# 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る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



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.

　－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

　－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)

　－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()で使用できるのは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)