

実時間超解像回路の試作—ICBI アルゴリズムの FPGA 実装—

松本尚^{†1} 山本有紗^{†2} 城和貴^{†3}

概要: 本稿では、超解像アルゴリズムの1つである ICBI (Iterative Curvature Based Interpolation) をハードウェア化するための実装方法を提案する。超解像アルゴリズムとは、低解像度画像を高解像度画像に画像処理するための手法である。ICBI をハードウェア実装することによって実時間で処理できるようにするために、FPGA (Field Programmable Gate Array) を用いる。試作品を実装した結果、超解像アルゴリズムの性能を確認するとともに、回路数の評価を行う。

Real-Time Super Resolution: FPGA Implementation for the ICBI Algorithm

TAKASHI MATSUMOTO^{†1} ARISA YAMAMOTO^{†2} JOE KAZUKI^{†3}

1. はじめに

近年、信号処理 LSI などの発達によりパソコンやテレビなどのパネルの大型化、高解像度化、低価格化が急速に進んでおり、表示デバイスの解像度が向上し、1920×1080 画素の HD (High Definition) サイズが一般的になりつつある。録画解像度が表示デバイスの解像度以上の場合は、特に支障なく閲覧することができる。しかしながら、放送コンテンツや特殊な高解像度カメラ以外からの映像、ロボットに搭載されているものや監視カメラなどの安価な小型カメラからの映像、内視鏡カメラや各種検査用カメラなどの超小型カメラからの映像などの低解像度のコンテンツを、高解像度の表示デバイスに描く場合、画像の粗さが際立つ問題が発生している。この粗さを目立たなくするためには、低解像度のコンテンツを実時間で画像処理する技術が望まれている。

最も単純かつ短時間に処理できる技術として、画像補間技術がある。画像補間技術とは、画像の拡大を行う際にピクセルを補間する画像処理技術のことで、代表的なものとしてニアレストネイバー法[1]、バイリニア法[1]、バイキュービック法[1]が挙げられる。これらの技術は、比較的短時間で画像処理することが可能であるが、実時間では困難である。また、画像処理後の画像が不鮮明になることがある。

そこで、超解像技術に関する研究がすすめられている。超解像技術とは解像度の低い画像から、鮮明で解像度の高い画像に復元する技術のことであり、高解像画像に正しく復元するためには、低解像度画像では失われている高周波成分を推定する必要がある。高周波成分とは、画素の濃度値

が急激に変化している部分のことで、画像の輪郭部分などの強調する部分のことであり、超解像度技術として、イメージエンハンサー[2]、再構成超解像[3]、学習型超解像[4]、NEDI (New Edge-Directed Interpolation) [5]、FCBI (Fast Curvature Based Interpolation) [5]、ICBI (Iterative Curvature Based Interpolation) [5]がある。

超解像アルゴリズムは、ソフトウェアの面の改良では、計算量が多いため実行時間を短縮することが難しい。そこで、超解像アルゴリズム専用のハードウェアを開発する。ハードウェア実装方法として、GPU と FPGA (Field Programmable Gate Array) [6]がある。GPU の場合、回路数、演算処理が多くなるにつれて発熱しやすく、消費電力が多い。一方、FPGA は消費電力も低く、安価に手に入れられ、必要に応じて何度も回路を変更できる。ゆえに、本研究では、超解像アルゴリズムの1つである ICBI を FPGA を用いてハードウェア実装するための方法を提案する。

以下、2章では FCBI、ICBI のアルゴリズムについて述べ、3章では FPGA を用いた ICBI の実装方法の提案を行い、4章ではプロトタイプの開発について述べ、5章ではまとめについて述べる。

2. 超解像アルゴリズム

超解像アルゴリズムとして、FCBI と ICBI がある。これらのアルゴリズムは、複数フレームではなく、1つのフレームを用いて超解像を行う。

FCBI では、まず補間したいピクセルに接する8つのピクセルが含まれる斜め方向を2つ定め、それぞれの方向に対して局所近似二次導関数(曲率)を以下の式のようにしてそれぞれ求める。FCBI のイメージ図を図1に示す。ここで図1Aの斜め方向の曲率を $I_{11}(2i+1, 2j+1)$ 、図1Bの斜め方向の曲率を $I_{12}(2i+1, 2j+1)$ とおくと、曲率の求め方の式は以下ようになる。

^{†1} 奈良女子大学
Nara Women's University
^{†2} 奈良女子大学
Nara Women's University
^{†3} 奈良女子大学
Nara Women's University

$$I_{11}(2i + 1, 2j + 1) = I(2i - 2, 2j + 2) + I(2i, 2j) + I(2i + 2, 2j - 2) - 3I(2i, 2j + 2) - 3I(2i + 2, 2j) + I(2i, 2j + 4) + I(2i + 2, 2j + 2) + I(2i + 4, 2j)$$

$$I_{12}(2i + 1, 2j + 1) = I(2i, 2j - 2) + I(2i + 2, 2j) + I(2i + 4, 2j + 2) - 3I(2i, 2j) - 3I(2i + 2, 2j + 2) + I(2i - 2, 2j) + I(2i, 2j + 2) + I(2i + 2, 2j + 4)$$

それぞれの方向のうち、曲率が小さい方の2つの近接したピクセルの輝度の平均値を求め、その値を補間する点とする。この際、もし $I_{11}(2i + 1, 2j + 1) > I_{12}(2i + 1, 2j + 1)$ ならば、

$$\frac{I(2i, 2j) + I(2i + 2, 2j + 2)}{2}$$

$I_{11}(2i + 1, 2j + 1) < I_{12}(2i + 1, 2j + 1)$ ならば、

$$\frac{I(2i + 2, 2j) + I(2i, 2j + 2)}{2}$$

で表される。

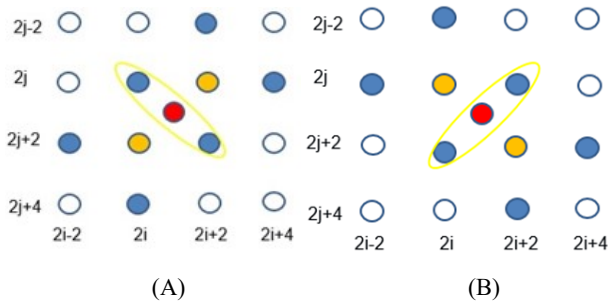
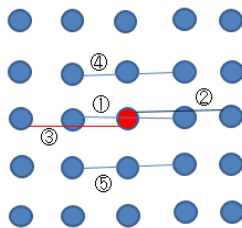


図1 FCBIのイメージ図

ICBIはFCBIの後工程のアルゴリズムで、FCBIで補間したピクセルをさらに曲率平滑化によって画像を滑らかにすることができる。FCBIで求めたピクセル周辺の二階微分の差分の総和の絶対値を求め、FCBIで補間したピクセルの輝度値を少しずつ変更して変更前と変更後の大小を比較して小さいほうを採用する。

$$|① - ②| + |① - ③| + |① - ④| + |① - ⑤|$$



3. FPGAを用いたICBIの実装方法の提案

3.1 基本構成

本提案では、ICBIをハードウェア化するために、集積回路FPGAを用いる。基本構成のイメージを図2に示す。構成要素は、Cameraモジュール、ZedBoard[7]、ディスプレイである。FPGAは、あらかじめ集積されている論理回路の組み合わせや接続を、設計者が変更しながら独自の論理回路を構築することができる集積回路である。記憶部にフリップ

フロップ回路を用いているSRAMという半導体メモリを用いることにより、設計した回路を変更することが可能である。

本提案におけるデータの流れを説明する。まず、Dual Port RAM(frame buffer)でカメラから出力されたラスタデータを一時的に保存し、カメラとディスプレイの非同期信号の同期合わせをする。次に、画像処理パイプライン(Image processing pipeline)で画像処理を行う。画像処理パイプラインにはピクセル補間パイプラインが含まれている。なお、パイプラインが対象とする処理は、ニアレストネイバ法、バイリニア法、バイキュービック法、FCBI、ICBIである。パイプラインの詳細については、3.2節で説明する。最後に超解像度されたデータをディスプレイに出力する。

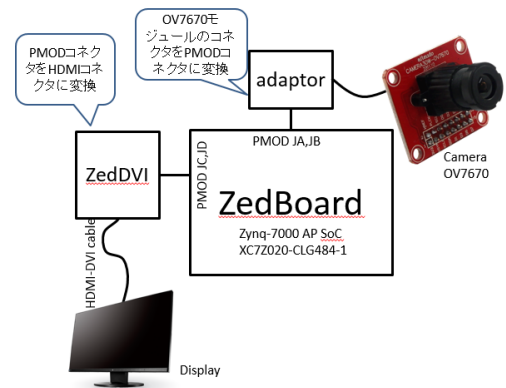


図2 基本構成

3.2 ピクセル補間パイプラインの構成

データ入力からColor conversionまでのイメージを図3に示す。入力で得られる色情報は、YUV422を採用する。

YUV422のイメージを図4に示す。U, V(色情報)は2ピクセルあたりに8bit×1個しか情報がないフォーマットである。そのため、パイプラインの前処理として、Data Arrangementを行う。具体的には、縦横4ピクセルに対して、輝度値はそれぞれのピクセルの値を代入し、色情報はすべて同じ値を代入する。

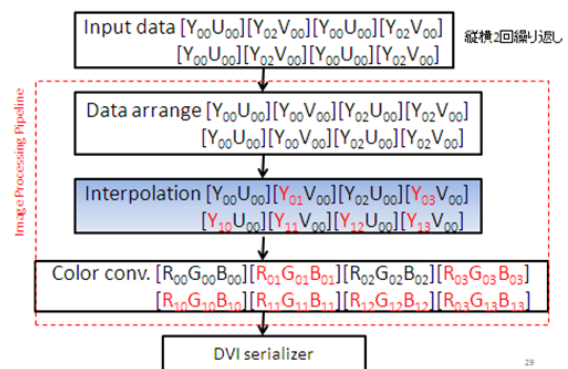


図3 データ入力からColor conversion

Interpolation PipelineはImage Processing Pipelineの1つである。FCBIは斜め方向と横方向の2回に分けて補間するピクセルの輝度値を算出するため、2回目の計算は1

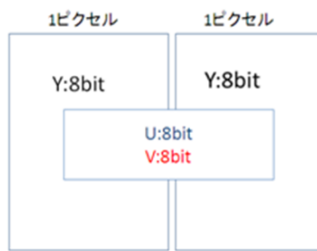


図4 YUV422

回目で求まるピクセルがすべて求まった後でないといけない。また、ICBIは1回目曲率の平滑化まで行った後、2回目のFCBIアルゴリズムおよび曲率の平滑化を行うものである。しかし、先にも述べたように、1回目と2回目のFCBIには依存関係があるため、逐次的に処理する必要がある。そのため、ICBIをパイプライン処理に入れることができない。そこで、ICBIアルゴリズムを1回ずつ行うのではなく、2つで1つの手順となるように改良する。具体的には、まずFCBIを2回連続で行い、FCBI適用後に2回連続で曲率平滑化を求める。この改良では、曲率平滑化の部分がパイプライン回路を複数段接続することが必要とされるため、各段のハードウェア量が大きくなってしまふ。ICBIをFCBIと曲率平滑化に分けて処理するため、平滑化処理の収束性が悪化する可能性がある。しかしながら、ソフトウェア実装を行い、確認を行った結果、画質や収束性に大きな差は確認できていない。そのため、提案手法では2つの処理に分けてハードウェア実装を行うこととする。曲率平滑化の計算処理は1クロックで処理することができない。このため、曲率平滑化の処理に関しては2クロックで実行することにする。これにより平滑化回路においてWNS(Worst Negative Slack)が発生することはなくなる。WNSは1クロックでの計算量が多く、次のクロックまでに計算処理が間に合わないときに発生するものである。また、曲率平滑化の処理パイプラインは8回路まで接続してLSI CADによる配置配線が可能である。

その後、画素補間パイプラインであるInterpolation Pipelineで画像のピクセル補間を行う。後処理として、ディスプレイへの出力の前にColor conversionにてYUV422からRGBのフォーマットに変換する。

3.3 カメラ出力、ディスプレイ出力

本研究では、解像度の低いカメラとして、1ピクセル当たり16bitの情報を出力するカメラを対象とする。1ピクセル当たり16bitであるためRGB系出力の最大情報量RGB565(R:5bit, G:6bit, B:5bit)であり、YUV系はYUV422(2ピクセルにつき Y:8bit*2, U:8bit*1, V:8bit*1)である。FCBIアルゴリズムでは、輝度の情報量が重要であるため、提案手法ではYUV422の方が有用である。仮に、RGBの各色プレーンに対してFCBIアルゴリズムを適用すると、細かい描写のある付近で補間する方向が各色で異なってしまい、偽色を生じる可能性がある。一方、Y(輝度)に基づく場合、

色情報をそれほど考慮しなくてもいいため、偽色を発生させる可能性が低い。この観点からも、提案手法では、輝度情報を適用すべきである。

また、YUV422は2ピクセルあたりに8bit×1個しかU(R), V(G)に関する情報がないフォーマットである。これは、人間の目が色情報よりも輝度に関する解像度が高いことに由来している。UV情報が2ピクセルあたり1個しか存在しないということは、Y情報よりも間引かれた状態から超解像度アルゴリズムを適用しなければならないことを意味する。ただし、輝度値で保管された値に、若干数値の異なる色情報を与えても、視覚的に大きな影響は生じない。このことは、白黒画像では明白である。そこで、本稿では、輝度値Yについてのみ着目し、FCBIアルゴリズムを実行する。ただし、今後は、色情報を考慮した方法を検討し、検証する必要があることに留意すべきである。

4 プロトタイプの開発

3章で提案した実装方法の有効性を確認するために、本章では、FPGAを用いて超解像アルゴリズムICBIを実行するためのハードウェアを開発する。ボードとして、ザイリンクスZedBoard(Zynq-7000 All Programmable SoC搭載)の低コスト開発ボードを用いる。カメラはOV7670というVGA形式のカラー映像を出力できる、広く利用されている安価なカメラを使用する。自動露出や自動ホワイトバランスなどの機能が組み込まれており、内臓レジスタの設定により、カメラの動作および機能を大幅に制御可能である。出力は1ピクセル当たり16bitであり、そのフォーマットはRGB系、YUV系、Raw出力であるBayer系を選択可能である。ディスプレイ出力フォーマットは1280×800近辺の解像度で、ドット周波数のなるべく低い規格を採用する。これらのことからディスプレイの出力フォーマットとしては、1280×768@60Hz 15:9, dot clock: 68MHz progressiveを採用して開発する。

まず、パイプラインの段数が及ぼす影響を確認する。このために、4, 7, 8回路カスケード接続する。ICBIの平滑化処理を実装するにあたって、7, 8回路をカスケード接続した設計を動作させると出力にノイズが乗ってしまい、良好に動作しているとは言い難い。これはパイプライン段数が大きすぎるのが原因であると考えられる。回路数が多いと良好に動作しないことから、ICBIにおいて平滑化パイプライン回路を4回路程度に収めないとZedBoardでは安定して動かないと結論づける。ここで、平滑化処理パイプラインのカスケード段数は曲率平滑化の反復回数となっている。平滑化パイプライン回路数を4回路としてICBIアルゴリズムをソフトウェア実装して実行したところ、エッジ付近にごみのようなピクセルが発生する。これはFCBIアルゴリズムが発生させたゴミをICBIの曲率平滑化のパイプライン

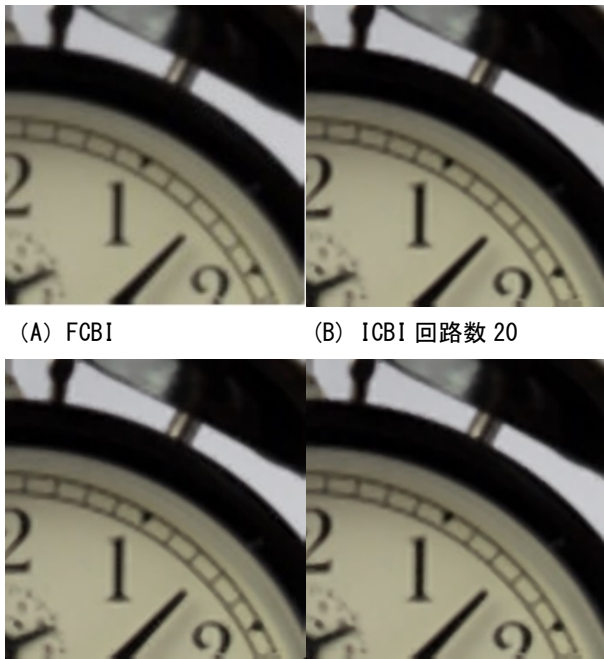


図 5 出力結果

イン 4 回路では除去しきれないためである。

回路数を減らした代わりに、ヒルクライム法の変化量の値を 2 倍強に大きくして平滑化処理の反復 1 回分の効果を増強させることにする。このことを本稿では ICBI ブースト版と呼ぶことにする。ゴミの出方から考えてブーストすることによる弊害はあまりないことが分かる。ブースト版の ICBI アルゴリズムをソフトウェア実装版に追加して実験した結果、回路数 4 回路でも良好な結果が得られている。図 5 に出力結果を示す。ブースト版 ICBI の回路数 4 と ICBI 回路数 4 を比較すると、ブースト版の方が解像度が高いことが分かる。ブースト版の ICBI でも良好な結果が得られることから、ICBI のハードウェア実装ではブースト版の ICBI アルゴリズムを採用する。

ハードウェア実装の結果、FCBI アルゴリズムは計算量が少ないため、少ないハードウェア量で実時間処理が可能である。その一方で ICBI アルゴリズムは曲率平滑化の部分がパイプライン回路を複数段接続することが必要とされるため、その 1 段 1 段のハードウェア量がかなり大きくなってしまふことが分かる。元の画像に超解像アルゴリズムを適用し 2 倍の画素数とした場合のディスプレイへの出力結果を図 6 に示す。左上から時計回りにニアレストネイバー法、バイリニア法、FCBI、ICBI の出力結果となる。それぞれの画像のエッジ部分に注目すると、ニアレストネイバー法、バイリニア法、FCBI、ICBI の順に解像度が良くなっていることが分かる。ゆえに、プロトタイプで実装する超解像アルゴリズムとして ICBI を用いることは適切であると考えられる。

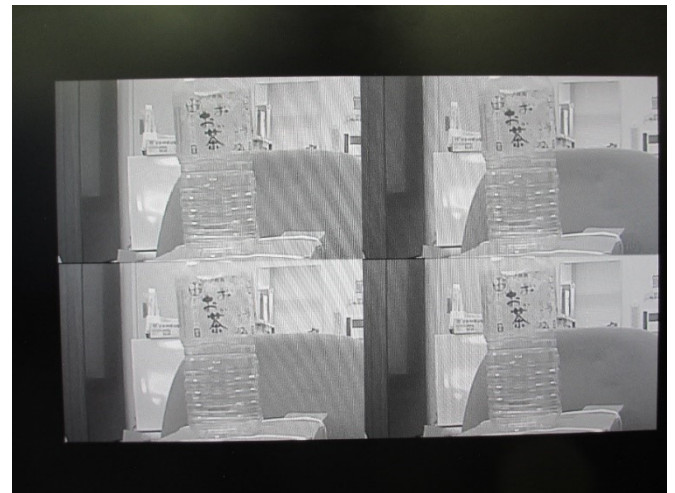


図 6 ディスプレイへの出力結果

3 まとめ

本稿では、FPGA を用いた ICBI アルゴリズムのハードウェア実装方法を提案している。これにより、カメラで撮影する映像を、ICBI アルゴリズムを適用して実時間で画像処理できるものである。今後の課題は、拡大率の向上である。拡大率を向上させるためには、プロトタイプの開発で用いた開発ボードを容量の大きいボードへ変更、パイプライン段数の多い ICBI の回路数の考慮が必要と考える。

参考文献

- [1]山本 裕 永原 正章 (2007) “デジタル制御理論による信号処理” 映像情報メディア学会誌 Vol. 61 No. 12 P 1710-1715
- [2]武内好男, 尾毛谷高 (1984) “高速度テレビカメラ・VTR システムの開発” テレビジョン学会誌 Vol. 38 No. 11 P 994-1000
- [3]松本 信幸 井田 孝 (2010) “画像のエッジ部の自己合同性を利用した再構成型超解像” 電子情報通信学会論文誌 D Vol. J93-D No. 2 pp. 118-126
- [4]石田 皓之 高橋 友和 (2006) “携帯カメラ入力型文字認識におけるぼけやぶれに対処するための生成型学習法” 電子情報通信学会論文誌 D Vol. J89-D No. 9 pp. 2055-2064
- [5]Andrea Giachetti Nicola Asuni (2011) “Real-Time Artifact-Free Image Upscaling” IEEE Transactions on Image Processing 20(10):2760 - 2768
- [6]小林 優 (2011) 『FPGA ボードで学ぶ組み込みシステム開発入門～Altera 編～』 技術評論社 384pp.
- [7] ” ZedBoard (Zynq-7000 All Programmable SoC 搭載) ” .AVNET. <http://www.avnet.co.jp/kits/Xilinx/AES-Z7EV-7Z020-G-J.asp> x, (参照 2016-06-20)