§2-II-1 LabVIEW による画像処理 I

(Image processing with LabVIEW I)

1. 実験の狙い

本実験では LabVIEW と Web カメラを使用した 2 値化と差分処理の画像処理のプログラムを作成する．作成過程において画像処理の流れを理解する．

2. 解説

世の中には数多くの画像ファイルフォーマットがある．本解説ではその中でも代表的なファイル形式であるビットマップ形式について解説する．

ビットマップ画像

ビットマップ画像は，ビットマップ表現（ラスタ表現）を用いている画像である．イメージがグリッド（格子状）に分割され，グリッドの各ピクセルの光の値（明るさ，暗さ，または色）はそれぞれ個別に記録される．

ビットマップファイルフォーマット

ビットマップファイルのファイル構造は，ファイルヘッダ，情報ヘッダ，ビットマップデータとなっている．ビットマップの種類には Windows と OS/2 があり，情報ヘッダの最初の 4bit の情報を読み取ることで，種類を識別することが可能である．

- ファイルヘッダ

  14 byte 固定のデータサイズを持ち，表 1 のような情報を有している．

表 1　Windows BITMAPFILEHEADER

オフセット	サイズ	名前	解説
0	2	bfType	ASCII “BM”
2	4	bfSize	ファイルのサイズ （バイト単位）
6	2	bfReserved1	ゼロ
8	2	bfReserved2	ゼロ
10	4	bfOffBits	ファイルの中でのイメージの開始位置

- ビットマップヘッダ（Windows）

  ファイルヘッダに続き，ビットマップヘッダがある．オプションでカラーマップもある．ビットマップヘッダ構造は，BITMAPINFO と呼ばれ，そのうちカラーマップまでのフィールドは BITMAPINFOHEADER と呼ばれることがある．ビットマップヘッダには表 2 のような情報が記載されている．

- カラーマップ（Windows）

  1 ピクセルあたり 1, 4 または 8 ビットを使用するイメージは，カラーマップがなければならない．カラーマップのサイズは通常それぞれ 2, 16 または 256 エントリであるが，イメージで

色の完全なセットを必要としなければ，それよりも小さい場合もある．biClrUsed のフィールドがゼロでなければ，そのフィールドに使用する色の数が入っており，その数はカラーマップのエントリ数でもある．biClrUsed のフィールドがゼロである場合はカラーマップはフルサイズである．カラーマップのエントリはそれぞれ 4 バイトであり，表 3 に示すとおりである．

表 2　Windows BITMAPINFOHEADER イメージヘッダ

オフセット	サイズ	名前	解説
14	4	biSize	ヘッダのサイズ．40 バイト
18	4	biWidth	イメージの幅（ピクセル単位）
22	4	biHeight	イメージの高さ（ピクセル単位）
26	2	biPlanes	イメージプレーン数．1 でなければならない
28	2	biBitCount	ピクセルあたりのビット数。1, 4, 8 または 24
30	4	biCompression	圧縮形式
34	4	biSizeImage	圧縮されたイメージのサイズ
38	4	biXPelsPerMeter	水平解像度
42	4	biYPelsPerMeter	垂直解像度
46	4	biClrUsed	使用するカラー数。
50	4	biClrImportant	「重要」な色の数
54	4 x N	bmiColors	カラーマップ

表 3　Windows RGBQUAD カラーマップエントリ

オフセット	名前	解説
0	rgbBlue	カラーマップエントリの青の値
1	rgbGreen	カラーマップエントリの緑の値
2	rgbRed	カラーマップエントリの赤の値
3	rgbReserved	ゼロ

・ ビットマップデータ（Windows）

ビットマップデータは，カラーマップの直後に続く．ビットは論理的に一度に一行ずつ保存される．それぞれの行は，境界が 4 バイトの倍数に合うように，値ゼロのバイトでパディングされる．

- ピクセルあたり 1 ビットのビットマップ

  ピクセルは単一のビットであり，8 ピクセルが 1 バイトにパックされる．バイトの上位のビットが最左端のピクセルである．

- ピクセルあたり 4 ビットのビットマップ

  圧縮されていないイメージでは，2 ピクセルが 1 バイトにパックされ，上位 2 ブル（4 ビット）が最左端のピクセルである．それぞれの行は，境界が 4 バイトの倍数に合うように，値ゼロでパディングされる．

- ピクセルあたり 8 ビットのビットマップ

  圧縮されていないイメージでは, 1 ピクセルが 1 バイトにパックされ, それぞれの行は, 境界が 4 バイトの倍数に合うようにパディングされる．

➢ ピクセルあたり 24 ビットのビットマップ

各ピクセルは 3 倍とであり，青，緑，赤の値をこの順に含んでいる．それぞれの行は，協会が 4 バイトの倍数にあうように値ゼロでパディングされる．

・ ビットマップヘッダ（OS/2）

OS/2 のビットマップ形式は，Widows 形式に似ているが，それよりもやや簡単である（表 4 参照）．ビットマップヘッダ構造は BITMAPCOREINFO と呼ばれ，カラーマップまでのフィールドは，BITMAPCOREHEADER と呼ばれる場合がある．

表 2　OS/2 BITMAPCOREINFO イメージヘッダ

オフセット	サイズ	名前	解説
14	4	bcSize	ヘッダのサイズ．12 バイト
18	2	bcWidth	イメージの幅（ピクセル単位）
20	2	biHeight	イメージの高さ（ピクセル単位）
22	2	bcPlanes	プレーン数．1 でなければならない．
24	2	bcBitCount	ピクセルあたりのビット数．1, 4, 8 または 24
26	3 x N	bmciColors	カラーマップ

・ カラーマップ（OS/2）

1 ピクセルあたり 1, 4 または 8 ビットを使用するイメージは，カラーマップがなければならない．カラーマップのサイズはそれぞれ 2, 16 または 256 エントリである．カラーマップのエントリは表 5 のように 3 バイトである．

表 5　OS/2 RGBTRIPLE カラーマップエントリ

オフセット	名前	解説
0	rgbBlue	カラーマップエントリの青の値
1	rgbGreen	カラーマップエントリの緑の値
2	rgbRed	カラーマップエントリの赤の値

・ ビットマップデータ（OS/2）

ビットマップデータは，カラーマップの直後に続く．それぞれの行は，境界が 4 バイトの倍数に合うようにパディングされる．

ビットマップファイルの画像処理

ビットマップの画像処理は，ビットマップデータの数値を変更することで行うことが可能である．実際に画像処理を行う場合は，ビットマップデータを 2 次元配列などに格納し，数値処理を施す場合が多い．

3. 実験の予習

***予習レポートとして整理して提出すること**

予習1： 下記の用語について，意味を調査し理解を進めておく．
黒字は必ず．その他は余裕があれば行うこと．

a. 2 値化	b. 画像のデータ構造	c. 画像処理
d. 差分処理	e. FFT	f. エッジ検出
g. RGB	h. バイナリ画像	i. グレースケール画像
j. HSV 表色系	k. JPEG 画像	l. PNG 画像
m. fps	n. 輝度	

予習2： 今回利用する Web カメラ（HD Pro Webcam C920）の性能について簡単に調べてくる．

4. 実験

4.1 実験の概要

別資料の手順を確認しながら，例題のプログラムを作成し，以下の項目ができることを確認する．

- Web カメラからの画像取得
- Web カメラからの動画像取得
- 2 枚の画像の差分画像を取得
- NI Vision Assistant を利用した 2 値化処理プログラムの作成

4.2 実験で使用する機材

・パーソナルコンピュータ
・Web カメラ（HD Pro Webcam C920）

実験室の PC には（各自が持参した）USB を接続しないこと

4.3 実験終了後の確認事項

実験終了後 TA に以下の項目を確認してもらうこと．

- 各ファイルが全てあるか
  vi ファイル，JPG，HTML ファイル
- 上記確認後デスクトップ（および PC 内）に実験データファイルが残っていないこと．
  実験データは各自が Moodle 管理すること．

4.3　実験手順

[1]　実験の進め方

～13:50	手順 1：例題 1 の実施　［資料 p. 1-2］ LabVIEW を用いて簡単に Web カメラからの画像が取得できることを確認する.
～14:10	手順 2：例題 2 の実施　［資料 p. 3］ 動画像の取得ができることを確認する.
～14:40	手順 3：例題 3 の実施　［資料 pp. 4-6］
～15:40	手順 4：例題 4 の実施　［資料 pp. 7-8］
～16:20	手順 5：例題 5 の実施　［資料 pp. 9-12］ NI Vision Assitant を用いることで簡単にプログラムが作れることを確認する.
～16:25	手順 11：TA による試問

[2]　各プログラムの詳細

- 各プログラムの詳細は Moodle 上に上がっているので，実験前に各自が確認をしておくこと.

5. 実験レポート

検討すべき事項（考察事項）

実験レポートでは，実験結果を踏まえて検討を行い，下記の項目に関してもよく考察をすること．

① 差分計算をするためにグレースケールに画像を変換するが，差分画像をカラーで表示するにはどのようなプログラムにすればよいか？

② 2 値化画像を取得する際には，閾値を的確に選ぶ必要があるが，環境に応じて適切な値が変動してしまう．閾値を動的に選択する方法としてどのような方法があるか考えよ．
※ある程度状況を設定して考えよ．

さらに調査すること

次回は本実験をベースに行うため，しっかりと理解した上で予習を行ってくること．

課題

課題1： 例題 4，5 で作成したプログラムのフローチャートを示せ

課題2： 本実験で使用した Web カメラの特徴と各解像度における fps (Frame Per Second)を調査せよ
（実験中に NI MAX で確認しておくこと．）

課題3： 画像処理は様々な現場で使用されている．特に品質管理においては不良品を排除するためによく用いられている．品質管理などに用いられている画像処理を調べ，その原理を説明せよ．（2 つ以上）

課題4： 例題 3 で作成したプロパティノードで設定可能な項目を調べて列挙せよ（項目と説明）．

6. 参考文献

・ 橋本岳・山本茂広・浦島智著 「LabVIEW 画像計測入門」（講談社）
・ David C. Kay & John R. Levine 著
「グラフィックファイルフォーマットハンドブック」（アスキー出版局）

[1] 例題 1： USB カメラの接続と画像の取り込み

USB カメラから画像取り込みを行う．

1) 画像を表示させる領域の設定（フロントパネル）
   - ブランク VI を開く．
   - （右クリック）Vision＞Image Display (Silver)  を配置する．

2) USB カメラからの画像を表示させる（ブロックダイアグラム）
   a. （右クリック）ビジョン&モーション＞Vision Utilities  ＞Image Management＞IMAQ Create  を選択，配置する．
   b. ビジョン&モーション ＞NI-IMAQdx  ＞IMAQdx Open  を配置
   c. ビジョン&モーション ＞NI-IMAQdx  ＞IMAQdx Snap  を配置
   d. ビジョン&モーション ＞NI-IMAQdx  ＞IMAQdx Close  を配置
   e. b. c. d.の順に error out → error in,　Session In → Session Out を接続する
   f. c. Snap の Image Out を Image Display の入力に接続
   g. b. Open の Session In を右クリック＞作成＞制御器　を選択
      使用できるカメラをフロントパネルで選択 (ex. cam0 など)
   h. d. Close の error Out を右クリック＞作成＞表示器を選択
   i. IMAQ Create の New Image を Snap の Image In に接続
      IMAQ Create の Image Name を右クリック＞作成＞定数 “camera Image”とする
   j. 実行ボタン押す．画像が取り込める

3) 画像の保存

a. ビジョン&モーション＞Vision Utilities＞Files＞IMAQ Write File 2 を配置し，c. Snap の Image Out と Image を接続し，File Path に制御器を接続し，ファイル形式を「bmp」から「jpg」に変更する．
b. 適当なフォルダ（デスクトップでもよい）に jpg ファイルを作成し，フロントパネルの File Path を作成した jpg ファイルとなるよう設定する．
c. 実行して，ファイルが正しく保存されていることを確認する．

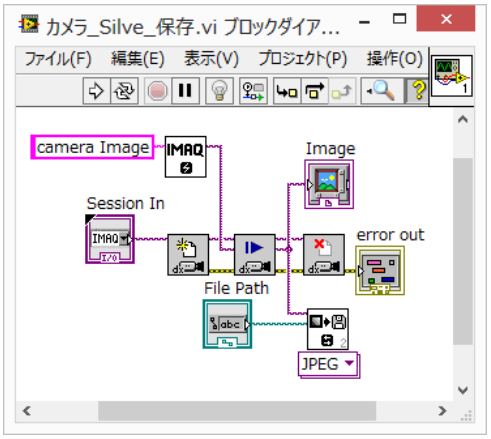

図 1 ブロックダイアグラム

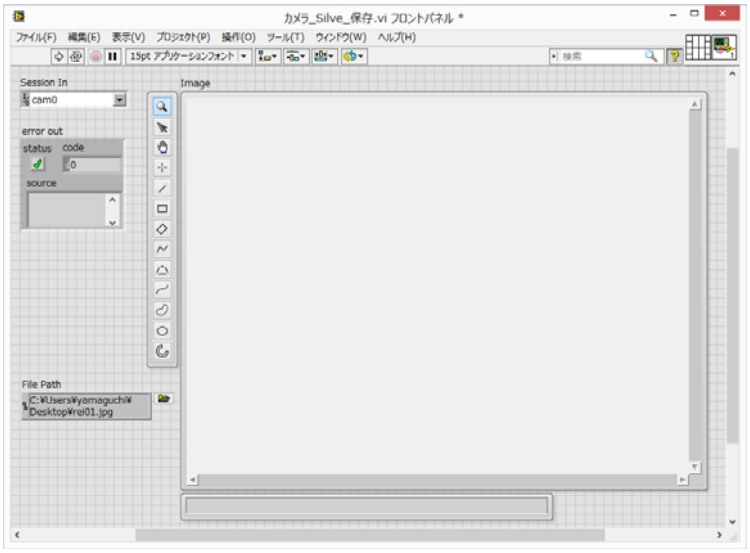

図 2 フロントパネル

4) USB カメラの設定

a. Measurement & Automation Explorer を起動 (アイコンは“NI MAX”)
b. マイシステム ＞ デバイスとインタフェース ＞NI-IMAQdx Devices を順にクリック
c. “cam0”など対象となるデバイスをダブルクリック
d. Video Mode で所望の解像度などに変更することが可能
本実験では“800 x 600 RGB24”を選択し使用する
e. 上部の Save をクリックし設定を保存する

[2] 例題 2： USB カメラの接続と画像の取り込み

USB カメラから動画を取り込む.

1) 画像を表示させる領域の設定（フロントパネル）
   - ブランク VI を開く.
   - （右クリック）Vision＞Image Display (Silver) を配置する.

2) USB カメラから動画像を取得する（ブロックダイアグラム）
   a. （右クリック）ビジョン&モーション＞Vision Utilities＞Image Management＞IMAQ Create を選択，配置する．IMAQ Create の Image Name を右クリック＞作成＞定数 “camera Image”とする
   b. ビジョン&モーション＞NI-IMAQdx＞IMAQdx Open を配置
   c. ビジョン&モーション＞NI-IMAQdx＞IMAQdx Configure Grab を配置
   d. b. c. の順に error out → error in, Session In → Session Out を接続する
   e. b. Open の Session In を右クリック＞作成＞制御器 を選択し，ラベルを IMAQdx Session とする.
      使用できるカメラをフロントパネルで選択 (ex. cam0 など)
   f. プログラム＞ストラクチャ＞While ループ を配置
      終了条件に制御器を接続する.
   g. ビジョン&モーション＞NI-IMAQdx＞IMAQdx Grab を While ループ内に配置し，c. の error out と Session Out からそれぞれ接続する.
   h. ブロックダイアグラム内の Image (Image Display (Silver))を While ループ内に移動し，g. の Image Out と接続する.
   i. ビジョン&モーション ＞ NI-IMAQdx＞IMAQdx Close を While ループ外に配置し，g. の error out，Session In と接続し，error out に表示器を接続する.
   j. 実行してフロントパネルに動画が表示されることを確認する.
   k. 動画の保存に関しては後述する.

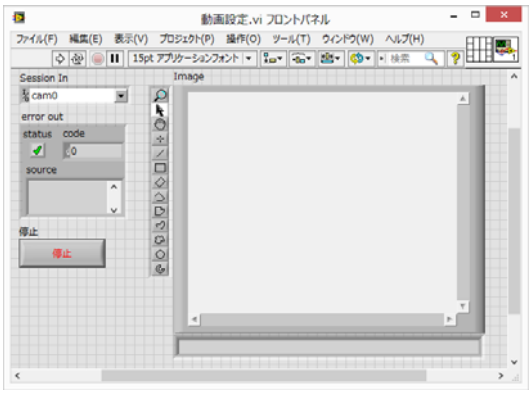

図 3 例題 2 のフロントパネル

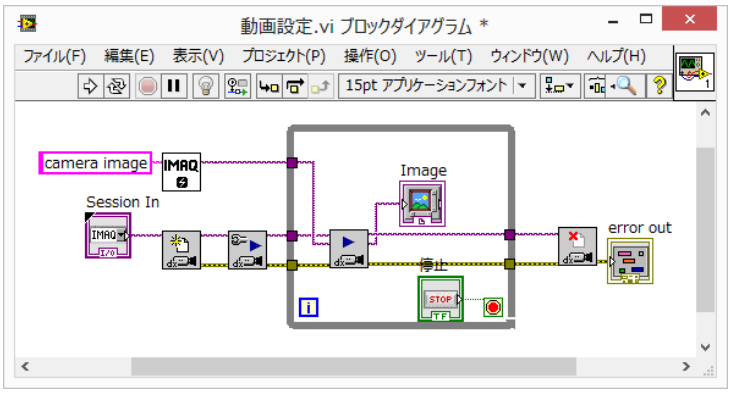

図 4 例題 2 のブロックダイアグラム

[3] 例題 3：カメラパラメータの設定（解像度，フレームレートの設定）

例題 1 において，カメラパラメータを Measurement & Automation Explorer というソフトを用いて変更を行っていた．しかし，実際の制御では解像度などをプログラム内で変更できた等が便利であることが多い．また，カメラパラメータは画像処理の数値計算に利用することができる．そこで，プロパティノードを使用したカメラパラメータの設定・取得方法を学ぶ．

概要

例題 2 を参考に，まず使用中のカメラの解像度をプルダウンメニューから選べるように変更し，撮影を開始した後は，現在の解像度を表示するようなプログラムとする．

1) プログラミング＞ストラクチャ  ＞フラットシーケンシャルストラクチャ  を配置し，シーンを 1 つ追加し全 2 シーンとする．
2) **2 番目のシーン**に例題 2 のプログラムを Copy&Paste し，IMAQdx Open と IMAQdx Configure Grab の error と Session の接続を外す．
3) ビジョン&モーション＞NI-IMAQdx  ＞プロパティノード  を配置し，IMAQdx の Session Out とリファレンスを接続する．正しく接続されると表示されている文字が App から IMAQdx に変わる．IMAQdx Open の error Out とプロパティノードの error In，プロパティノードの error Out と Session Out と IMAQdx Configure Grab の error In と Session In とを接続する．
4) プロパティノードにポインタを合わせたときに現れる中央下のボックスを使いプロパティボックスを 3 つに増やす．プロパティと表示されている場所を左クリックし，Acquisition Attributes＞Video Mode に変更し，右クリックを押して書き込みに変更するを選択する．残りのプロパティを Acquisition Attributes＞Height と Acquisition Attributes＞Width とし，表示器を接続する．
5) フロントパネルにおいて モダン＞リング＆列挙体＞メニューリング を配置し，ブロックダイアグラムにおいて 4)で作成した Video Mode に接続する．
6) 図 4 中の “Session In” を “IMAQdx Session” に変更する．
7) **最初のシーン**に プログラム＞ケーストラクチャ＞While ループ  を配置し，その中に プログラム＞ケーストラクチャ＞イベントストラクチャ  を配置する．
8) イベントストラクチャ内に OK ボタンを配置する．
9) IMAQdx Session を右クリック＞プロパティノード＞値 を選択し，作成されたプロパティノードをイベントストラクチャ内に配置する．
10) イベントストラクチャの上部中央部分を右クリック＞イベントケースを追加 を選択し下記の 2 つのイベントを追加する．
    a. OK ボタン
    b. IMAQdx Session
11) イベントストラクチャの上部中央部分を左クリックし「“IMAQdx Session” : 値変更」を選択し，内部に IMAQdx Open とプロパティノードと IMAQdx Close を配置し，順に error と Session を接続する．
12) IMAQdx Session (制御器)のプロパティノード(値)を作成し，9) IMAQdx Open の Session In と接続する．
13) 11)のプロパティノードのプロパティを 2 つに増やし，1 つめを Cammera Attributes＞Active Attribute とし，書き込みに変更し定数文字列「videomode」を接続する．2 つめを Cammera

Attributes＞Enum＞Strings[]　とし，5)のメニューリングのプロパティノード(文字列 [])を書き込みに変更したものと接続する．

14) イベントストラクチャの上部中央部分を左クリックし「“OK ボタン”：値変更」を選択し，内部に IMAQdx Session (制御器)のプロパティノード(値)と プログラミング＞文字列＞空白文字列定数 を作成し．プログラミング＞比較＞等しくない を配置し，2 つを入力する．

15) プログラミング＞ブール ＞And を配置し，12)の出力と OK ボタンを接続し，出力を While ループの終了条件に接続する．

16) While ループ外に IMAQdx Session (制御器)のプロパティノード(値)と(Disabled)，メニューリングのプロパティノード(Disabled)，OK ボタンのプロパティノード(Disabled)をすべて書き込みに変更に変更して配置する．IMAQdx Session (制御器)のプロパティノード(値)と空白文字列定数を接続し，(Disabled)のいずれか 1 つの入力に右クリック 作成＞定数を選択し，定数を「Enabled」とし，残りの 2 つとも接続する．

17) 2 番目のシーンにも(Disabled)を同様に作成し，定数を「Disabled and Grayed Out」とする．

18) 実行して，IMAQdx Session を変更すると，そのカメラで表示可能な解像度の一覧がメニューリングより選択可能となる．

19) OK ボタンを押すと選択した解像度で動画が取得される．

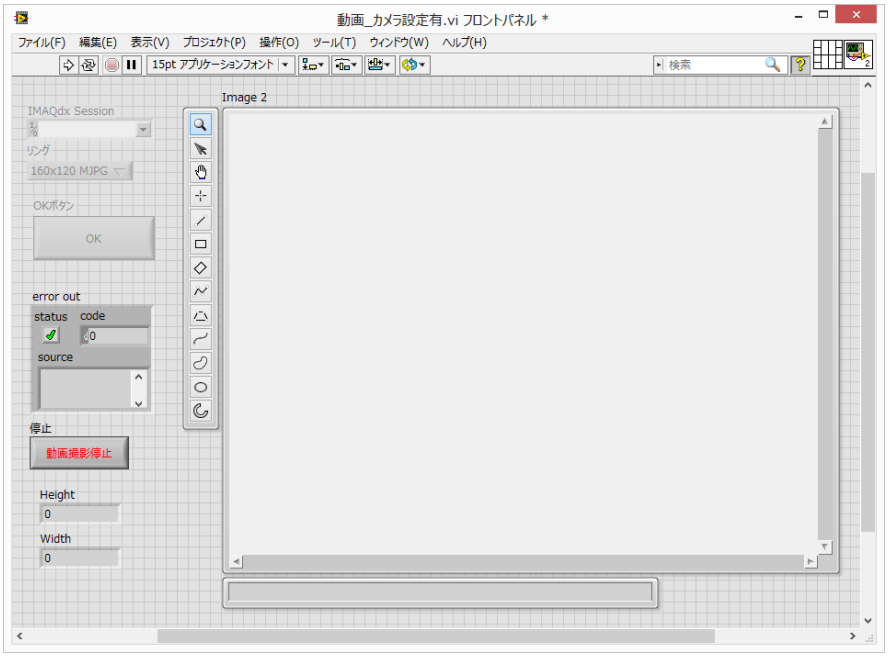

図 5 例題 3 のフロントパネル

実験番号：2-II-1
実験タイトル：LabVIEW による画像処理

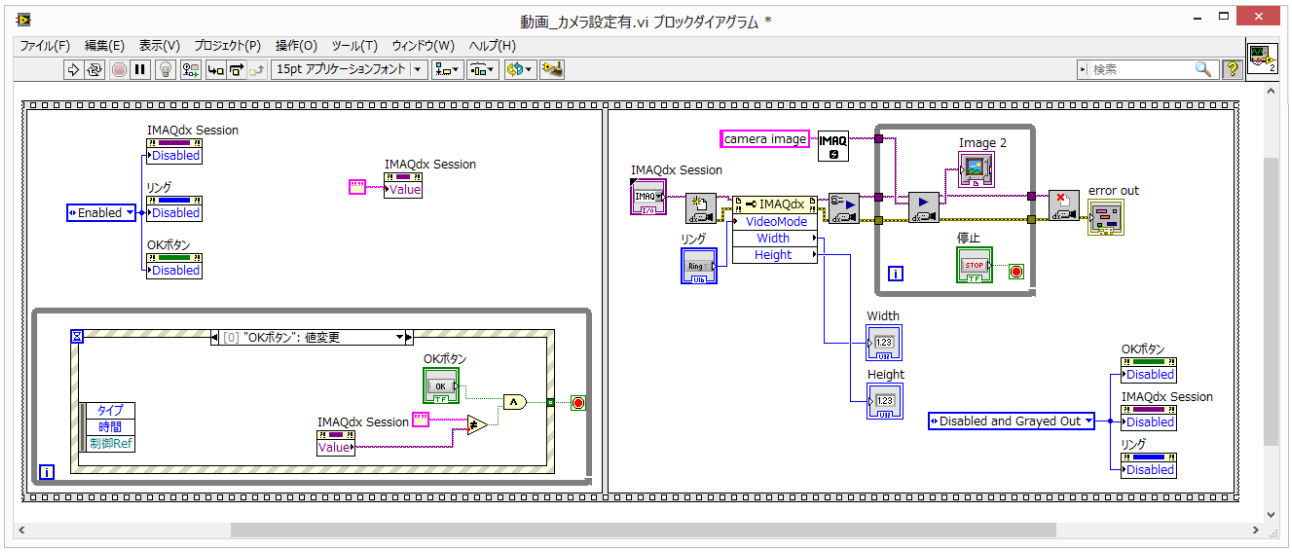

図 6 例題 3 のブロックダイアグラム 1

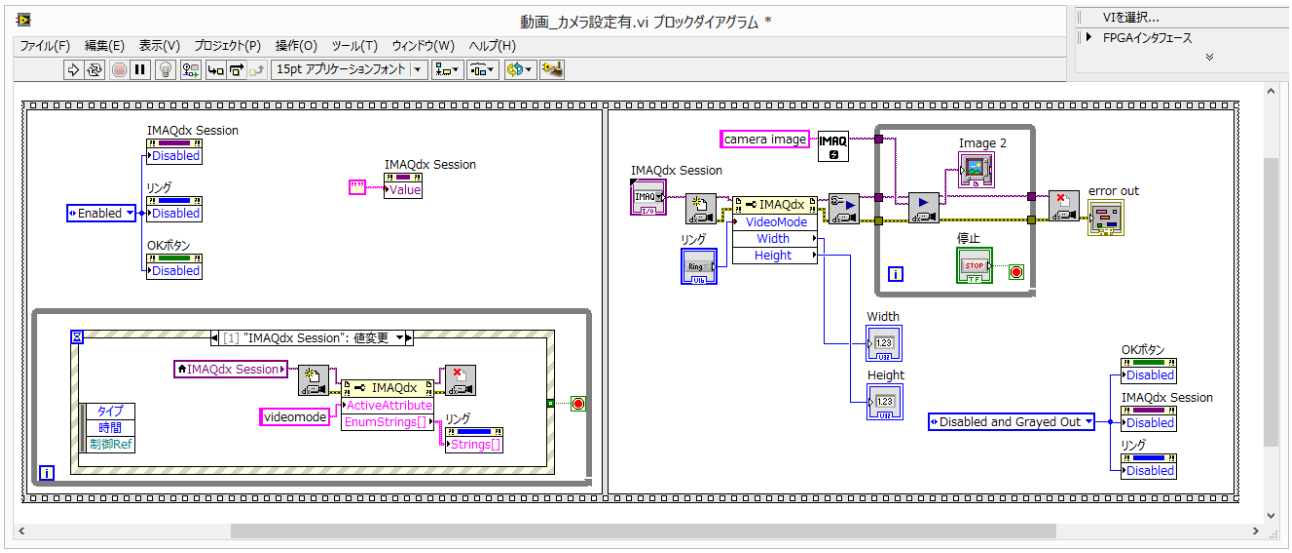

図 7 例題 3 のブロックダイアグラム 2

[4] 例題 4：画像の差分処理

これまでの実験において，カメラから画像や映像が取得可能であることが確認された．例題 4 では 2 枚の画像の差分をとるプログラムを作成することで，実際の画像処理を確認する．

1) フラットシーケンシャルストラクチャを作成し，シーン数を 5 とする．
2) 最初のシーンと 3 番目のシーンに While ループを作成し，終了条件に制御器を接続する．
3) 2 番目のシーンと 4 番目のシーンに画像を取得するプログラム(例題 1)を作成する．
   ※画像の保存処理は必要ない．
   ※IMAQ Create の Image Name の定数はそれぞれ異なるものとすること．
4) ビジョン&モーション＞Vision Utilities＞Image Management＞IMAQ Create を 3 つ配置し，Image Name に文字列定数 “img1”，“img2”，“calculated Image”とする．
5) ビジョン&モーション＞Vision Utilities＞Color Utilities＞IMAQ ExtractSingleColorPlane を 2 つ配置．Image DST と 4)で作成した IMAQ Create の New Image（“img1”，“img2”）を接続する．Image Src と 3)で作成した IMAQdx Snap の Image Out を接続する．片方の Color Plane に制御器を接続し，もう片方ともつなぐ．
   ※フロントパネルにおいて，Color Plane を選択するが，選択する色は対象物によって変化させること．
6) ビジョン&モーション＞Image Processing＞Operators＞IMAQ Subtract を配置．Image Src A と B に 5)で作成した IMAQ ExtractSingleColorPlane の Image Dst Out を接続する．Image Dst に 4)で作成した IMAQ Create の New Image（“calculated Image”）を接続する．
   error Out に表示器を接続する．※その他の error に関しては適宜つなげる．
7) フロントパネルに Image Display (Silver)を作成し，6)で作成した IMAQ Subtract の Image Dst Out と接続する．
8) 実行をして，1 回目の撮影ボタンを押し，取得画像に変化を与えるものを置き，2 回目の撮影を行い，差分画像が取得できることを確認する．

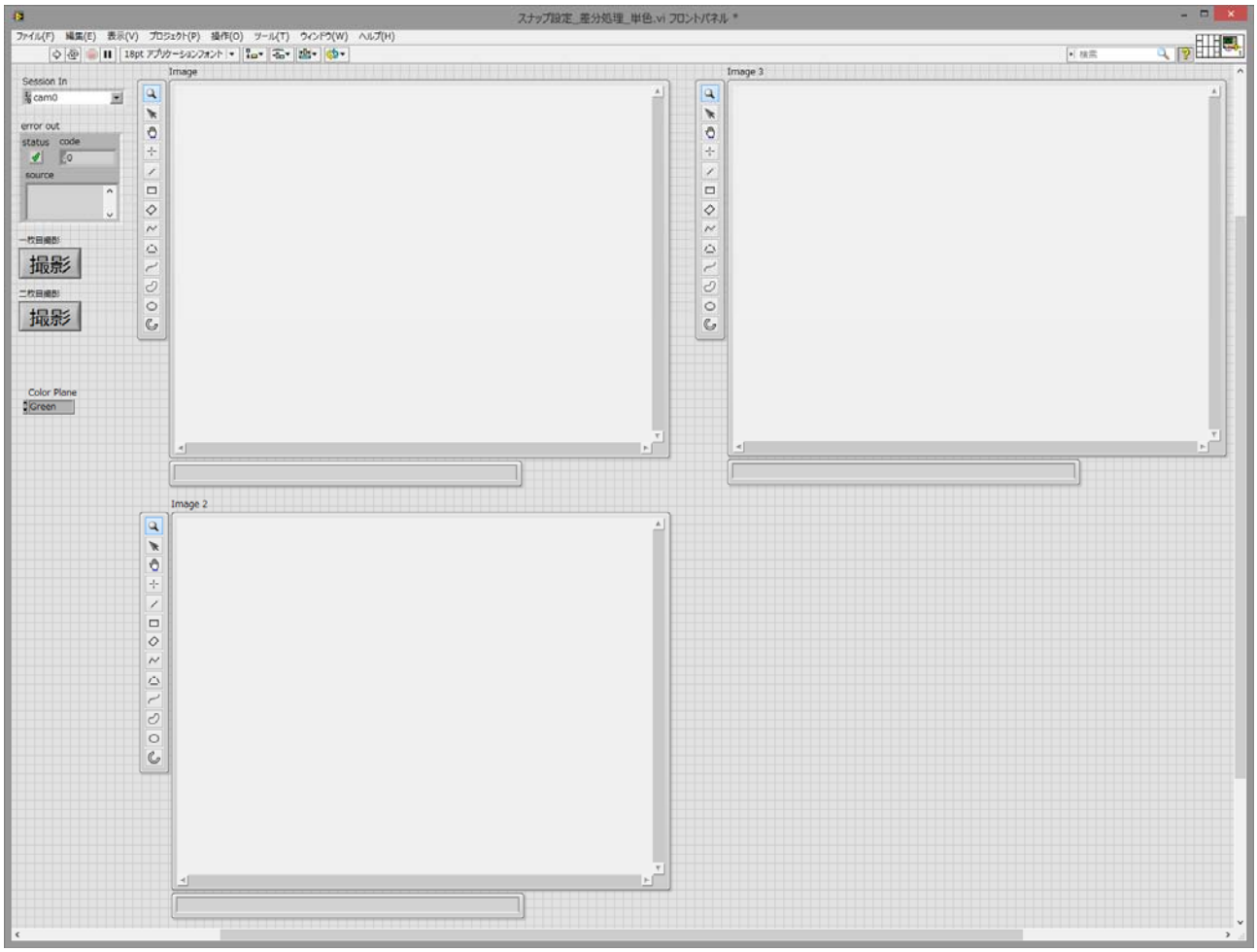

図 8　例題 4 のフロントパネル

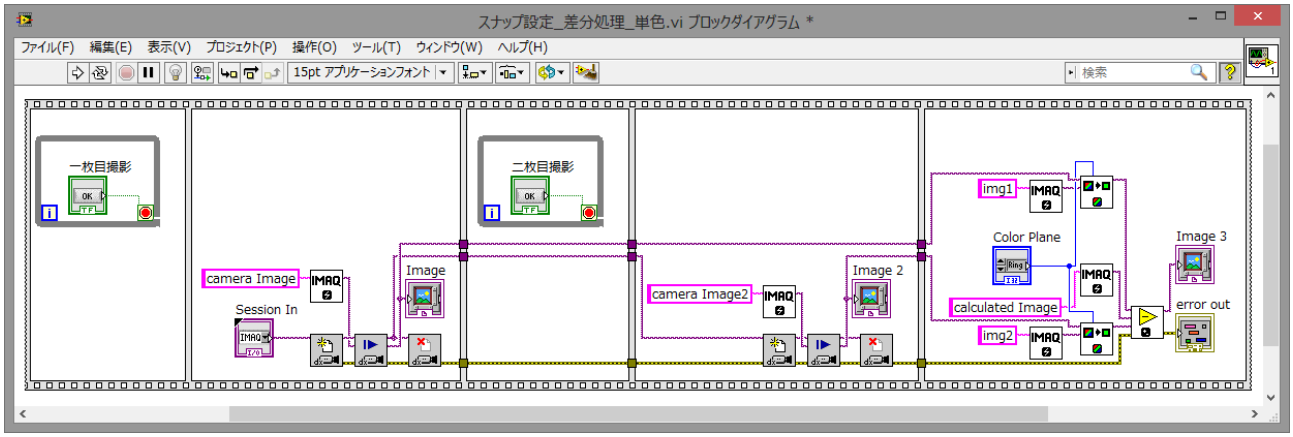

図 9　例題 4 のブロックダイアグラム

[5] 例題 5： NI Vision Assistant を利用した画像の 2 値化処理プログラムの作成

LabVIEW による画像処理プログラムにはこれまでの実験で確認したような，ブロックを一つ一つ配置し作成していくことで，目的の処理を行わさせることができる．しかし，毎回同じような処理を書かなくてはならない，画像処理特有の癖があるなどわかりにくいところも多い．このため，LabVIEW には NI Vision Assistant と呼ばれる，処理プログラム作成のためのサポートソフトウェアが実装されており，例題 5 では NI Vision Assistant を用いた 2 値化処理プログラムの作成を行う．その後プログラムを追加することで，2 値化されたピクセル数を表示するプログラムとする。

1) 例題 1 のプログラムを使用して，画像ファイルを作成する．
   カメラの解像度を「800 x 600」とすること．
2) デスクトップから NI Vision Assistant を開く．
3) NI Vision Assistant – ようこその画面が出たら，「画像を開く」をクリックして，1)で作成した画像を選択する．
4) 下図のような画面が現れる．※下図は真っ白の画像を選択している．

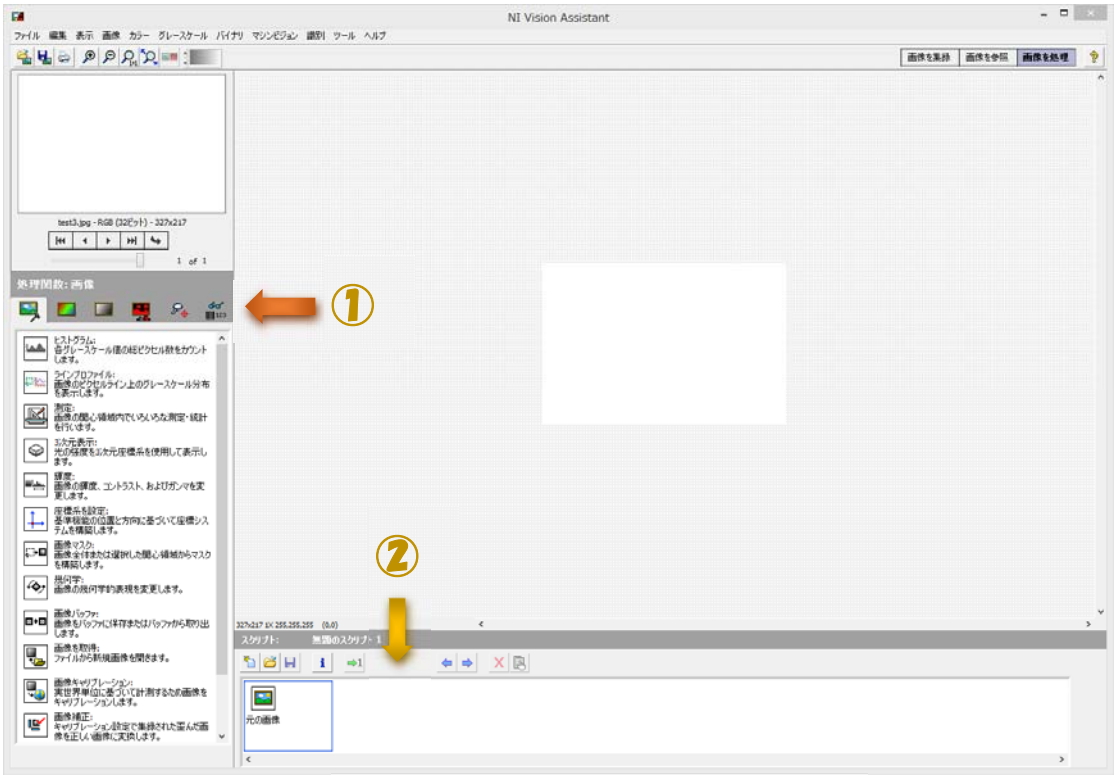

図 10　NI Vision Assistant 画面

5) 図 10 の①の箇所（処理関数）を選択することで様々な処理を実行することが可能となる．
   また，②の箇所には処理の流れが表示される．
   ※処理関数には左から「画像」「カラー」「グレースケール」「バイナリ」「マシンビジョン」「識別」と項目ごとにまとまっている．
6) 処理関数（カラー）からカラープレーン抽出を選択し，「HSL－明度(L)プレーン」を選択し，OK ボタン（画面下）を押す．
7) 処理関数（グレースケール）から 2 値化を選択し，手動 2 値化（デフォルト）とし，閾値の範囲を適当な値（後程変えるので適当な値でよい）にする．OK ボタンを押す．

8) 処理関数（グレースケール  ）から数量化  を選択し，画像全体をドラッグして選択する．
※メニューバーの下のツールが長方形になっていることを確認すること．
図 9 の②の箇所の面積が 100%であることを確認したら OK を押す．
9) これで 2 値化処理が作成できたので，最後にこの処理の vi ファイルを作成する．
   a. メニューバーより ツール＞LabVIEW VI を作成 を選択．
   b. ステップ 1/4：ファイルパスを選択し，次へ
   c. ステップ 2/4：変更せず次へ
   d. ステップ 3/4：IMAQdx 画像収録を選択し，次へ
   e. ステップ 4/4：全てのチェックボックスにチェックを入れて，終了．
10) 作成された vi ファイルを開き，メニューバー下の「ダイヤグラムをクリーンアップ」を選択する．フロントパネルも適宜整理しておく．
11) IMAQdx Open の Session In の定数文字列を削除し，制御器を接続する．
12) フロントパネルの Range (2 値化 1)の各制御器を右クリックし，置換＞水平ポインタスライダに変更する．
変更後，右クリックをしてプロパティを選択後，「スケール」のスケール範囲の最大値を「255」とし，「概観」の[デジタル表示器を表示]にチェックをして，増分/減分ボタンを表示のチェックを外す．見やすいように，配置を変える．
13) ビジョン&モーション＞NI-IMAQdx  ＞プロパティノード  を配置し，リファレンスと IMAQdx Open の Session Out を接続する．例題 3 の 4)を参考にプロパティを 2 つにして，Height と Width のプロパティを作成する．
14) プログラミン＞数値  ＞積  を配置し，13)の Height と Width を接続する．
15) Result (数値化 1) (IVA Quantify の表示器) を削除し，プログラミン＞配列  ＞指標配列  を配置し[配列]に接続する．指標（行）は 0 とし，[部分配列]に表示器を接続する．
16) 指標配列をさらに 2 つ追加し，[配列]に 15)で作成した指標配列の[部分配列]を接続する．
指標(行)をそれぞれ，1 と 0 とし[部分配列]に表示を作成し，指標 1 のラベルを “比率”，指標 0 のラベルを“面積比率”とする．
17) プログラミン＞数値  ＞積  を配置し，16)の指標 1 の[部分配列]と 14)の積の出力を入力する．出力に表示器を作成する．
18) プログラムを実行し，閾値によってピクセル数が変わることを確認する．
※面積比率が 100 であることを確認すること．面積比率は現在対象としている面積が画像全体にたいしてどの程度かを表すものである．比率は全面積に対する，閾値以上のピクセルの割合である．

実験番号：2-II-1
実験タイトル：LabVIEW による画像処理

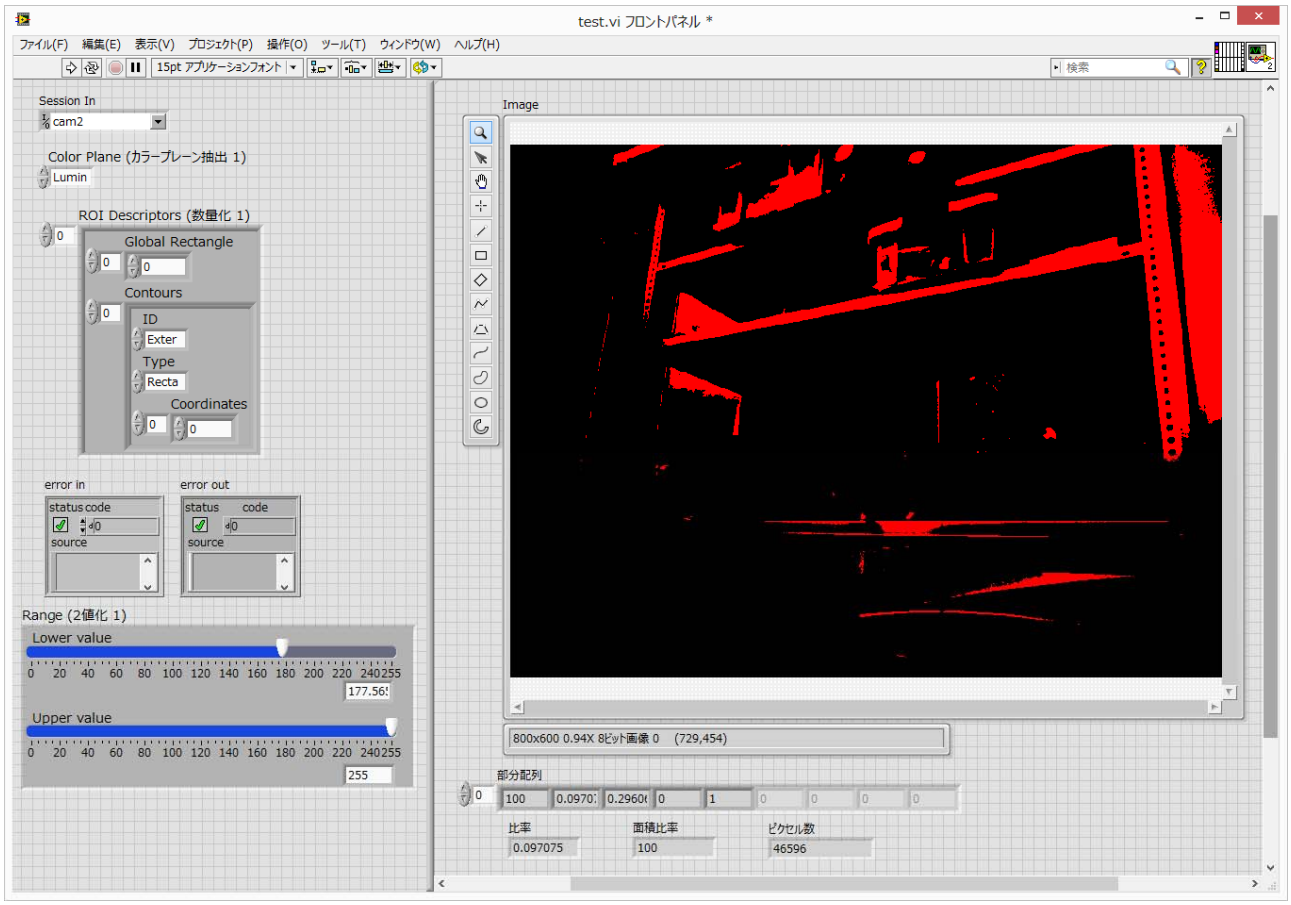

図 11　例題 5 のフロントパネル

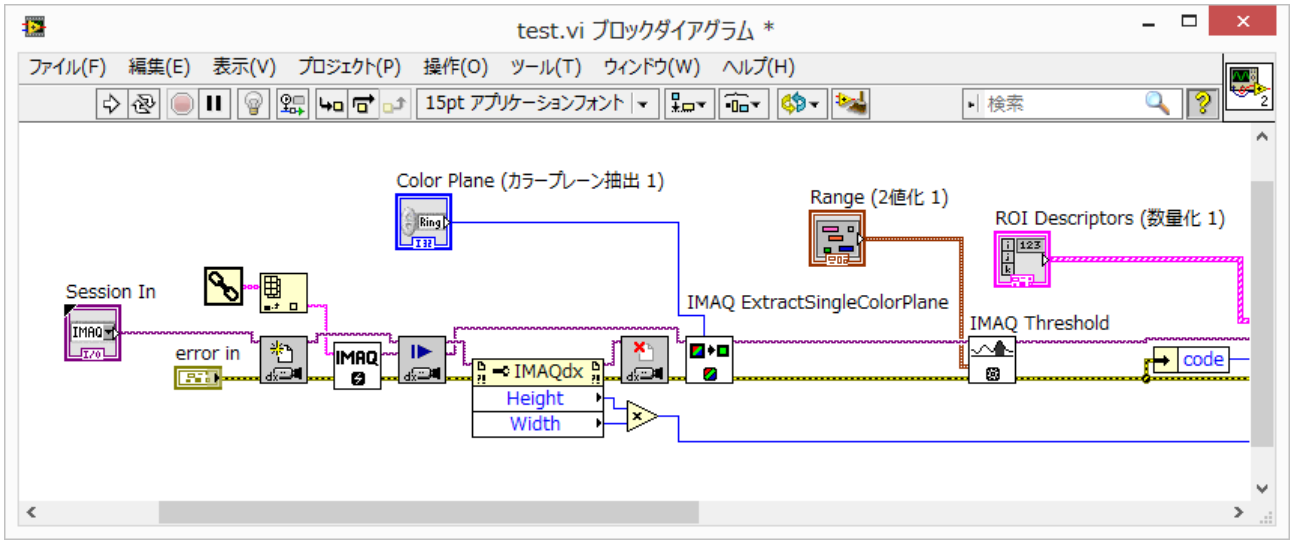

図 12　例題 5 のブロックダイアグラム左

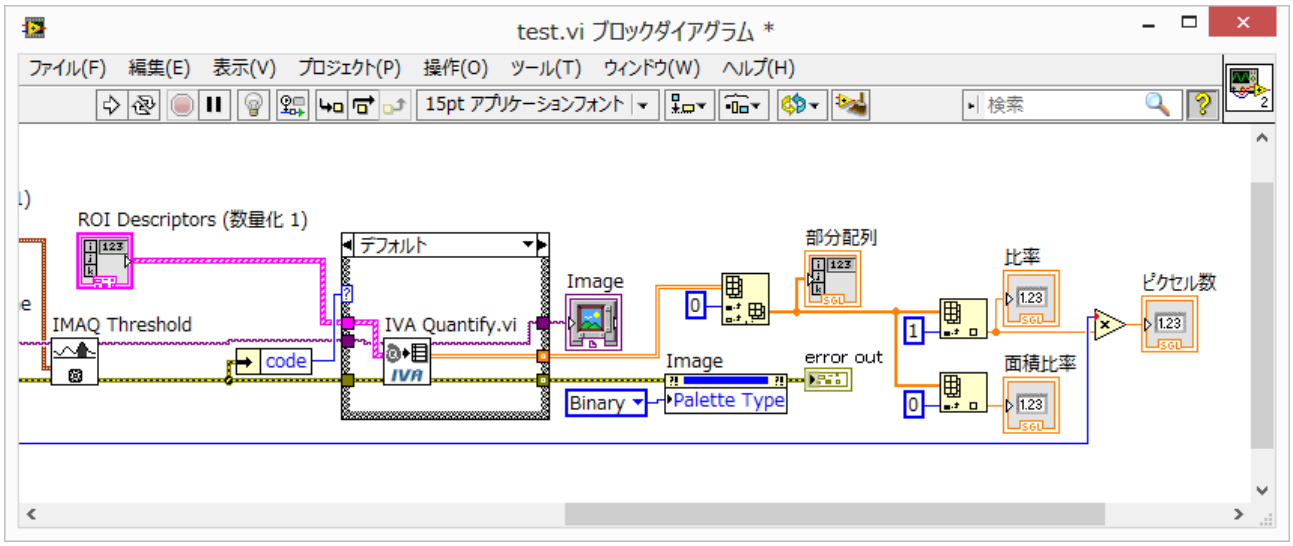

図 13　例題 5 のブロックダイアグラム右