

画像情報研究室 個別ミーティング報告資料

氏名： 釜坂 一歩

報告日： 2015 年 5 月 19 日

これまでの経緯 (今週の作業の背景)

- ・樽見さんプログラム引継ぎ

今週取り組んだこと

- ・樽見さんプログラム移植
 - アルゴリズムについては完了
 - 訓練画像軍のデータベース化
 - 毎回同じ訓練画像軍に対してキーポイントと特徴量を取得するのは時間がかかるため、キーポイントをテキストファイル, 特徴量をバイナリファイルに書き出す.
 - 各種ファイルが存在する場合はそちらを読み込むように変更.
- ・プログラム実行テスト
 - query, train 共に 100 フレーム刻みの 30 枚の画像を使用する
 - 結果

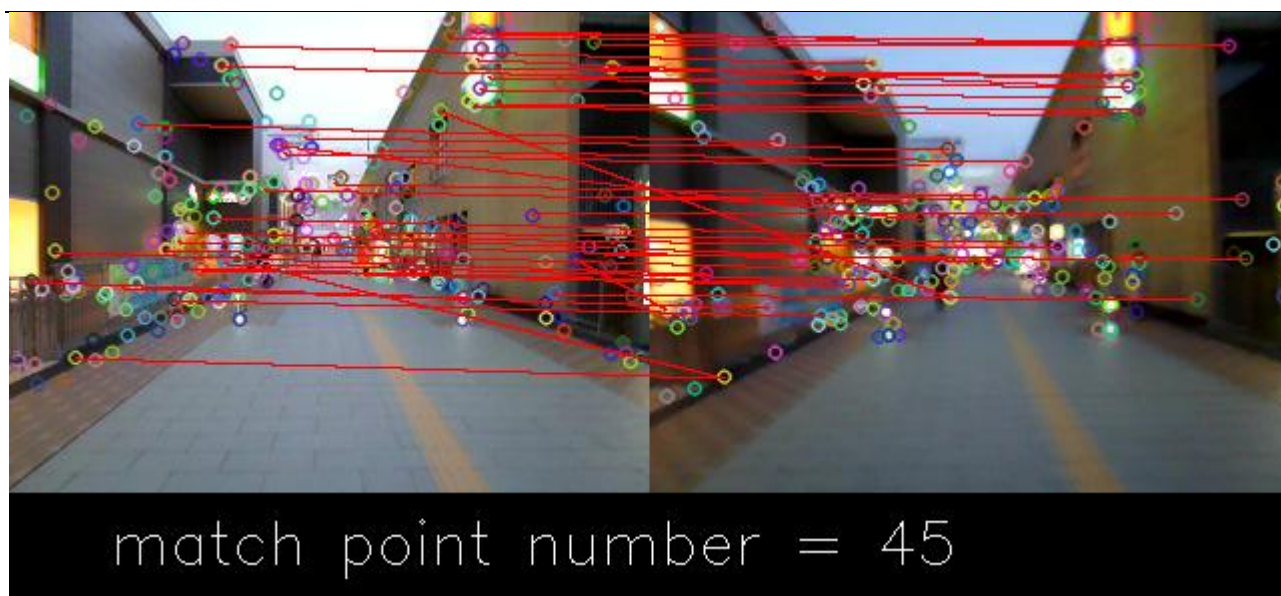


Figure 1 第1フレームのマッチング結果

- 同一地点のマッチングができていない点とできていない点が混在する.
- マッチングが正常でない点の除去を考える.
- ← キーポイントの距離を調べてみて閾値処理を行う.
- ← 使用するキーポイントを全訓練画像群から選別する
 - 特徴量をクラスタリングし, フレーム数の分散が大きいクラスに含まれるキーを除外する.
- ← クロスチェックを行う.

Table 1 query と train の対応フレームとマッチ数

query	train	マッチ数
100	100	45
200	200	48
300	300	27
400	400	51
500	500	28
600	700	31
700	800	33
800	900	35
900	1000	42
1000	1100	43
1100	1200	46
1200	1200	43
1300	1400	40
1400	1600	49
1500	1600	69
1600	1700	44
1700	1700	57
1800	1900	50
1900	2000	48
2000	2100	64
2100	2200	80
2200	2300	143
2300	2300	36
2400	2300	43
2500	2600	45
2600	2800	61
2700	2800	45
2800	2900	66
2900	2900	47
3000	3000	20

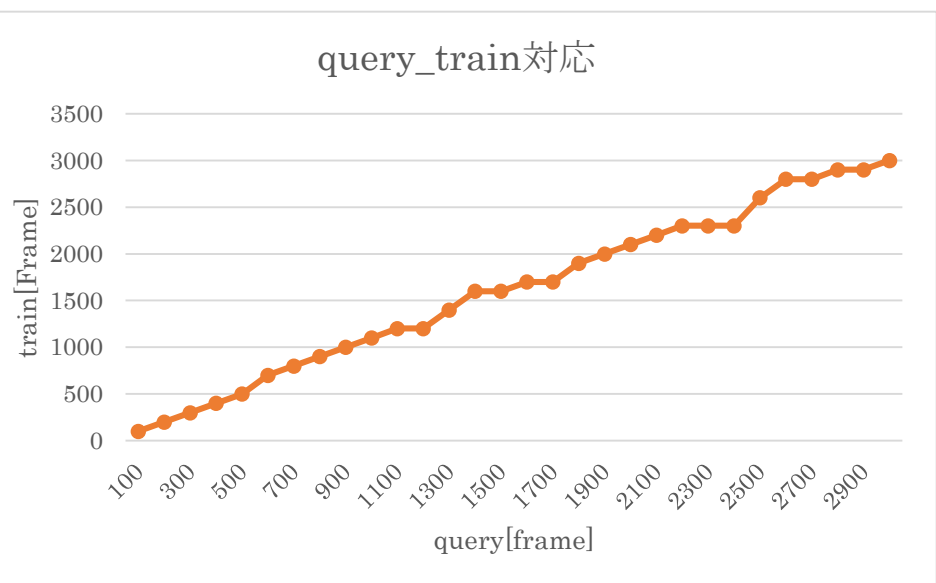


Figure 2 フレーム対応グラフ

→検索結果が線形となっているため,大雑把な検索としてはある程度の精度がみられる.

ー実行速度の検証

←ノート PC の Linux 環境でテスト(corei7,8GB)

→cv2.CV_AA で AttributeError

←int 型の数字(4)を代わりの引数とすることで解決

ーquery3270 枚,train3450 枚のデータセットに対して,データベースの作成と画像の出力を含めて
1172.877 秒(20 分弱)

←元プログラムはグラフ作成用のファイル I/O がボトルネックだった?(未検証)

ーデータベースを読み込んだ場合キーポイントの読み込みで配列外参照のエラー

今週の疑問点

- 研究の背景について
 - GPS を用いないナビゲーションシステム
 - 一般的な DGPS による推定誤差の大きさ数メートルでは誤差が大きい
 - 室内外を問わない
 - ←これらのニーズは？
 - ←異なる地点での類似画像(例では室内の廊下)に弱い

今週の問題点

- SIFT 以外の特徴量に関して,FAST での実行を試みたが,Python ではうまくいかなかった(プログラム停止)
 - FeatureDetector_create()にパラメータを渡すだけではダメ？
 - ←opencv2 系の FeatureDetector_create()で使えるのは SIFT,SURF,BRIEF,BRISK,FREAK
 - SIFT,SURF 以外で実行できない
 - キーポイントの検出時点で失敗している模様
 - ←SIFT で十分な速度と精度が得られれば変更の必要はない？
 - AKAZE に関しては python-opencv3.0 のドキュメントが充実したら試す

次週までの予定 (打合せの内容を反映して策定)

- 所信表明の準備(5/27)
-