なメモリ帯域の問題について述べる．第 4 章では，リマップ処理のメモリ帯域

[1] 富士通株式会社
Fujitsu Limited
[2] Intellectual Highway 合同会社
Intellectual Highway, LLC
[a)] t.ichiba@fujitsu.com

Vol.2022-ARC-248 No.34
Vol.2022-SLDM-198 No.34
Vol.2022-EMB-59 No.34
2022/3/10

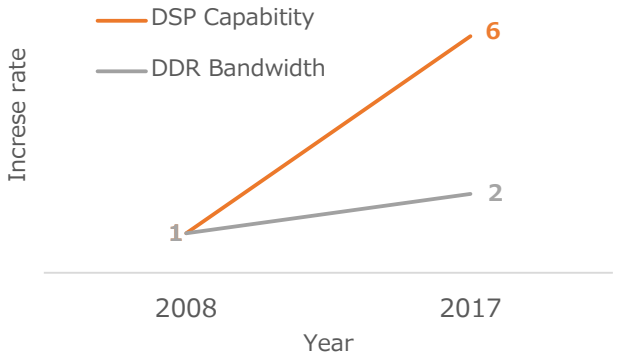

図 1 10 年間の FPGA DSP と DDR メモリ帯域の変化（文献 [3] を基に作成）

を低減する手法を提案し，メモリ帯域の見積りを述べる．そして，第 5 章でまとめる．

2. 多入力画像解析システム

2.1 要件

多入力画像解析システムへの要件を以下のように定めた．

要件 (1) 可能な限り多数の画像をサーバ 1 台に集約する

要件 (2) リアルタイム性を持つ様々な画像処理を実行できる

要件 (3) 入力画像は RTP (Real-time Transport Protocol) による転送で，H.264 で符号化されている

要件 (1) を満たすことで，次のようなメリットがある．まず，複数画像を同時に解析することによって，異常検知や行動解析などの認識に使える情報が増え，より正確な識別が可能となる．また，多入力画像解析システムは，拠点に設置したサーバで運用する想定をしている．そのため，ストレージの制約が厳しいカメラ側（エッジ側）ではなく，拠点での画像保存が実現できることから，後の詳細な解析や人間による確認が可能となるメリットがある．そして，拠点では少数のサーバで多くのカメラ画像を処理できれば，設置や運用にかかるコストを削減できる．

要件 (2) に関して，まず画像処理の必要性について述べる．AI 処理の前に画像処理を実行する理由は，以下の 2 つである．

必要性 (a) データ量削減による AI 処理の負荷低減

必要性 (b) AI 処理の認識精度向上

必要性 (a) に関しては，数十の入力（以降，画像の数を表すチャネルという単語も併用する）画像を集約して受信した時に，全てのチャネルの全てのフレームに対して AI による認識処理を行うのは，現在のコンピュータの処理性能では困難である．そのため，画像の解像度を 400 から 600 ピクセル程度に落とす解像度変換や，フレーム数を間引く処理などを事前に施すことで，AI 処理の負荷を低減する必要がある．しかし，単純な解像度変換やフレーム間引きでは，認識率低下や認識漏れを引き起こしてしまう恐れがあるため，例えば，監視カメラであれば移動する人・物体を検出したフレームや領域だけを用いるなどのデータ量を適応的に削減する工夫によって，認識率を落とさないまま AI 処理の負荷を低減することが求められる．

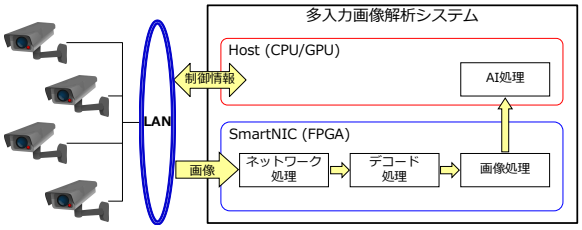

図 2 多入力画像解析システムの構成

必要性 (b) に関して，カメラ画像のノイズ，ぼけや，スキュー，回転，レンズ歪みなどの幾何変形が認識精度に影響を与え，特に幾何的な歪みは大きく認識精度を悪化させる [1] ため，これらの幾何歪みを取り除くことで，AI による認識処理の精度悪化を低減させることが出来る．

以上のことから，AI 処理の前に画像処理を実行する必要がある．

そして，画像処理の内容は多岐にわたる．画像処理は，カメラ毎に存在する個体差に応じた変化だけでなく，時間による変化，解析の条件や目的に応じた変化がある．例えば，昼と夜などの時間変化によって画像処理を変えたり，車載カメラなのか監視カメラなのかで AI による認識処理の解析内容が変化し，それに伴って画像処理も変える必要があると考えられる．さらに，画像処理にはフレーム落ちが発生しないようリアルタイム性が求められる．要件 (2) は，このような理由から定めた．

要件 (3) は，現在，標準的に用いられている通信プロトコルと符号化であることから定めた．

2.2 構成

要件 (3) に対応するネットワーク処理と H.264 のデコード処理を CPU のみで実行すると，多数の画像を処理することから，メモリ帯域が性能のボトルネックになる懸念がある．また，画像処理も必要であることから，CPU のみで多入力画像解析システムの処理を全て実行するのは現実的でない．

我々は，図 2 のように，ネットワーク処理・デコード処理・画像処理の全てを FPGA で実行することで，前述の要件を満たせると考え，開発を行った．ネットワーク処理は Intellectual Highway 社の PTU (Protocol Termina-

Vol.2022-ARC-248 No.34
Vol.2022-SLDM-198 No.34
Vol.2022-EMB-59 No.34
2022/3/10

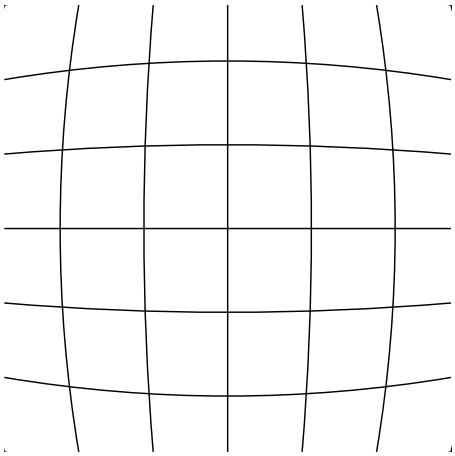

図 3 レンズ歪み（樽型）の例

tion Unit) を用いており，デコード処理は System-on-Chip Technologies 社の H.264 AVC Video Decoder IP Core を用いている．画像処理は，要件 (2) について述べたように，様々な処理を実行できる必要があり，この実装が容易にできるよう OpenCL での記述が可能である．FPGA によるこれらの処理後の画像に対して，CPU/GPU による画像認識などの AI 処理を行う．

3. リマップによるレンズ歪み補正

ここではまず，レンズ歪みの補正と，補正を実現するリマップ処理について述べる．次に，リマップ処理を含めた多入力画像解析システムの実装結果を述べ，レンズ歪みによる認識率への影響を述べる．そして，メモリ帯域の問題を述べる．

3.1 レンズ歪み補正の必要性

本論文で扱うレンズ歪みは，広角レンズで発生するもので，格子状の画像が図 3 のように樽型に歪む．

レンズ歪みの補正を実現する手段として，半径方向歪み (radial distortion) と円周方向歪み (tangential distortion) のパラメータによるレンズ歪みモデルを利用した補正手法と，画素の座標を変換するリマップ処理による補正手法が存在する．

リマップ処理は，画素毎の座標テーブルをもつリマップテーブルを変更することにより，1 つの回路で一般的な幾何変形に対応できる．そのため，本論文では，リマップ処理を用いて歪み補正を実現する．

レンズ歪みを補正するためのキャリブレーションには，Zhang の手法 [4] が広く使われている．キャリブレーションでは，平面に直線上にならぶ格子やドットなどをカメラで撮影することで，個々のカメラのキャリブレーションを行うことができる．歪みは画像の端にいくほど大きくなる．レンズ歪みの個体差について，文献 [5] では，3 台の Kinect でキャリブレーションを行い補正パラメータを測

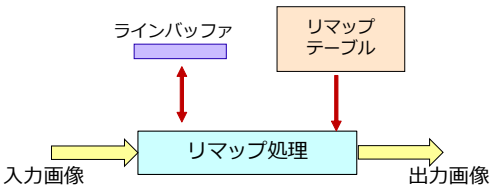

図 4 リマップ処理の概要

定した結果，640 × 480 の解像度で最大 4 ピクセル程度のズレが生じることが報告されている．

3.2 リマップ処理の流れ

リマップ処理の実装には，ラインバッファとリマップテーブルを用いる（図 4）．ラインバッファは，リマップに必要な入力画像の一部を保存するものであり，画像の 1 ラインが大きさの単位となる．リマップテーブルは，入力画素の座標を出力画素のどの座標に変換するのかを対応づける．ラインバッファに一時的に保存されている画素値と，リマップテーブルの座標値から出力画素値を求めることでリマップ処理が行われる．

リマップテーブルは，単純には入力画像と同じサイズの座標値が必要となる．例えば，入力画像サイズが 640×480 であれば，X 軸 Y 軸の座標値が必要なので，$640 \times 480 \times 2$ 個の座標値を保持するリマップテーブルとなる．このように，リマップテーブルはデータ量が大きいため，FPGA 内の SRAM に置くことはできなく，外部の DRAM に置く必要がある．多入力の画像を扱うためには，リマップテーブルを読み出す際のメモリ帯域が問題になる．メモリ帯域を低減するため，リマップテーブルの値を 1/2 や 1/4 に間引いて，座標値がない部分は近傍の座標値から補間計算により得る方法が考えられる．ただし，補間の計算方法によっては，リマップ処理の結果が変化してしまう恐れがある．

3.3 実装結果

多入力画像解析システムを実装した結果について述べる．入力するカメラ画像の解像度は 1280 × 720 とし， Intel FPGA PAC D5005（以降，D5005 と記す）を対象として合成を行う．

FPGA 処理のうち，画像処理ブロックは，図 5 のようになっており，OpenCL で記述された各画像処理ブロックの処理を経て，最終結果のみが DRAM に出力される．すなわち，H.264 デコーダの出力は，YCbCr から RGB への色変換ブロックに渡り，その後段に，フレームレートの調整が必要となった際のフレームレート間引きブロック，リマップブロック，解像度変換ブロック，そして DRAM へ書き込みを行う出力ブロックの順に接続されている．この連結された処理ブロックを 1 つのパイプラインと呼び，このパイプライン 1 つで 7 チャネルの動画に対する処理を行

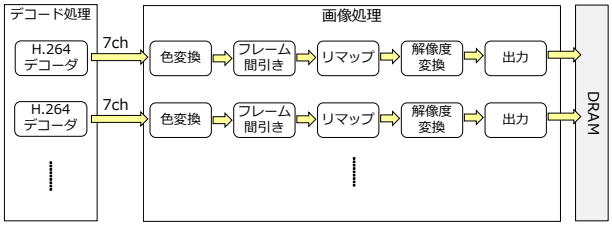

図 5 画像処理の実装内容

う． D5005 では，5 つのパイプラインを実装したため，合計 35 チャネルの動画を同時に処理できる．

実装したパイプラインについて，各ブロックの遅延は 1 ライン分の遅延に収まるため，全ての処理を行っても遅延は小さいと言える．また，全てブロック間の信号インタフェースは統一されているため，間に任意の画像処理ブロックを新たに追加することや，ブロックの並べ換えが容易に可能な構成となっている．

D5005 を対象に合成した結果を表 1 に示す．

ALM の空き容量からは，1 コアあたり 60,000 から 70,000 ALM 程度であるデコーダコアの追加が可能に思えるが，追加しようとするとリソース不足になる．これは，Intel FPGA では ALM 10 個を 1 つにまとめた LAB (Logic Array Block) と呼ばれる単位で配置を行っており，合成した結果の LAB 使用率が 96.2%となっているためである．

3.4 認識処理への影響

AI 処理として，道路交通標識の認識を行い，レンズ歪みが認識処理に与える影響について評価する．道路交通標識は，ドイツのデータである GTSDB [6] を用いる．認識処理には，このデータベースを使ってトレーニングされた `faster_rcnn_inception_resnet_v2_atrous` [7] を使う．

GTSDB の画像には，図 3 のような大きいレンズ歪みが発生していないため，ここでの評価は，原画像と歪みを付加した画像とで比較を行う．実際のシステムでは，レンズによって歪んだカメラ画像が入力である．したがって，ここでの評価における歪み付加による認識精度の差分が，歪み補正によって与える影響と捉えることができる．

テストケースは原画像 (`original`)，水平画角 110 度の AXIS 製カメラ M1045-LW 相当の歪みを付加したケース (`distortion`)，さらに歪みを強くし水平画角 120 度相当のケース (`high_distortion`) の 3 つを比較する．

評価結果を図 6 に示す．歪みが大きい程，全ての検出精度の評価値が下がっているのが分かる．このことから，レンズ歪みによる交通標識検出の精度に対する影響が大きく，レンズ歪み補正の重要性が高いと言える．

3.5 メモリ帯域

1 フレームあたりのデータサイズは，解像度を 1280×720，リマップテーブルに保存する座標値を 16bit 固定小数表現

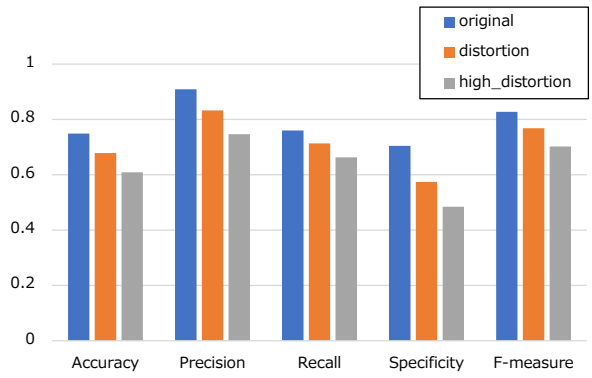

図 6 レンズ歪みによる交通標識検出処理への影響

（整数部 12bit，小数部 4bit）とすると，29.5Mbit が必要である．そして，30fps で 35 入力を処理できるとすると，31.0Gbps の帯域が必要である．

参考までに，この見積もり値がメモリ帯域のどの程度を占めるのかを試算する． D5005 に搭載されている DDR4-2400 は 1 チャネルあたり 19.2GB/s (=153.6Gbps) であるが，実効 50%とすると 76.8Gbps である．リマップ処理がこのうちの 40.4%を占めることになる．もし，リマップテーブルを 1/2 に間引けば，15.5Gbps の帯域が必要で，実効帯域の 20.2%を占める．

現状の見積もりでは，リマップ処理が使うメモリ帯域はやや大きいが，リマップテーブルの間引きと補間計算を用いれば，問題なく実現できるといえる．しかし，今後，画像の解像度を 4K や 8K に対応させたり，他の画像処理を追加実装したり，画像の入力数を増やそうとしたときには，メモリ帯域の増加は FPGA のリソース量の増加より緩やか [3] なため，メモリ帯域が問題となる可能性が高い．

4. 提案手法

多入力画像解析システムにおける，リマップ処理のメモリ帯域を低減させる手法を示す．

4.1 リマップ処理のグループ化

リマップテーブルは，レンズ歪みのため個々の入力ごとに異なる値を保持する必要がある．しかし，同じ製品であればリマップテーブルは似たような値になるため，リマップテーブルのグループ化ができる．グループ化したリマップテーブルは，製品によって決まるベース値と個体差を表現する差分値に分ける．ベース値は，あらかじめ製品のキャリブレーションにより平均的な補正パラメータを測定しておき，計算により求める．補正パラメータは，サイズが小さく FPGA 内の SRAM に置くことができるため，ベース値を得るために DRAM のメモリ帯域は使わない．差分値は，個々の入力ごとに異なる値を，ベース値からの差分で表現する．元の値と比較して差分値は絶対値が小さ

Vol.2022-ARC-248 No.34
Vol.2022-SLDM-198 No.34
Vol.2022-EMB-59 No.34
2022/3/10

表 1　D5005 への合成結果

	ALM	BRAM	Register	DSP	周波数
PTU	30,829	401	33,178	0	125 MHz
デコーダ	323,339	1,950	418,388	130	161 MHz
色変換	16,458	120	40,803	75	234 MHz
フレーム間引き	12,362	210	31,809	0	234 MHz
リマップ	45,009	4,155	94,880	165	234 MHz
縮小	25,213	510	68,500	270	234 MHz
出力	46,656	295	113,595	20	234 MHz
その他	140,167	650	276,539	0	-
使用率	68.6%	70.7%	28.9%	11.5%	-

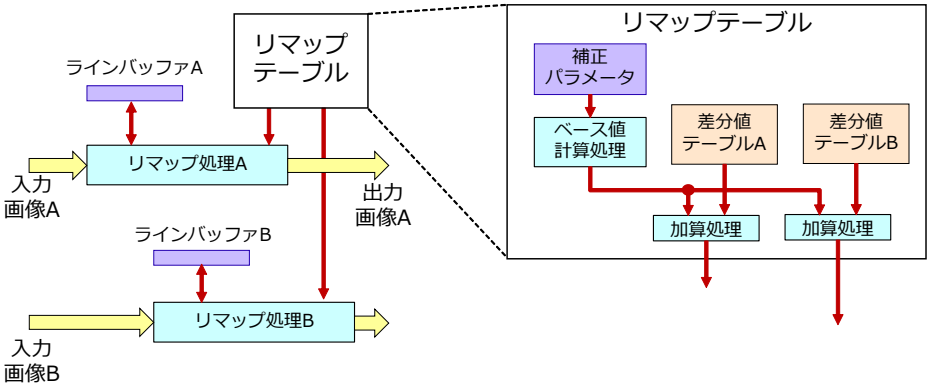

図 7　グループ化したリマップ処理

いため，必要なビット数を減らすことができ，メモリ帯域の低減となる．グループ化したリマップ処理の構造を図 7 に示す．図 7 の差分値テーブルは，リマップテーブルの差分値データである．

本論文におけるリマップブロックは，Algorithm 1 に示すアルゴリズムによって補正後の画像を生成する．リマップブロックは，前段ブロックからストリーム入力としてブロッキングで読み込み，後段のブロックへストリーム出力を行う．OpenCL の記述によるカーネル関数では，IO チャネルと呼ばれるストリーム入出力用の命令が用意されており，Algorithm 1 では，3 行目の READ_CHANNEL 関数，14 行目の WRITE_CHANNEL 関数がこれにあたる．出力画素を生成するための参照座標は，まずベース値を次式により計算し，DRAM から読み出したカメラ固有の差分値を加算することにより得られる．

$$\bar{x} = (k_1 r^2 + k_2 r^4)x + 2p_1 xy + p_2(r^2 + 2x^2) \quad (1)$$

$$\bar{y} = (k_1 r^2 + k_2 r^4)y + 2p_2 xy + p_1(r^2 + 2y^2) \quad (2)$$

$$where \quad r^2 = x^2 + y^2$$

ベース値を計算するための歪みパラメータは，ホストで実行されるソフトウェアによる起動時に，リマップ処理のカーネルに対して引数として渡される．差分値テーブルは，ホストによりグローバルメモリと呼ばれる DRAM の共有領域に，各チャネルごとに配置される．カーネルはリマップ対象の画素を処理するタイミングで，対応するチャネルの対応する画素位置のリマップテーブルを DRAM から読み出して，リマップ後の画素を生成する．この時，参照画素位置は固定小数点で示されるため，左上，右上，左下，右下に当たる 4 つの画素をラインバッファから取り出し，補間演算を行った画素値となる．

Algorithm 1 Undistorition Function

Input: Remap table address
Input: Distortion params
Input: Input Channel of pixel
Output: Output Channel of pixel

```
1: function UNDISTORTION(table, params, ich, och)
2:     while true do
3:         channel, pix, valid ← READ_CHANNEL(ich)
4:         if valid then
5:             WRITE_TO_LINE_BUFFER(pix)
6:             UPDATE_X_Y
7:         end if
8:         base_x ← CALC_UNDIST_X(params)
9:         base_y ← CALC_UNDIST_Y(params)
10:        dx, dy ← READ_DIFF_MAP(channel, x, y)
11:        ref_x ← base_x + dx
12:        ref_y ← base_y + dy
13:        pix_out ← INTERPOLATE(ref_x, ref_y)
14:        WRITE_CHANNEL(pix_out)
15:    end while
16: end function
```

4.2 メモリ帯域の見積もり

文献 [5] によると，640 × 480 の解像度で最大 4 ピクセル程度が発生する．扱う解像度を 1280 × 720 とすると，8 ピクセル程度のずれが発生すると見積もれる．このような座標値のずれを表現する差分値テーブルは，1 座標値あたり 8bit（整数部 4bit, 小数部 4bit）あれば十分である．

従来 16bit であったリマップテーブルの座標値は，提案手法により，ベース値はメモリからの読み出しがなくなり，差分値は 8bit になる．これらのことから，提案手法では最大で 50%のメモリ帯域を減らすことができる．

5. おわりに

我々は，多入力の画像を一拠点に集約して解析する多入力画像解析システムを開発している．これは，ネットワーク処理・デコード処理・画像処理の全てを FPGA で実行する． D5005 への実装では 35 チャネル処理できる．FPGA は，FPGA 内のリソース量の増加よりも FPGA 外のメモリ帯域の増加が緩やかなことから，メモリ帯域の制約が厳しい傾向にある．本論文では，画像処理のなかでも重要な歪み補正を実現するリマップ処理について，画像間でパラメータを共有することでメモリ帯域を低減する方法を提案した． D5005 への実装に対する見積もりにより，メモリ帯域は最大 50%低減できることがわかった．

今後の課題は，提案手法を D5005 に実装し，実際のシステムで評価することである．リマップ処理のグループ化の度合いでメモリ帯域の低減効果が変化するため，実際のシステムでどの程度の効果が得られるのかを評価する必要がある．また，本論文で見積もった差分値のビット数が十分であるかどうかを評価する必要がある．

参考文献

[1] Östberg, R.: Robustness of a neural network used for image classification : The effect of applying distortions on adversarial examples, `http://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Ahig%3Adiva-26118` (accessed Jan. 26, 2022.) (2018).
[2] Tomioka, Y., Takasu, R., Aoki, T., Hosoya, E. and Kitazawa, H.: FPGA implementation of exclusive block matching for robust moving object extraction and tracking, *IEICE TRANSACTIONS on Information and Systems*, Vol. 97, No. 3, pp. 573–582 (2014).
[3] Xilinx: Virtex UltraScale+ HBM FPGA: A Revolutionary Increase in Memory Performance (WP485), White paper (2019).
[4] Zhang, Z.: A flexible new technique for camera calibration, *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 22, No. 11, pp. 1330–1334 (2000).
[5] 門田仁，藤田悟：3D キャプチャリングの歪み補正方式，情報処理学会第 76 回全国大会，Vol. 2014, No. 1, pp. 215–216 (2014).
[6] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M. and Igel, C.: Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark, *International Joint Conference on Neural Networks*, No. 1288 (2013).
[7] Álvaro Arcos-García, Álvarez García, J. A. and Soria-Morillo, L. M.: Evaluation of deep neural networks for traffic sign detection systems, *Neurocomputing*, Vol. 316, pp. 332–344 (2018).