令和２（２０２０）年度　卒業研究報告書概要

課程，学籍番号，氏名	課程：電気・電子情報工学課程，学籍番号：B193212，氏名：猪谷 真吾
工学分野名：情報通信システム	指導教員名：市川 周一

題　　目：和　超解像画像生成のための

Back Projection ハードウェアの高位合成

（英　High-level synthesis of Back Projection Hardware for Super Resolution Image Generation　）

Abstract

Super-resolution is a technique to estimate a high-resolution image from one or more low-resolution input images. Back Projection (BP) is a method for estimating a high-resolution image so that the error of the downsampled image is minimized against the input low resolution image. This research designs and evaluates the BP hardware generated by C language and high-level synthesis. Two design optimizations were examined; that are pipeline processing using PIPELINE pragma (pipeline), and array partitioning using ARRAY _PARTITION pragma. Number of array divisions are 2, 4 and 8. C simulation and high-level synthesis were performed using Vivado HLS 2020.1. Latency and interval decreased by 59.7% using pipelining and array partitioning together. FF and LUT of optimization increased by 17.6% and 14.5%, respectively, using pipelining and array partitioning.

概　　要

超解像とは，1 つ以上の低解像度入力画像から，高解像度画像を推定することである．Back Projection とは，入力低解像度画像と，高解像度画像に対する撮像プロセスで生じる劣化とダウンサンプリングを経ることで得た低解像度画像との誤差が最小となるように，高解像度画像を更新し求める手法である．本研究では，先行研究でハードウェア化した Back Projection 処理を，Xilinx Vivado HLS を用いた C 言語と高位合成によって設計・評価することを目的とする．

Back Projection のうち低解像度画像と生成画像の距離を求める部分を，C 言語＋高位合成（HLS）を使用してハードウェアとして実装した．ハードウェアのパフォーマンスを向上させるため，Vivado HLS を使用して，高パフォーマンスが得られるようなハードウェアの検討を行った．本研究では，最適化手法として，パフォーマンスのためのパイプライン処理と，パフォーマンスのための構造最適化の 2 つの最適化手法を行った．パイプライン処理として，PIPELINE プラグマを下位階層のループに記述した．構造最適化として，ARRAY_PARTITION プラグマを使用して，大型の配列をより小型の配列に分割した．本研究では，最適化手法として，パイプライン処理を行う手法（pipeline），2 つの配列に分割する手法（array_partition_2），4 つの配列に分割する手法（array_partition_4），8 つの配列に分割する手法（array_partition_8）の 4 種類の手法で行なった．

Vivado HLS 2020.1 を用いて C シミュレーションおよび高位合成を行なった．合成レポートより，タイミング，開始間隔，レイテンシ，使用率の見積もりを調べて，ハードウェアリソースを評価した．最適化手法を用いない場合（none）と pipeline を比較すると，レイテンシおよび開始間隔が 19.9%減少し，ハードウェアリソースは FF（Flip Flop）が 9.6%減少したのみで大きな変化が得られなかった．none と array_partition では，分割数が 2，4，8 のいずれの場合でもレイテンシと開始間隔が 40.3%減少した．ハードウェアリソースは分割数が 2 つの場合，BRAM（Block Random Access Memory）は 50%，FF は 49.6%，LUT（Look Up Table）は 58.4%減少した．配列の分割数を増やすと，BRAM と FF，LUT は増加した．パイプライン処理および配列 2 分割を併用する最適化手法を評価した（optimization）．none と optimization では，レイテンシと開始間隔が 59.7%減少した．ハードウェアリソースについては，array_partition_2 と optimization では，FF と LUT が 17.6%，14.8%増加した．パイプライン処理をすることで同時処理演算とレジスタの使用量が増大するためと考える．

発表する際の課程を記入　電気・電子情報工学　課程　　発表番号　9

（学籍が他課程所属の学生も発表する課程を記入すること）