

# FPGA 上でのデュアルパイプラインを用いた BLOB 検出

野尻 直人† 孟 林‡ 山崎 勝弘‡  
立命館大学大学院 理工学研究科† 立命館大学 理工学部‡

## 1. はじめに

Binary Large Object(BLOB)検出は自立走行車の誘導や車載カメラ、道路標識認識、監視システムなど様々な分野で用いられており、認識速度と正確性を向上させることが重要である。BLOB 検出にはラベリングが必要であるが、データの依存性があるため並列化が困難である。本研究では、Field Programmable Gate Array (FPGA)上でデュアルパイプラインを用いて BLOB 検出をリアルタイムで実現する。デュアルパイプラインとは、画像を上下に2分割し、それぞれパイプライン処理を並列に行うことである。ラベリングではルックアップテーブル(LUT)内に連結成分を保存し、連結情報を用いて BLOB を検出する。本研究ではラベリングを簡略化し、仮ラベル生成、LUT の作成と更新、及び連結成分の評価のみを行い、並列化を図る。

## 2. デュアルパイプラインを用いた BLOB 検出

### 2.1 デュアルパイプラインシステムの構成

図 1 にデュアルパイプラインを用いた BLOB 検出システムを示す。本システムでは、4 つの画像処理モジュール (FE, GAU, LAB, BLOB)、3 つの制御モジュール (PIX\_Ctrl, GAU\_Ctrl, LAB\_Ctrl)、アドレス生成モジュール (Generate\_ADDR)、及び 3 つのパイプラインレジスタ (PIX\_Register, GAU\_Register, LAB\_Register) から構成される。FE は Block RAM (BRAM) から画像データを読み込む。GAU はガウシアンフィルタと 2 値化を用いて、ノイズを除去する。LAB は仮ラベルと LUT を作成する。最後に、BLOB で LUT 内の仮ラベル連結成分を検出し、BLOB の個数と各 BLOB の面積、重心を算出する。1 枚の画像を上部和下部に 2 分割し、上のパイプラインが上部を、下のパイプラインが下部を担当し、FE から BLOB までの処理を並列に行う。FPGA の BRAM を十分に使用するために、1024x256 サイズの画像を対象にシステムを作成した。

### 2.2 ラベリングの簡略化

画像処理において、ラベリングは基本的かつ重要な処理である。通常、仮ラベル生成、LUT の作成と更新、ラベル補正から構成される。1 つの BLOB 内に仮ラベルが複数存在する場合、LUT 内の同じ行に連結成分を保存し、ラベル補正の際に、LUT を参照して画像中の仮ラベル番号を全て書き換える必要がある。

本研究での簡略化したラベリングを図 2 に示す。仮ラベル生成において、図 2 a) のように 1 つの BLOB に仮ラベル番号 1,2,3 が存在する。これらの連結成分を LUT 内に保存し、その情報を用いて 1 つの BLOB として検出する。

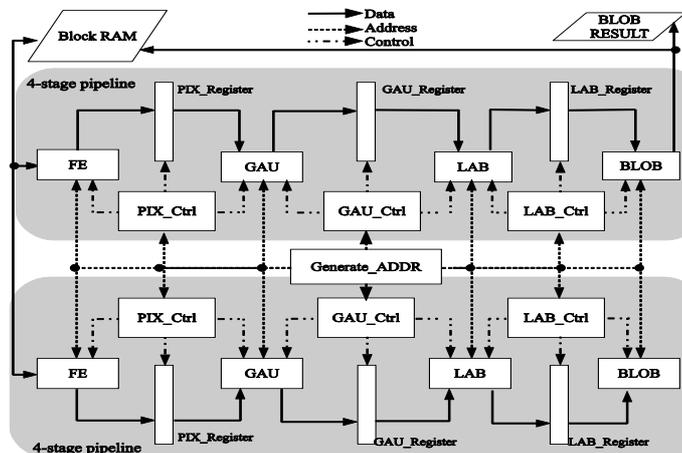
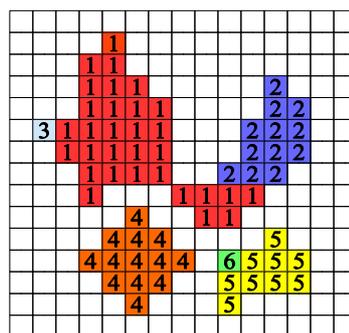


図 1. デュアルパイプラインを用いた BLOB 検出システム



BLOB	仮ラベル連結成分
1	1,2,3
2	4
3	5,6

a) 仮ラベル生成      b) LUT  
図 2. 簡略化したラベリング

また本研究では、境界線付近に BLOB が存在する場合でも連結成分を評価することで検出が可能となる。

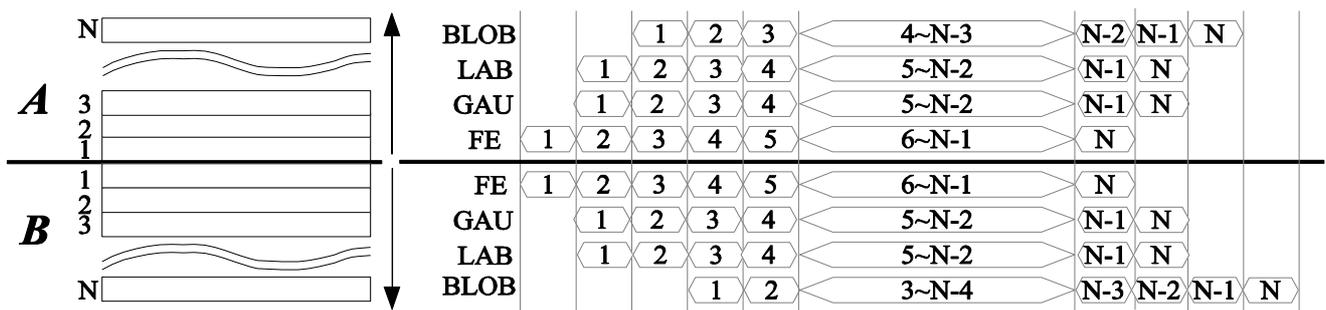
### 2.3 タイミングチャート

デュアルパイプラインを実現するために、原画像を図 3 a) のように上部(A)と下部(B)に分割する。行番号を図の左側に示し、行番号 1 から開始して矢印方向に処理を進める。

図 3 b) に各処理モジュールのタイミングチャートを示す。数値は画像の行番号を示す。ガウシアンフィルタは注目画素の次のラインに依存するので、GAU の実行は A と B において FE から 1 ラインの遅延が必要である。ラベリングは注目画素の次のラインに依存しないので、LAB の開始は GAU と同一となる。

本研究では、画像を 2 分割して BLOB を検出しているため、境界線付近に BLOB が存在する可能性がある。境界線付近ではデータの依存関係が生じるので、B1 の BLOB は A1 の BLOB より 1 ライン遅延させる必要がある。

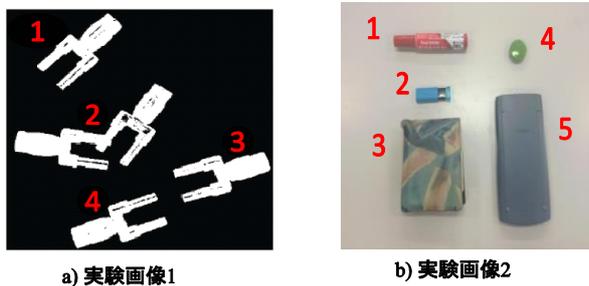
FPGA-based BLOB Detection Using Dual-pipelining  
Naoto Nojiri†, Lin Meng‡, Katsuhiko Yamazaki‡  
†Graduate School of Science and Engineering, Ritsumeikan University  
‡College of Science and Engineering, Ritsumeikan University



a) 画像の2分割

b) タイミングチャート

図3. デュアルパイプラインのタイミングチャート



a) 実験画像1

b) 実験画像2

図4. 実験画像

表1. ハードウェア使用量

	デュアルパイプライン	シングルパイプライン	Kiran	Virtex5
<b>Registers</b>	660 (1.5%)	453 (1.0%)	--	44800
<b>LUTs</b>	3758 (8.4%)	1945 (4.3%)	2784 (6.2%)	44800
<b>LUT-FF pairs</b>	518 (24.4%)	271 (12.7%)	--	2127
<b>BRAM</b>	138 (93.2%)	132 (89.2%)	4 (2.7%)	148

カッコ内の値は、デバイスの使用率である。

表2. 動作周波数と実行時間

	デュアルパイプライン	シングルパイプライン	Kiran
<b>動作周波数(MHz)</b>	99.18		100
<b>実行時間/ピクセル (ns)</b>	10.08		--
<b>全実行時間 (μs)</b>	57.70	112.13	220
<b>速度向上率 (倍)</b>	3.81	1.96	1

### 3. FPGA 上での実験

#### 3.1 実験条件

本研究で用いた実験画像の例を図4に示す。画像サイズは 100x100 ピクセルである。比較対象は、先行研究[1]とデュアルパイプラインと同様の処理を行うシングルパイプラインとする。図4 a)は[1]から引用する。評価対象はハードウェアサイズと実行時間である。

FPGA は Xilinx 社の Virtex5(XC5VFX-70T)評価ボード、デザインツールは ISE 14.5、シミュレーションツールは ISim、ハードウェア設計言語は Verilog 2001 である。

#### 3.2 ハードウェアサイズ

表1にハードウェア使用量を示す。デュアルパイプラインシステムのハードウェア使用量はシングルパイプラインシステムの約2倍となっている。理由としては、デュアルパイプラインシステムは2つの画像処理パイプラインを有しているからである。

#### 3.3 動作周波数と実行時間

表2に動作周波数と実行時間を示す。デュアルパイプラインシステムの動作周波数は 99.18MHz で、シングルパイプラインシステムと同じである。またピクセル当りの実行時間は 10.08ns となり、全実行時間は 57.70μs である。

デュアルパイプラインによる BLOB 検出は、リアルタイムで実行でき、[1]よりも 3.81 倍速く、シングルパイプラインよりも 1.94 倍速い。

### 4. おわりに

本研究では、FPGA 上でデュアルパイプラインを用いて BLOB 検出を高速に行う手法を提案した。100x100 ピクセル画像において、デュアルパイプラインによる BLOB 検出は 57.70μs で実行でき、[1]よりも 3.81 倍速い。デュアルパイプラインによる BLOB 検出は、Vertex5 のハードウェアをわずかしか使用していない。現在、車両検出や道路標識認識のような特定のアプリケーションを対象として、研究を進めている。

### 参考文献

- [1] D.Kiran, A.I.Rasheed and H.Ramasangu, "FPGA Implementation of Blob Detection Algorithm for Object Detection in Visual Navigation," the 2013 Int. Conf. on Circuits, Controls and Communications (CCUBE), pp.1-5, 2013.
- [2] N.Nojiri, L.Meng and K.Yamazaki, "FPGA-based BLOB Detection Using Dual-pipelining," the 2015 ACM/SIGDA Int. Symp. on Field-programmable gate arrays.