RT コンポーネント操作マニュアル

屋内地図モデルの簡易生成コンポーネント

2014 年 10 月 31 日版

芝浦工業大学

ロボティクスシステムデザイン研究室

立川将　土屋彩茜　奥野万丈　竹村友晃　佐々木毅

更新履歴

2014 年 10 月 31 日版　第 1 版.

目次

1. 本提案の概要

1.1 提案の背景

現在大規模商業施設の増加など、屋内環境を把握する必要がある場面は増えてきています。しかし、テナントなどにより構造が多様化するといった理由から、環境把握は困難なものとなっております。そこで、既存の地図を利用してシステム上の地図モデルを簡易生成することができれば、屋内サービスへの発展が期待されると考えました。

このような背景から、屋内地図の簡易生成アプリケーションを RT コンポーネント群として開発することとなりました。これにより屋内向けサービスロボットへの軌道生成や、Google glass など別アプリケーション上での利用など、他の RT と関連した様々な用途が可能となります。また、一つのアプリケーションとしてではなく、複数のコンポーネントとして開発を行ったため、テストケースや画像処理コンポーネントとしてそれぞれ利用していただくことができます。

コンポーネントの設計指針等につきましては、

立川　将, 土屋　彩茜, 奥野　万丈, 竹村　友晃, 佐々木　毅,
“屋内地図モデルの簡易生成コンポーネント”,
第14回計測自動制御学会システムインテグレーション部門講演会, 2014.

に詳細がありますので、そちらもご参照いただけましたら幸いです。

1.2 開発環境

本コンポーネント群は Windows にて動作確認を行いました。開発環境は以下の通りです。

- Windows 7（64bit 版）
- RT ミドルウェア：OpenRTM-aist-1.1.0-RELEASE（C++版、java 版）
- コンパイラ：Microsoft　Visual　C++ 2010　Express（SP1）
- Eclipse：Eclipse 3.8.1 + OpenRTM Eclipse tools 1.1.0-RC4
- CMake：CMake 2.8.8

1.3 開発したコンポーネント群

今回地図画像を処理する画像処理のコンポーネントとして、以下のコンポーネントを開発しました。

- エッジ検出コンポーネント(CVCanny)
- 画像二値化コンポーネント(CVImgThreshold)
- ハフ変換コンポーネント(CVHoughThransform)

・特徴点検出コンポーネント(CVFeatureDetector)
・輪郭検出コンポーネント(CVFindContorours)
・選択領域修復コンポーネント(CVInpaint)

また、地図モデルを生成するためのツールとして、以下のコンポーネント群を開発しました。

・複数画像ファイル送信コンポーネント(OpenPicts)
・輪郭情報データを用いた画像修復コンポーネント(imageInpaint)
・ラインマップ変換コンポーネント(convToLineMap)
・地図モデル生成管理コンポーネント(mapsManager)
・UI(User Interface）コンポーネント(controlPanel)

それぞれのコンポーネントの詳細については 2 章を、使用方法については 3 章を参照してください。

2. 画像処理コンポーネント群の説明

2.1 エッジ検出コンポーネント(CVCanny)

CVCanny は、InPort の srcImage から入力された画像に対して、コンフィギュレーションパラメータを用いてエッジ検出を行うコンポーネントです。OutPort の cannyImage からは、入力画像から抽出したエッジデータの画像が出力されます。各コンフィギュレーションパラメータの説明は以下の表を参照してください。

・InPort

名称	型	説明
srcImage	CameraImage	エッジ検出を行う画像データを取得するポート。

・OutPort

名称	型	説明
cannyImage	CameraImage	入力画像から抽出したエッジデータの画像を出力するためのポート。

・Configuration

名称	型	デフォルト値	説明
01_ImageView	string	OFF	画像表示の ON/OFF を選択するための変数。radio 型で宣言されており、制約条件は

			(ON,OFF)。
02_threshold1	double	50	ヒステリシスが存在する処理の一番目の閾値。 text 型で宣言されており、制約条件は x > 0。
03_threshold2	double	200	ヒステリシスが存在する処理の二番目の閾値。 text 型で宣言されており、制約条件は x > 0。
04_ apertureSize	int	3	オペレータのアパーチャサイズ。 text 型で宣言されており、制約条件は x > 0。
05_L2gradient	string	false	画像勾配の強度を求めるための精度を選択するための変数。 true：L2 ノルム。 false：L1 ノルム。 （L2 ノルムの方がより高い精度となる。） radio 型で宣言されており、制約条件は(true, false)。

2.2 画像二値化コンポーネント(CVImgThreshold)

CVImgThreshold は、InPort の srcImage から入力された画像に対して、コンフィギュレーションパラメータで与えられた値を用いて二値化処理を行い、OutPort の thresholdImage から作成した二値画像を出力するコンポーネント。

二値化処理アルゴリズムは、cvThreshold と cvAdaptiveThreshold を使うことができ、一つの画像に複数回の処理を行うことが可能です。

・InPort

名称	型	説明
srcImage	CameraImage	画像データを取得するポート。 この画像に対し、二値化処理を行う。

・OutPort

名称	型	説明
thresholdImg	CameraImage	二値化処理で作成された二値画像を出力するポート。

・Configuration

名称	型	デフォルト値	説明
01_ImageView	string	OFF	画像表示を選択するための変数。

			radio 型で宣言されており、制約条件は(ON,OFF)。
02_Algorithm	vector<string>	NON	二値化処理アルゴリズムを指定するための変数。 *同じ画像に対して複数回使用する場合に対応。 ordered_list 型で宣言されており、制約条件としては (NON,Binary,BinaryInv,Trunc,ToZero,ToZeroInv,Adaptive_MEAN_CCV,Adaptive_GAUSSIAN_C)。
03_ Parameter	vector<double>	0,0,255,0,0,255,0,0,255	cvThreshold()及び cvAdapticeThreshold()で用いる引数。 cvThreshold()での引数は[0]~[3]を cvAdapticThreshold()での引数は[0]~[5]を使用します。また、入力に不備がある場合はデフォルト値を用いて処理を行います。 ------引数の説明------ cvThreshold()の場合 threPara[0] – usedFlag 0 ならデフォルト値で実行。 1 なら引数を用いて実行。 threPara[1] – double threshold 処理の閾値として使用。 制約：0<=x<=255 デフォルト：0 threPara[2] – double maxValue threAlgo が BINARY と BINARY_INV の場合のみ使用する。二値化の MAX 値として使用。 制約：0<=x<=255 デフォルト：255 cvAdaptedThreshold()の場合

			threPara[0] – usedFlag 0 ならデフォルト値で実行。 1 なら引数を用いて実行。 threPara[1] – double maxValue 二値化の MAX 値として使用。 制約：0<=x<=255 デフォルト：255 threPara[2] – int threshold_type 閾値処理の種類を 0 か 1 かで選択する。 CV_THRESH_BINARY – 0 CV_THRESH_BINARY_INV – 1 制約：0 , 1 デフォルト：0 threPara[3] – int blockSize ピクセルに対する閾値を計算するために利用する、そのピクセルの近傍領域のサイズ(3, 5, 7 など) 制約：1 < x かつ奇数 デフォルト：3 threPara[4] – double param1 利用する手法にパラメータ デフォルト：10
04_ GaussianBlur	string	ON	GaussianBlur を行うかどうかを決めるための変数。 radio 型で宣言されており、制約条件は(ON, OFF)。
05_ GaussianSize	vector<short>	3,3	画像の平均化に用いるガウシアンカーネルサイズ。 text 型で宣言されており、制約条件は x[0] >= 0, x[1] >= 0 かつ奇数。
06_BitwiseNot	string	ON	Bitwise_not を行うかどうかを決めるための変数。 radio 型で宣言されており、制約条件は(ON, OFF)。

2.3 ハフ変換コンポーネント(CVHoughThransform)

CVHoughThransformは、InPortの thresholdImageから入力された二値画像データに対して、コンフィギュレーションパラメータで設定した変換メソッドや値を用いてハフ変換を行い、その結果を InPort の srcImage から取得した画像データに書き込んでウィンドウに表示するコンポーネントです。InPort の srcImage を、二値化する前の画像を出力するポートに接続することで、ハフ変換の結果を二値化する前の画像に描画することができるため、変換閾値などを確認することができます。OutPort の houghLines からは、ハフ変換の結果となる Lines を配列化したデータが出力されます。各コンフィギュレーションパラメータの説明は以下の表を参照してください。

・InPort

名称	型	説明
srcImage	CameraImage	二値化する前の画像を取得するポート。 この画像に対し、ハフ変換の結果を書き込む。
thresholdImage	CameraImage	二値画像を取得するポート。この画像を用いてハフ変換を行う。 この接続がない場合、InPort の srcImage の画像を固定値で二値化し、ハフ変換を行う。

・OutPort

名称	型	説明
houghLines	TimedShortSeq	ハフ変換によって得た線のデータを出力するポート。 線は line の端点として用いる Point p1,p2 の座標を - l1.p1.x [0] - l1.p1.y [1] - l1.p2.x [2] - l1.p2.y [3] として配列に格納したデータである。 [0]~[3]によって一本の線が表されている。

・Configuration

名称	型	デフォルト値	説明
01_ImageView	string	OFF	画像表示を選択するための変数。 radio 型で宣言されており、制約条件は(ON,OFF)。
02_HoughMethod	string	STANDARD	ハフ変換の method を選択するための変数。

			STANDARD：標準的ハフ変換 MULTI_SCALE ： マルチスケール型の古典的ハフ変換 PROBABILISTIC：確率的ハフ変換 radio 型で宣言されており、制約条件は (STANDARD, MULTI_SCALE, POBABILISTIC)。
03_rho	double	1	ピクセル単位で表される投票空間の距離分解能。 text 型で宣言されており、制約条件は x > 0。
04_theta	double	0.0174533 (CV_PI/180)	ラジアン単位で表される投票空間の角度分解能。 text 型で宣言されており、制約条件は x > 0。
05_threshold	int	100	投票の閾値。 十分な票（ >threshold ）を得た直線のみが出力される。 text 型で宣言されており、制約条件は x > 0。
06_srn	double	0	マルチスケールハフ変換において、 距離分解能 rho の除数となる値。 投票空間の粗い距離分解能は rho となり、細かい分解能は rho/srn となる。 srn=0 かつ stn=0 の場合は，古典的ハフ変換が利用され、 そうでない場合は、両方のパラメータが正値である必要がある。 text 型で宣言されており、 制約条件は x >= 0。
07_stn	double	0	マルチスケールハフ変換において、角度分解能 theta の除数となる値。 text 型で宣言されており、 制約条件は x >= 0。
08_minLineLength	double	30	最小の線分長。 これより短い線分は棄却される。 text 型で宣言されており、制約条件は x > 0。

09_maxLineGap	double	10	2 点が同一線分上にあると見なす場合に許容される最大距離。 text 型で宣言されており、制約条件は x > 0。

2.4 特徴点検出コンポーネント(CVFeatureDetector)

CVFeatureDetector は、InPort の srcImage から入力された画像に対して、コンフィギュレーションパラメータで選択されたメソッドやアダプタ、入力された閾値を用いて特徴点検出を行うコンポーネントです。OutPor の featurePoint からは、特徴点検出の結果となる KeyPoint を配列化したデータを出力します。各コンフィギュレーションパラメータの説明は以下の表を参照してください。

・InPort

名称	型	説明
srcImage	CameraImage	特徴点検出を行う画像データを取得するポート。

・OutPort

名称	型	説明
featurePoint	TimedFloatSeq	特徴点検出で得た KeyPoint のデータを出力するポート。 KeyPoint 型のデータを、以下のように分解して出力する。 KeyPoint - float angle [0] - int class_id [1] - int octave [2] - Point pt (pt.x [3] pt.y [4]) - float response [5] - float size [6] これら 7 個のデータがそれぞれ配列の一要素として用いられている。

・Configuration

名称	型	デフォルト値	説明
01_ImageView	string	OFF	画像表示を選択するための変数。 radio 型で宣言されており、制約条件は (ON,OFF)。
02_Method	string	FAST	特徴検出のメソッドを選択するための変

			数。 FAST：FAST()メソッドを用いた特徴検出。 Good：GoodFeaturesToTrack()クラスを用いた特徴検出。 GoodHarris : GoodFeaturesToTrack()クラスにおいて、HarrisDetector に true を与えた特徴検出。 Star ：StarDetector()クラスを用いた特徴検出。 SIFT：SIFT()クラスを用いた特徴検出。 SURF：SURF()クラスを用いた特徴検出。 MSER：MSER()クラスを用いた特徴検出。 radio 型で宣言されており、制約条件とは(FAST, Good, GoodHarris, Star, SIFT, SURF, MSER)。
03_Adapter	string	NON	特徴検出のアダプタを選択するための変数。 NON：アダプタを使用せず検出。 Grid：入力画像をグリッド状に分割して各セルでポイント検出。 Pyramid : ガウシアンピラミッドの複数レベルから検出。 Dynamic：指定した固定数の特徴が見つかるまで繰り返し検出。 radio 型で宣言されており、制約条件は(NON, Grid, Pyramid, Dynamic)。
04_FAST-threshold	int	1	FAST()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
05_Good-maxCorners	int	1000	GoodFeatureToTrack()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
06_Good-qualityLevel	double	0.01	GoodFeatureToTrack()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
07_Good-minDistan	double	1	GoodFeatureToTrack()のパラメータ。

ce			text 型で宣言されており、制約条件は x > 0。
08_Good-blockSize	int	3	GoodFeatureToTrack()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
09_Good-k	double	0.04	GoodFeatureToTrack()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
10_Star-maxSize	int	16	StarDetector()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
11_Star-responseThreshold	int	30	StarDetector()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
12_Star-lineThresholdProjected	int	10	StarDetector()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
13_Star-lineThresholdBinarized	int	8	StarDetector()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
14_Star-suppressNonmaxSize	int	5	StarDetector()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
15_SIFT-threshold	double	0.05	SIFT()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
16_SIFT-edgeThreshold	double	10.0	SIFT()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
17_SURF-hessianThreshold	double	400.0	SURF()のパラメータ。 text 型で宣言されており、制約条件は x > 0。
18_Grid-maxTotalKeypoints	int	200	画像から検出されるキーポイントの最大数。 text 型で宣言されており、制約条件は x > 0。
19_Grid-gridRows	int	10	グリッドの行数。 text 型で宣言されており、制約条件は x > 0。
20_Grid-gridCols	int	10	グリッドの列数。 text 型で宣言されており、制約条件は x > 0。
21_Pyramid-levels	int	3	スケーリングレベル。 text 型で宣言されており、制約条件は x > 0。
22_Dynamic_minFeatures	int	400	出力キーポイント個数の最小数。 text 型で宣言されており、制約条件は x > 0。
23_Dynamic_maxFeatures	int	500	出力キーポイント個数の最大数。 text 型で宣言されており、制約条件は x > 0。
24_Dynamic_maxIt	int	10	特徴検出処理の最大繰り返し回数。

ers	text 型で宣言されており、制約条件は x > 0。

2.5 輪郭検出コンポーネント(CVFindContours)

CVFindContours は、InPort の thresholdImage から入力された二値画像データに対して、コンフィギュレーションパラメータで設定した変換メソッドや値を用いて輪郭検出を行い、その結果を InPort の srcImage から取得した画像データに書き込んでウィンドウに表示するコンポーネントです。InPort の srcImage を、二値化する前の画像を出力するポートに接続することで、輪郭検出の結果を二値化する前の画像に描画することができるため、変換閾値などを確認することができます。OutPort からは輪郭検出の結果となる Contours を配列化したデータが出力され、contoursRectangles からは長方形化した輪郭データ、contoursConvex は凸図形化した輪郭データが出力されます。

・InPort

名称	型	説明
srcImage	CameraImage	二値化する前の画像を取得するポート。 この画像に対し、輪郭検出の結果を書き込む。
thresholdImage	CameraImage	二値画像を取得するポート。この画像を用いて輪郭検出を行う。 この接続がない場合、InPort の srcImage の画像を固定値で二値化し、輪郭検出を行う。

・OutPort

名称	型	説明
contoursRectangles	TimedShortSeq	輪郭検出によって得た輪郭点群からなる、長方形のデータを出力するポート。 長方形は Rectangle r1 の左上の点 p1 の座標と width,height を - r1.p1.x [0] - r1.p1.y [1] - r1.width [2] - r1.height [3] として配列に格納したデータである。 [0]~[3]によって一つの長方形が表されている。
contourConvex	TimedShortSeq	輪郭検出によって得た輪郭点群からなる、凸図形のデータを出力するポート。 凸図形は、convex1 として用いる vect<Point>のデータ

	である Point p1,p2,p3,p4 を - convex1.size() [0] - convex1.p1.x [1] - convex1.p1.y [2] - convex1.p2.x [3] - convex1.p2.y [4] - convex1.p3.x [5] - convex1.p3.y [6] - convex1.p4.x [7] - convex1.p4.y [8] として配列に格納したデータである。 [0]~[8]で一つの図形が表されている。 受け取り側は、[0]にあたるデータ数分の Point が来ると認識し、その Point 数分格納したら次の vector 処理へ進む。

・Configuration

名称	型	デフォルト値	説明
01_ImageView	string	OFF	画像表示を選択するための変数。 radio 型で宣言されており、制約条件は(ON,OFF)。
02_FindContourMode	string	EXTERNAL	輪郭検出のモードを選択するための変数。 CV_RETR_EXTERNAL：最も外側の輪郭のみを抽出する。 CV_RETR_LIST：すべての輪郭を抽出し、それらをリストに保存する。 CV_RETR_CCOMP : すべての輪郭を抽出し、それらを 2 階層構造として保存する。 *上のレベルには, 連結成分の外側の境界線が，下のレベルには，連結成分の内側に存在する穴の境界線が属する。 CV_RETR_TREE : すべての輪郭を抽出し、入れ子構造になった輪郭を完全に

			表現する階層構造を構成する。 radio 型で宣言されており、制約条件は(EXTERNAL,LIST,CCOMP,TREE)。
03_FindContourMethod	string	NONE	輪郭検出のメソッドを選択するための変数。 CV_CHAIN_APPROX_NONE：チェーンコードの全ての点を、通常の点群に変換する。 CV_CHAIN_APPROX_SIMPLE : 水平・垂直・斜めの線分を圧縮し、それらの端点のみを残す処理を行う。 CV_CHAIN_APPROX_TC89_L1：チェーン近似アルゴリズムを適用する。 CV_CHAIN_APPROX_TC89_KCOS - Teh-Chin：チェーン近似アルゴリズムを適用する。 CV_LINK_RUNS : 値が 1 のセグメントを水平方向に探索し、接続する輪郭を抽出するアルゴリズムを適用する。 *CV_LINK_RUNS は CVRETR_LIST が選択されていなければ使うことができないため、モードが LIST でない場合は、CODE を用いるように変更される。 radio 型で宣言されており、制約条件は(NONE,SIMPLE,TC89_L1,TC89_KCOS,LINK_RUNS)。
04_offset	vector <int>	0,0	オプションのオフセット。 各輪郭点はこの値の分だけシフトする。 text 型で宣言されている。

2.6 選択領域修復コンポーネント(CVInpaint)

CVInpaint は、入力された画像に対して、コンフィギュレーションパラメータで与えられた値をから得られる領域に対して修復処理を行い、修復された画像を OutPort の inpaintImage

から出力するコンポーネントです。

InPort は、修復処理の対象画像を取得する inpaintSrcImage と、修復マスク画像を取得する maskImage と、修復領域を示す長方形のデータを取得する maskRectangle の 3 つがあります。

・InPort

名称	型	説明
inpaintSrcImage	CameraImage	画像データを受け取るポート。 この画像対して画像修復処理が行われる。 修復の基になる領域は、InPort の maskImage、maskRectangle または コンフィギュレーションパラメータより選択する。
maskImage	CameraImage	修復マスク画像を受け取るポート。
maskRectangle	TimedShortSeq	修復領域を示す長方形のデータ。 左上の x 座標，左上の y 座標，横幅，縦幅を持つ配列。

・OutPort

名称	型	説明
inpaintImage	CameraImage	修復処理後の画像を出力するポート。

・Configuration

名称	型	デフォルト値	説明
01_ImageView	string	OFF	画像表示を選択するための変数。 radio 型で宣言されており、制約条件は (ON,OFF)。
02_InpaintSpace	vector<long>	0,0,0,0	修復したい領域をユーザが設定するた めのパラメータ。 左上の x 座標，左上の y 座標，横幅，縦 幅を持つ long の配列。 text 型で宣言されており、制約条件は x[0], x[1], x[2] >= 0, x[3] >= 0。
03_ InpaintRadius	double	0.0	修復される各点を中心とする近傍円形 領域の半径。 text 型で宣言されており、制約条件は x[0] >= 0, x[1] >= 0。
04_ InpaintFlags	string	INPAINT_NS	inpaint の修復手法。

	INPAINT_NS：ナビエ・ストークス (Navier-Stokes) ベース手法。 INPAINT_TELEA：Alexandru Telea による手法。（Telea04） radio 型で宣言されており、制約条件は (INPAINT_NS,INPAINT_TELEA)。

3. 地図モデル生成のためのツールコンポーネント群の説明

3.1 複数画像ファイル送信コンポーネント(OpenPicts)

OpenPicts は、読み込んだ画像ファイルを CameraImage 型に変換し OutPort の sendCameraImage から出力するコンポーネントです。読み込む画像ファイルは、コンフィギュレーションパラメータの FileName、又は入力ポート FileName 指定できます。また、InPort の backCameraImage は送り先からのフィードバック処理を受けるためのポート、OutPort の sendFlagData は画像の送信終了・送信のやり直しなどの状態を伝えるためのポートとなっており、フィードバック処理を模した構造となっています。これにより、送り先のコンポーネントと相互通信を模したやり取りを行うことができ、複数枚の画像を CameraImage 型で送ることができます。

フィードバック処理を適切に行うため、OutPort の sendCameraImage のデータ送信ポリシーは、New を選択することを推奨しています。

・InPort

名称	型	説明
FileName	TimedString	読み込む画像データのファイル名を取得するポート。 ","で区切ることで複数取得可能。
backCameraImage	CameraImage	送り先から受け取った画像データを取得するポート。 このポートの CameraImage と OutPor の sendCameraImage から出力される CameraImage が合致した場合、次画像へのフラグ処理へ進む。

・OutPort

名称	型	説明
sendCameraImage	CameraImage	読み込んだ画像データを出力するポート。 データ送信ポリシー：New 推奨。

sendFlagData	TimedShort	送り先へ現在の送信状態を出力するポート。 <フラグ処理の説明> -2:ファイルの元データが書き換わったことを示すフラグ（要初期化）。 -1:すべての画像を送信したことを示すフラグ。

・Configuration

名称	型	デフォルト値	説明
01_ImageView	string	OFF	画像表示を選択するための変数。 radio 型で宣言されており、制約条件は(ON,OFF)。
02_FileName	string	*.jpg	読み込みたいファイルのアドレス。 ","で区切ることで複数取得可能。 text 型で宣言されている。

3.2 輪郭情報を用いた画像修復コンポーネント(imageInpaint)

imageInpaint は、入力ポートから受け取った輪郭点データ群の情報を用いて画像修復を行うコンポーネントです。受け取った輪郭点データ群に対し、UI コンポーネント（controlPanel）を用いて削除等の操作を行うことができます。また、修復に対しても、UI コンポーネント（controlPanel）でのドラッグ入力を用いることで、処理に用いるマスクイメージを作り、修復処理を行うコンポーネント（CVInpaint）に出力することができます。これにより、容易に指定領域の画像修復を行うことができます。

InPort には修復処理を行いたい画像データを取得する srcImage、輪郭点群がベースとなっている輪郭長方形データ群を取得する contoursRectangles、同じく輪郭点群がベースとなっている輪郭凸図形データ群を取得する contoursConvex の 3 つのポートに加え、UI コンポーネント（controlPanel）のクリック情報を取得する clickPoint、ドラッグによって生成された長方形情報を取得する draggedRect、画像情報送信に用いる一時保存領域を指定する tempFolderPath、修復処理の実行結果となる画像を取得する inpaintImage の計 7 つがあります。contoursRectangles と contoursConvex は輪郭検出コンポーネント（CVFindContour）と、clickPoint と draggedRect は UI コンポーネント（controlPanel）コンポーネントとの接続が前提とされています。

OutPort は、UI コンポーネントへ表示する画像及び画像の保存アドレスを出力する modifyImage、modifyImagePath、選択領域修復コンポーネント（CVInpaint）へ修復に用いる情報を送るための inpaintSrcImage、inpaintMaskImage、 inpaintMaskArea、全ての処理が終わった結果の画像を送るための processedImage の計 6 つがあります。

・InPort

名称	型	説明
srcImage	CameraImage	修復処理を行う画像データを受け取るポート。 この画像は修復処理以外にも、UI コンポーネント(controlPanel)上に表示される画像としても用いる。
contoursRectangles	TimedShortSeq	輪郭検出によって得た輪郭点群からなる長方形のデータを取得するポート。 長方形は Rectangle r1 の左上の点 p1 の座標と width,height を - r1.p1.x [0] - r1.p1.y [1] - r1.width [2] - r1.height [3] として配列に格納したデータである。 [0]~[3]によって一つの長方形が表されている。
contoursConvex	TimedShortSeq	輪郭検出によって得た輪郭点群からなる凸図形のデータを出力するポート。 凸図形は convex1 として用いる vect<Point>のデータである Point p1,p2,p3,p4 を - convex1.size() [0] - convex1.p1.x [1] - convex1.p1.y [2] - convex1.p2.x [3] - convex1.p2.y [4] - convex1.p3.x [5] - convex1.p3.y [6] - convex1.p4.x [7] - convex1.p4.y [8] として配列に格納したデータである。 [0]~[8]で一つの図形が表されている。 受け取り側は、[0]にあたるデータ数分のPointが来ると認識し、そのPoint数分格納したら次のvector処理へ進む。
tempFolderPath	TimedString	大きい画像データを送受信する場合に用いる画像パスの保管場所。
clickPoint	TimedPoint3D	UI コンポーネント(controlPanel)上でクリックされた

		座標を取得するポート。 データは point.x：controlPanel 上の x 座標 point.y：controlPanel 上の y 座標 point.z：操作に対するフラグ情報 を意味する。
draggedRect	TimedShortSeq	UI コンポーネント(controlPanel)上でドラッグ指定した長方形情報を取得するポート。 データは左上の x 座標、左上の y 座標、横幅,縦幅を持つ配列となっている。
inpaintImage	CameraImage	修復処理が行われた画像を受け取るポート。 CVInpaint の OutPort に繋がることを前提としている。

・OutPort

名称	型	説明
modifyImage	CameraImage	ユーザの修正処理のために UI コンポーネント(controlPanel)上で表示する画像を出力するポート。 抽出領域が書き込まれた画像や修復処理が終わった後の画像が出力される。
modifyImagePath	TimedString	ユーザの修正処理のために UI コンポーネント(controlPanel)上で表示する画像のパスを出力するポート。
inpaintSrcImage	CameraImage	修復処理に用いる画像を出力するポート。 CVInpaint の InPort: : inpaintSrcImage に繋がることを前提としている。
inpaintMaskImage	CameraImage	修復処理に用いるマスク画像を出力するポート。 CVInpaint の InPort: : maskImage に繋がることを前提としている。 （同タイミングで長方形座標データを送った場合こちらからの情報が優先して処理される）
inpaintMaskArea	TimedShortSeq	マスク画像作成に用いる座標群を出力するポート。 CVInpaint の InPort: : maskRectangle に繋がることを前提としている。
processedImage	CameraImage	すべての処理が終わった画像を出力するポート。

・Configuration

名称	型	デフォルト値	説明
01_contoursType	string	Rectangle	処理に用いる輪郭データタイプを選択する。 長方形と凸図形の両方の形式のデータをポートから受け取っている場合に用いる。 Rectangle：長方形データ型の輪郭データ。 Convex：凸図形データ型の輪郭データ。 radio 型で宣言されており、制約条件は(Rectangle, Convex)。
02_MaskDataSelect	string	Image	CVInpaint へ送るマスクの情報を選択する。 Image：マスク画像を作成して inpaintMaskImage から CVInpaint へ出力する。 Rectangle：マスク領域を inpaintMaskArea から CVInpaint へ出力する。 radio 型で宣言されており、制約条件は(Image, Rectangle)。
03_MaskTypeSelect	double	Single	02_MaskDataSelect で maskImage を送る場合、全輪郭点データを一度に処理するかどうかを選択する。 Single：単体のデータを送る All：すべてのデータを送る radio 型で宣言されており、制約条件は(Single, All)。

3.3 ラインマップ変換コンポーネント(convToLineMap)

convToLineMap は、InPort から得られるデータ群を用いて、ラインマップに変換し、OutPort の compLineMap から出力するコンポーネントです。ここでのラインマップは「地図の頂点を示す座標群と、その座標同士の関係性」を持つ点と線の情報群を指します。

InPort の contoursConvex、clickPoint、draggedRect、featurePoints からは、それぞれ輪郭凸図形データ群、クリックされた座標データ、ドラッグ指定した長方形情報、特徴点検出で得たデータを取得し、これらを基にラインマップへの変換を行います。

また、UI コンポーネント(controlPanel）上に OutPort の modifyImage と modifyImagePath から出力した画像を表示し、それを見て修正を行うことができます。

・InPort

名称	型	説明
tempFolderPath	TimedString	画像情報送信に用いる一時作業領域のパスを受け取るためのポート。
srcImage	CameraImage	修復処理を行う画像データを受け取るポート。 この画像に対し修復処理が行われる。 また、UI コンポーネント(controlPanel）上で表示される画像もこの画像が用いられる。
contoursConvex	TimedShortSeq	輪郭点群がベースとなっている輪郭凸図形データ群を取得するポート。 凸図形は convex1 として用いる vect<Point>のデータである Point p1,p2,p3,p4 を - convex1.size() [0] - convex1.p1.x [1] - convex1.p1.y [2] - convex1.p2.x [3] - convex1.p2.y [4] - convex1.p3.x [5] - convex1.p3.y [6] - convex1.p4.x [7] - convex1.p4.y [8] として配列に格納したデータである。 [0]~[8]で一つの図形が表されている。 受け取り側は、[0]にあたるデータ数分の Point が来ると認識し、その Point 数分格納したら次の vector 処理へ進む。
clickPoint	TimedPoint3D	UI コンポーネント(controlPanel)上でクリックされた座標を取得するポート。 データは point.x：controlPanel 上の x 座標。 point.y：controlPanel 上の y 座標。 point.z：操作に対するフラグ情報。 を意味する。

draggedRect	TimedShortSeq	UI コンポーネント(controlPanel)上でドラッグ指定した長方形情報を取得するポート。 データは左上の x 座標、左上の y 座標、横幅,縦幅を持つ配列となっている。
featurePoints	TimedFloatSeq	特徴点検出で得た KeyPoint のデータを取得するポート。 KeyPoint 型のデータを、以下のように分解して出力する。 KeyPoint - float angle [0] - int class_id [1] - int octave [2] - Point pt (pt.x [3] pt.y [4]) - float response [5] - float size [6] これら 7 個のデータがそれぞれ配列の一要素として用いられている。

・OutPort

名称	型	説明
modifyImage	CameraImage	UI コンポーネント(controlPanel)上で表示する画像を出力するポート。 抽出領域を書き込んだ画像や修復処理後の画像を出力する。
modifyImagePath	TimedString	UI コンポーネント(controlPanel)上で表示する画像（modifyImage）のパスを出力するポート。
compLineMap	TimedShortSeq	変換が完了した地図のライン情報を取得するポート。 ライン情報は convex1 として用いる vect<Point>のデータである Point p1,p2,p3,p4 を - convex1.size() [0] - convex1.p1.x [1] - convex1.p1.y [2] - convex1.p2.x [3] - convex1.p2.y [4] - convex1.p3.x [5] - convex1.p3.y [6] - convex1.p4.x [7]

	- convex1.p4.y [8] として配列に格納したデータである。 [0]~[8]で一つの図形が表されている。 受け取り側は、[0]にあたるデータ数分の Point が来ると認識し、その Point 数分格納したら次の vector 処理へ進む。

・Configuration

名称	型	デフォルト値	説明
modifyType	string	erase	輪郭データ群に対するユーザからの操作の種類を選択する。 erase : クリックされた一番近い凸図形頂点を、輪郭データ群から消す add：クリック操作によって隣り合う凸図形頂点を選択してもらい、その二つの頂点の間に新たに頂点を追加する追加する頂点は特徴点の中から選択する radio 型で宣言されており、制約条件は(erase,add)。

3.4 地図モデル生成管理コンポーネント(mapsManager)

mapsManager は、地図画像から地図モデルへの変換を管理し、結果の保存を行うコンポーネントです。

何枚目の画像に対し地図モデル生成を行うか、処理アルゴリズムが無事完了したかなど、UI コンポーネント(controlPanel)を介してユーザの指示を受け付けることができます。

複数画像ファイル送信コンポーネント(OpenPicts)から画像を取得し、1 枚目から順に地図モデル化処理を行う。OutPort の backCamImg から出力した画像と処理が完了した画像の TimeStamp を比較し、同一なら UI コンポーネント(controlPanel）上に処理前と処理後の画像を表示する。InPort の receiveFlagData で UI コンポーネント(controlPanel）からの完了指示を取得したら来たら次の画像、もう一度の指令が受け渡されたら、もう一度同じ画像に処理を行う。最終処理でラインマップが完成したと判断されたら、コンフィギュレーションパラメータの 03_SaveSpace で指定された場所へ OutPort の lineMapsPath から全座標群情報を、compLineImg から描画イメージを出力する。

・InPort

名称	型	説明
receiveCamImg	CameraImage	地図モデル生成に用いる画像データを取得するポート。
receiveFlagData	TimedShort	送り先コンポーネントの画像データ送信状況を取得するポート。 <フラグ処理の説明> -2：受け取り状態の初期化のフラグ。 -1：全画像データ送信済みのフラグ。
tempFloderPath	TimedString	一時作業領域のパスを取得するポート。 大きい画像データを入出力するときに用いる。
clickPoint	TimedPoint3D	UI コンポーネント(controlPanel)上でクリックされた座標を取得するポート。 データは point.x：controlPanel 上の x 座標。 point.y：controlPanel 上の y 座標。 point.z：操作に対するフラグ情報。 を意味する。
compNormalizedMap	CameraImage	正規化された地図画像データを取得するポート。
compLineMap	TimedShortSeq	変換されたラインマップの座標群を取得するポート。 ラインマップを描画する際に用いる。 データは vector<vector<Point>>型の地図座標群で、[0][1][2][3][4]… に対し[vector<Point>サイズ n*2][p1.x][p1.y][p2.x][p2.y]…[pn.x][pn.y][vector<Point>のサイズ m*2]… という要素を持つ。

・OutPort

名称	型	説明
backCamImg	CameraImage	送り先から受け取った地図画像を出力するポート。 フィードバッグ処理を疑似的に再現するために用いられ、送り先側でデータの一致を確認することで次画像が送られる。
tempFolderPathOut	TimedString	一時作業領域のパスを出力するポート。 大きい画像データを入出力するときに用いる。
makeNormSrcImg	CameraImage	正規化処理を行う画像データを出力するポート。

makeNormSrcImgPath	TimedString	正規化処理を行う画像データのパスを出力するポート。
compNormImg	CameraImage	正規化処理が完了した画像データを出力するポート。
compNormImgPath	TimedString	正規化処理が完了した画像データのパスを出力するポート。
makeLineSrcImg	CameraImage	ラインマップに変換する正規化処理が完了した画像を出力するポート。
makeLineSrcImgPath	TimedString	ラインマップに変換する正規化処理が完了した画像のパスを出力するポート。
compLineImg	CameraImage	ラインマップを描画した画像を出力するポート。
compLineImgPath	TimedString	ラインマップを描画した画像のパスを出力するポート。
lineMaps	TimedShortSeq	全地図画像のラインマップ情報を出力するポート。 全地図画像のラインマップは vector<vector<vector<Point>> map 型で、これらを一括で出力するために - map1.size() [0] - map1.floor1.size() [1] - map1.floor1.conv1.size() [2] - map1.floor1.conv1.p1.x [3] ・・・ - map1.floor1.conv1.p3.y [8] - map1.floor1.conv2.size() [9] - map1.floor1.conv2.p1.x [10] ・・・ - map1.floor1.conv2.p3.y [15] - map1.floor2.size() [16] - map1.floor2.conv1.size() [17] - map1.floor2.conv1.p1.x [18] ・・・ - map1.floor2.conv2.p3.y [30] のように 30 の要素を配列に格納して出力する。 [0]~[15]で一つのラインマップを表している。
lineMapsPath	TimedString	全地図画像のラインマップ情報のパスを出力するポート。

・Configuration

名称	型	デフォルト値	説明
01_ImageView	string	OFF	画像表示を選択するための変数。 radio 型で宣言されており、制約条件は(ON,OFF)。
02_SelectPictNum	int	0	画像処理及び線画化処理を行う画像の番号を指定できる。 先のステップから戻ることも可能。 text 型で宣言されており、制約条件は x>=0 かつ x<=最大枚数。
03_SaveSpace	string	C:¥¥tmp¥¥maps¥¥linemaps	完成したラインマップの座標情報群と描画画像を保存する先を指定する。 text 型で宣言されている。

3.5 UI(User Interface）コンポーネント(controlPanel)

controlPanel は、地図モデル生成中に必要となるユーザからの操作を取得するコンポーネントです。InPort の srcImage、srcImagePath から処理前の画像を取得、modifyImage、modifyImagePath から処理中の画像を取得、completeImage、completeImagePath から処理が完了した画像を取得します。ここで受け取った処理前、処理中、処理後の画像をウィンドウに表示します。

また、このウィンドウではユーザからの操作を取得することができ、所要機能としては、読み込み画像の指定・一時作業領域の指定・クリック情報の取得・ドラッグ情報の取得・ボタンによるイベント情報(決定操作やリセット操作など)の取得（情報形態はクリック情報と同一）があります。処理の状態ににあったイベントを提示し、その選択情報を OutPort から出力します。

・InPort

名称	型	説明
srcImage	CameraImage	処理前の画像を取得するポート。
srcImagePath	TimedString	処理前の画像のパスを取得するポート。
modifyImage	CameraImage	ユーザの操作を必要とする処理中の画像を取得するポート。
modifyImagePath	TimedString	ユーザの操作を必要とする処理中の画像のパスを取得するポート。
completeImage	CameraImage	処理が完了した画像を取得するポート。

completeImagePath	TimedString	処理が完了した画像のパスを取得するポート。

・OutPort

名称	型	説明
openFilePath	TimedString	生成されたウィンドウ上のメニューで指定した、読み込む画像のパスを出力するポート。
tempFolderPath	TimedString	生成されたウィンドウ上のメニューで指定した、一時作業領域のパスを出力するポート。
clickPoint	TimedPoint3D	生成されたウィンドウ上でクリックした座標のデータを出力するポート。
dragRectangle	TimedShortSeq	生成されたウィンドウ上でドラッグされた領域のデータを出力するポート。

4. コンポーネント群の使用方法

4.1 RTSystemeEditor を用いたシステム構築の手順

RTSystemeEditor を用いたシステム構築は、通常以下の手順で行われます。

1. ネームサーバを起動する
2. RTSystemEditor を起動する
3. ネームサーバへ接続する
4. コンポーネントを起動する
5. システムを構築し、実行する

これらの手順はどのようなシステムでも共通です。RTSystemEditor の詳しい操作方法は OpenRTM-aist のホームページ（http://www.openrtm.org/）を参照してください。以下の節では手順 5 に関して説明します。

4.2 システムの構築と実行手順

本コンポーネント群は、一般的な地図画像すべてに対応することを目的として作られているため、固定のコンポーネント間の接続という概念は存在していません。そのため、今回は地図モデル生成の一例として、デパートの一つを地図モデル化する手順を紹介いたします。

今回のシステムの大まかな流れは以下のようになります。

① 地図の画像を読み込む

② 地図画像からモデル生成に余計な情報を削除し正規化する

③ 正規化地図画像から地図の線情報を取得する

以下の章から各流れの具体的説明へと移ります。

4.2.1 ① 地図の画像を読み込む

使用するコンポーネント

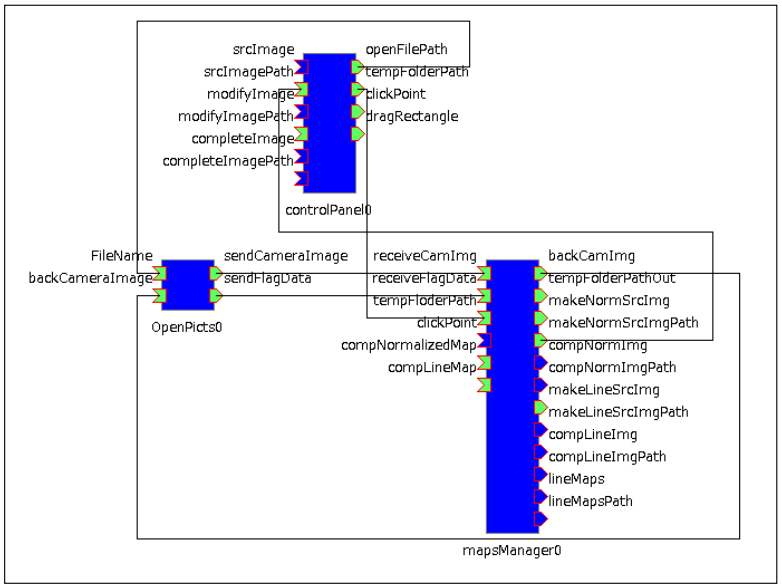
・controlPanel

・OpenPict

・mapManager

コンポーネント間の接続

InPort		OutPort	
コンポーネント名	Port 名	コンポーネント名	Port 名
OpenPicts	FileName	controlPanel	openFilePath
OpenPicts	backCameraImage	mapsManager	backCamImg
mapsManager	receiveCamImg	OpenPicts	sendCameraImage
mapsManager	receiveFlagData	OpenPicts	sendFlagData
mapsManager	tempFolderPath	controlPanel	tempFolderPath
controlPanel	srcImagePath	mapsManager	makeNormSrcImgPath



4.2.1 の接続図

すべてアクティブにすると、controlPanel の UserInterface ウィンドウが立ち上がります。このウィンドウのメニューバーにある File(F) -> FileOpen(O)を選択すると、ファイルマネージャが立ち上がるので、適当な地図画像を選択し(複数選択可)画像を読み込んでください。読み込まれた画像は、MapManager の makeNormSrcImg と makeNormSrcImgPath から出力されます。

4.2.2 ② 地図画像からモデル生成に余計な情報を削除し正規化する

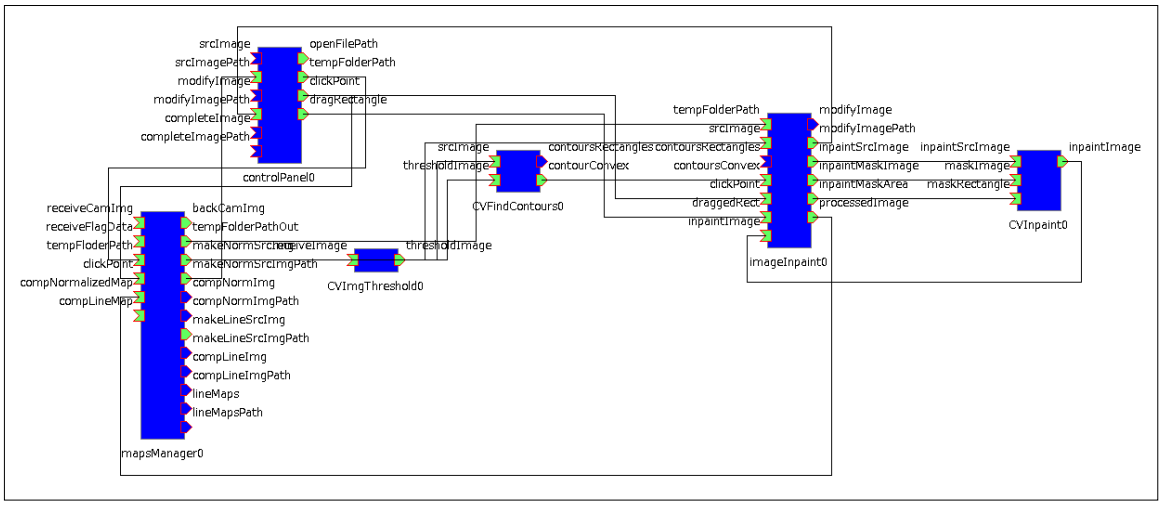
使用するコンポーネント

・controlPanel
・mapManager
・imageInpaint
・CVImgThreshold
・CVFindContours
・CVInpaint

コンポーネント間の接続

InPort		OutPort	
コンポーネント名	Port 名	コンポーネント名	Port 名
mapsManager	tempFolderPath	controlPanel	tempFolderPath
mapsManager	clickPoint	controlPanel	clickPoint
mapsManager	compNormalizedMap	imageInpaint	processedImage
controlPanel	modifyImagePath	imageInpaint	modifyImagePath
controlPanel	srcImagePath	mapsManager	makeNormSrcImgPath
CVImgThreshold	receiveImage	mapsManager	makeNormSrcImg
CVFindContours	srcImage	mapsManager	makeNormSrcImg
CVFindContours	thresholdImage	CVImgThreshold	thresholdImage
CVInpaint	inpaintSrcImage	mapsManager	inpaintSrcImage
CVInpaint	maskImage	mapsManager	inpaintMaskImage
CVInpaint	maskRectangle	mapsManager	inpaintMaskArea
imageInpaint	tempFolderPath	mapsManager	tempFolderPath
imageInpaint	srcImage	mapsManager	makeNormSrcImg
imageInpaint	contoursRectangles	CVFindContours	contoursRectangles
imageInpaint	contoursConvex	CVFindContours	contoursConvex
imageInpaint	clickPoint	controlPanel	clickPoint

imageInpaint	draggedRect	controlPanel	dragRectangle
imageInapint	inpaintImage	CVInapint	inpaintImage



4.2.2 の接続図

処理が始まると、controlPanel のウィンドウにグレースケールの地図画像に輪郭データが上書きされた図が表示されます(Fig.4.2.2.1)。この輪郭データはクリック動作やドラッグ操作で消すことが出来るため、「削除したくない部分」はこの処理で取り除いてください。また、輪郭データが望む形で得られない場合は、CVImgThreshold の閾値設定で調整を行ってください(Fig.4.2.2.2)。修正処理が完了したら、左上の「修正完了」を選択してください。囲われていた輪郭データの部分が、すべて修復処理されます。自動での修復処理が完了すると、今度は、ユーザがマウスのドラッグ操作で修復領域を選択する行程に移ります。この処理で、輪郭検出などで得ることが出来なかった部位の修復を行ってください。この修復修正作業が完了したら「修正完了」を押してください。これら一連の作業が完了すると、上のボタンで、「次画像へ」や「再処理」などが選択できるようなりますので、お好みで選択してください。

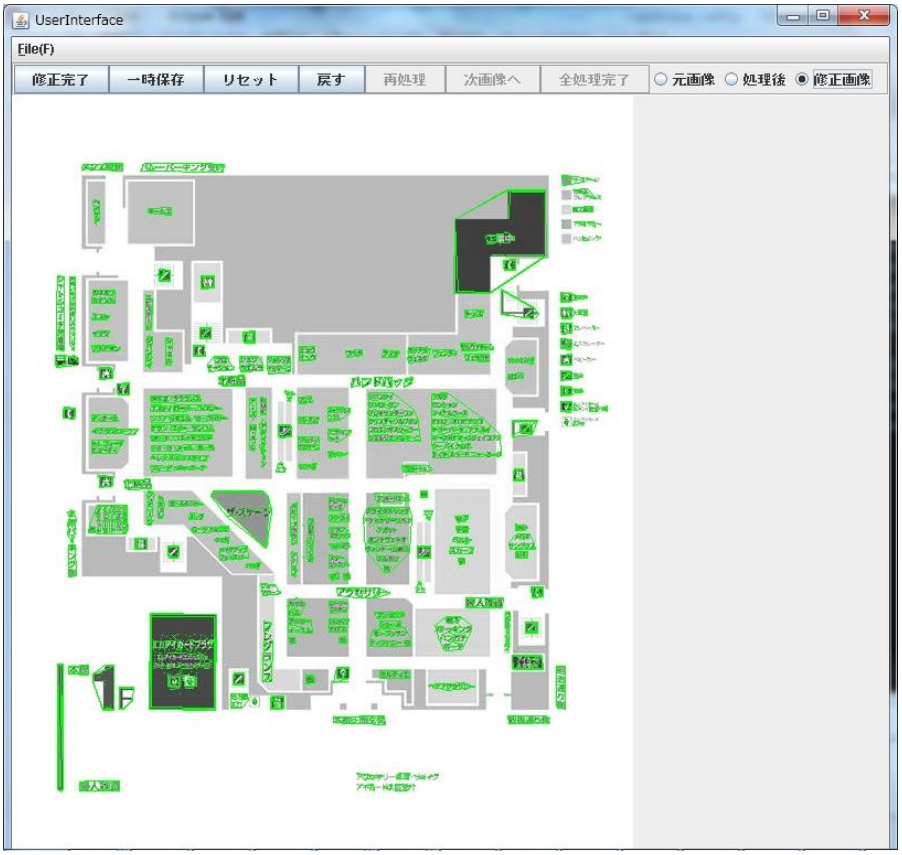


Fig.4.2.2.1

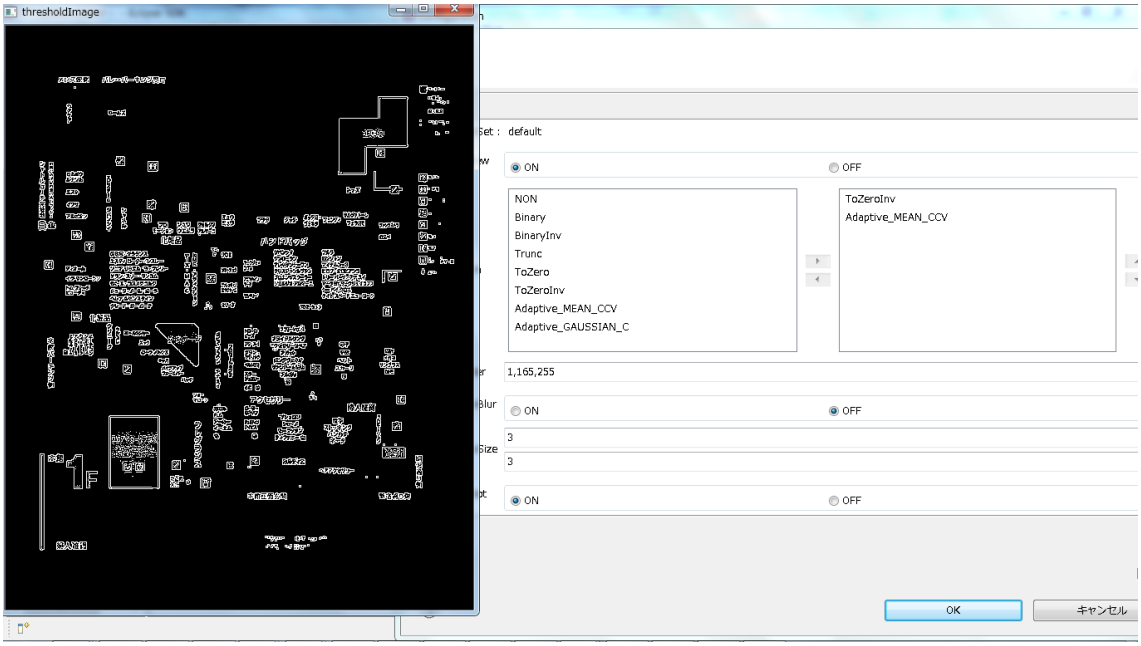


Fig.4.2.2.2

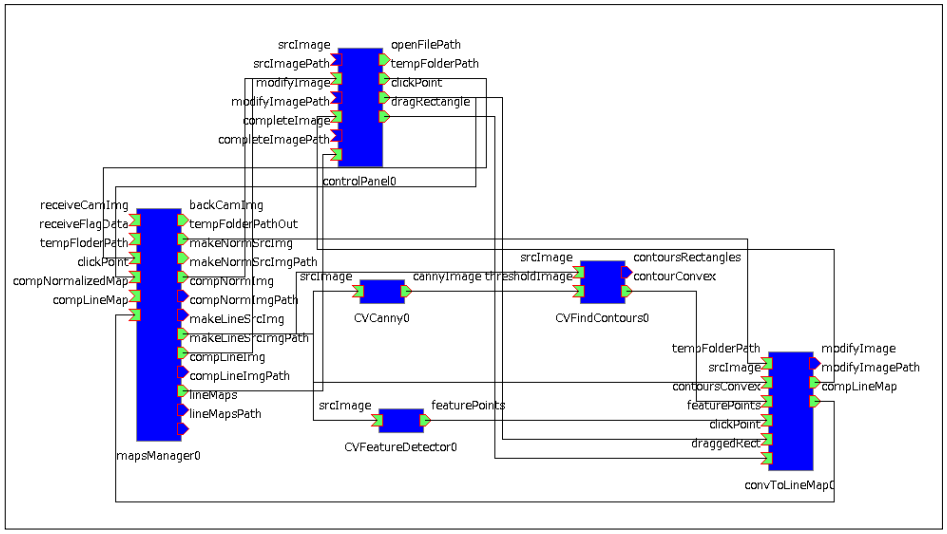
4.2.3 ③ 正規化地図画像から地図の線情報を取得しラインマップへ変換する

使用するコンポーネント

・controlPanel
・mapManager
・convToLineMap
・CVCanny
・CVFindContours
・CVFeatureDetector

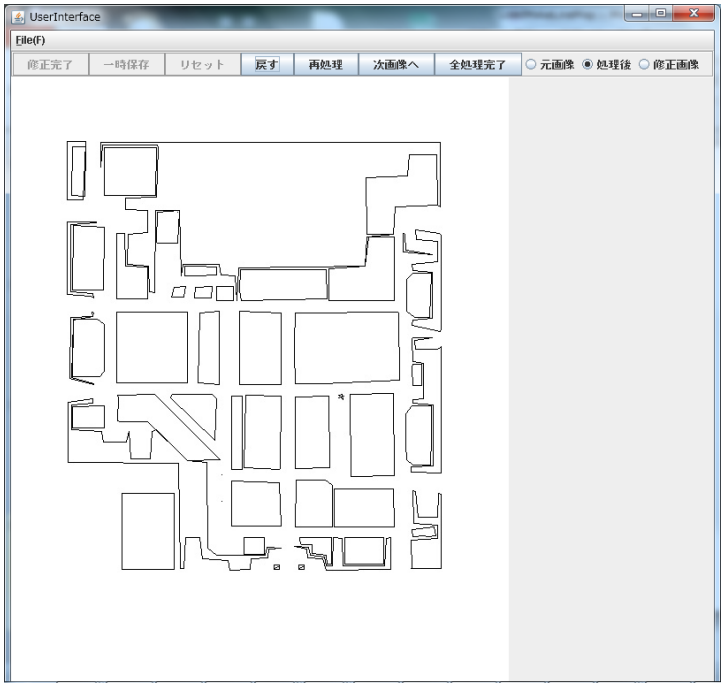
コンポーネント間の接続

InPort		OutPort	
コンポーネント名	Port 名	コンポーネント名	Port 名
mapsManager	tempFolderPath	controlPanel	tempFolderPath
mapsManager	clickPoint	controlPanel	clickPoint
mapsManager	compLineMap	convToLineMap	compLineMap
CVFeatureDetector	srcImage	mapsManager	makeLineSrcImg
CVCanny	srcImage	mapsManager	makeLineSrcImg
CVFindContours	srcImage	mapsManager	makeLineSrcImg
CVFindContours	thresholdImage	CVCanny	cannyImage
convToLineMap	tempFolderPath	mapsManager	tempFolderPath
convToLineMap	srcImage	mapsManager	makeLineSrcImg
convToLineMap	contoursConvex	CVFindContours	contoursConvex
convToLineMap	featurePoints	CVFeatureDetector	featurePoints
convToLineMap	clickPoint	controlPanel	clickPoint
convToLineMap	draggedRect	controlPanel	draggedRect
controlPanel	modifyImage	convToLineMap	modifyImagePath
controlPanel	completeImagePath	mapsManager	compLineImgPath



4.2.2 の接続図

処理が始まると、ラインマップのもととなる輪郭データが画像に出力されます。まず、余分な輪郭データを、マウスのクリックとドラッグ動作で削除してください。必要な削除がお済みになりましたら、左上の「修正完了」を押してください。その後、輪郭データの頂点に対する修正の動作に移ります。convToLineMaps のコンフィギュレーションパラメータで erase と add を選択することができます。erase の場合はクリックされた頂点を削除し、add の場合はクリックされた頂点と次にクリックされた頂点の間に選択した特徴点を追加します。



ラインマップの完成図

5. お問い合わせ

本コンポーネント群につきましては、フィードバックに対し随時修正・開発・公開をしているため、まだまだ展開・改善の余地があるものと考えております。提案・要望・バグ報告・マニュアル記述の不備等に関しましては下記までご連絡ください。

【問い合わせ先】

〒108-8548

東京都港区芝浦 3-9-14,611

芝浦工業大学大学院電気電子情報工学専攻

ロボティクスシステムデザイン研究室

立川　将

Email:y09148@shibaura-it.ac.jp