**Image Registration Algorithms and Companies (5) (Dennis and Chris)**

[**http://stackoverflow.com/questions/11172408/surf-vs-sift-is-surf-really-faster**](http://stackoverflow.com/questions/11172408/surf-vs-sift-is-surf-really-faster) **// good link**

1. **(Dennis) Matlab? (**[**https://sites.google.com/site/myronenko/research/cpd**](https://sites.google.com/site/myronenko/research/cpd)**)**

* **UNUSABLE NOT OPEN SOURCE UNLESS COMPATIBLE WITH FREEMAT, ETC. (OPEN SOURCE MATLAB EQUIVALENT)**
* MatLab Computer Vision System Toolbox algorithm
* Enables automatic image registration.
* Developed by MathWorks company
* Team/researcher: ??

1. **(Dennis) OpenCV library (**[**http://docs.opencv.org/3.1.0/db/d61/group\_\_reg.html**](http://docs.opencv.org/3.1.0/db/d61/group__reg.html)**)**

* Image Registration Library
* Uses pixel based registration as opposed to feature based registration
* Feature based better under varying lighting coniditions or only partial image overlap, perhaps better for our applications
* Map class which creates a map between two (or more) images

1. **(Dennis) SURF (**[**http://www.sciencedirect.com/science/article/pii/S1077314207001555**](http://www.sciencedirect.com/science/article/pii/S1077314207001555)**)**

* **Speeded-Up Robust Features**
* **First presented by Herbert Bay in 2006**
* Faster than SIFT but OpenCv doesn’t have best implementation for speed/stability
* Algorithm not usable at peak rates with OpenCV
* Feature based detection which is faster and more applicable to our project
* Similar principles and steps as SIFT but different details, uses square shaped filters as an approximation of gaussian smoothing
  + Sift uses cascaded filters to detect scale-invariant characteristic points

1. **(Dennis) SIFT -- Dive in**

* **Scale Invariant Feature Transform**
* **Published by David Lowe in 1999**
* **Originally patented in US by University of British Columbia**
* More robust than SURF
* Compatible with OpenCV but possibly not as fast as original implementation
* 1 slide ppt presentation on algorithm side
* Gradient
* DoG

1. **(Dennis) FAST (**[**http://docs.opencv.org/3.0-beta/doc/py\_tutorials/py\_feature2d/py\_fast/py\_fast.html**](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html)**)**

* **Features from Accerlerated Segment Test**
* **Proposed by Edward Rosten and Tom Drummond in 2006 paper and revised by them in 2010 paper**
* OpenCV compatible
* Can be used in conjunction with SIFT
* Extremely fast, but at times less stable than other feature detecting algorithms
  + E.g. trade off speed for reliability?
* Good for real time usage as its extremely fast compared to some others like SURF.

**6. (Chris) ORB (Oriented FAST and rotated BRIEF)**

[**http://www.willowgarage.com/sites/default/files/orb\_final.pdf**](http://www.willowgarage.com/sites/default/files/orb_final.pdf)

* **The addition of a fast and accurate orientation component to FAST.**
* **The efficient computation of oriented BRIEF features.**
* **Analysis of variance and correlation of oriented BRIEF features.å**
* **A learning method for de-correlating BRIEF features under rotational invariance, leading to better performance in nearest-neighbor applications.**

**7. (Chris) Harris Corner**

**Corner detection is an approach used within** [**computer vision**](https://en.wikipedia.org/wiki/Computer_vision) **systems to extract certain kinds of** [**features**](https://en.wikipedia.org/wiki/Feature_detection_(computer_vision)) **and infer the contents of an image. Corner detection is frequently used in** [**motion detection**](https://en.wikipedia.org/wiki/Motion_detection)**,** [**image registration**](https://en.wikipedia.org/wiki/Image_registration)**,** [**video tracking**](https://en.wikipedia.org/wiki/Video_tracking)**,** [**image mosaicing**](https://en.wikipedia.org/wiki/Photographic_mosaic)**,** [**panorama stitching**](https://en.wikipedia.org/wiki/Panorama_stitching)**,** [**3D modelling**](https://en.wikipedia.org/wiki/3D_modelling) **and** [**object recognition**](https://en.wikipedia.org/wiki/Object_recognition)**. Corner detection overlaps with the topic of** [**interest point detection**](https://en.wikipedia.org/wiki/Interest_point_detection)**.**

**8. (Chris) Shi-Tomasi - most popular one**

[**http://docs.opencv.org/3.0-beta/doc/py\_tutorials/py\_feature2d/py\_shi\_tomasi/py\_shi\_tomasi.html**](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html)

**Later in 1994, J. Shi and C. Tomasi made a small modification to it in their paper Good Features to Track which shows better results compared to Harris Corner Detector.**

* **Start construct one slide**

**9: (Chris) FLANN**

[**http://www.cs.ubc.ca/research/flann/**](http://www.cs.ubc.ca/research/flann/)

**FLANN is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. It contains a collection of algorithms we found to work best for nearest neighbor search and a system for automatically choosing the best algorithm and optimum parameters depending on the dataset.**

**OpenCV has some exists algorithm to help us to do image registration. I want to mention the most common one is Harris Corner.**

**Dennis (Image Stitching Open Source Algorithms)**

Statistics (e.g. color image, video, comp time, resolution) need to be found still, issues finding through online resources thus far.

1. Algorithm 1: OpenCV SURF
   1. As mentioned earlier SURF is used to find descriptors and extract them which can then be used to stitch images together
   2. 3x faster than SIFT
2. Algorithm 2: Account.NET library
   1. https://www.codeproject.com/Articles/95453/Automatic-Image-Stitching-with-Accord-NET
   2. .net framework that manipulates images, stitches two images together but can be modified more multiple more.
   3. Steps:
      1. Interest point detection
      2. Correlation matching
      3. Robust homography estimation
      4. Gradient blending
   4. Simple but outdated
3. Algorithm 3: OpenCV panorama stitching using Python and OpenCV guide
   1. (<http://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/>)
   2. Uses keypoints and extracts local invariant descriptors from two input images
      1. Uses SIFT(scale-invariant feature transform) or SURF(speeded-up robust features) which is a sped up version of SIFT, depending on what version OpenCV used
   3. Steps:
      1. **Step #1:** Detect keypoints (DoG, Harris, etc.) and extract local invariant descriptors (SIFT, SURF, etc.) from the two input images.
      2. **Step #2:** Match the descriptors between the two images.
      3. **Step #3:** Use the [RANSAC algorithm](https://en.wikipedia.org/wiki/RANSAC) to estimate a [homography matrix](https://en.wikipedia.org/wiki/Homography_(computer_vision)) using our matched feature vectors.
      4. **Step #4:** Apply a warping transformation using the homography matrix obtained from **Step #3**
   4. Benefits: Simple, easy to use with OpenCV
   5. Cons: Originally intended for 2 images but modifiable for multiple, can it be used in real time?? Computation time? Not many details available so need to test ourselves
   6. s
4. Panotools/pano13 <https://software.intel.com/en-us/articles/fast-panorama-stitching>
   1. Supposedly faster than OpenCV according to Intel article (from 2014) so more usage for real time purpose like ours
   2. Created by Professor Helmut Dersch from University of Applied Sciences Furtwangen
   3. Outdated (last sourceforge update 2013)
   4. s
5. Hugin (<http://hugin.sourceforge.net/>) (?)
   1. Uses pano13 library
   2. Need more research.

Chris (Image Registration)

OpenCV has exist Image Registration Class

The Registration module implements parametric image registration. The implemented method is direct alignment, that is, it uses directly the pixel values for calculating the registration between a pair of images, as opposed to feature-based registration. The implementation follows essentially the corresponding part of

<http://docs.opencv.org/3.1.0/db/d61/group__reg.html>

Image Registration methods <https://docs.google.com/presentation/d/14TUbIQcfDAO9yc8Zog0Xbd_JoBIrDDnoc6CLVVAWHgE/edit#slide=id.p14>

-Doesn’t contain pre-existing algorithms but explains how image registration methods work

**Dennis(Image Recognition)**

1. Algorithm 1: OpenCV Cascade Classifier
2. Algorithm 2: VLFeat Library
3. Algorithm 3: Kaze
   1. Grayscale image
   2. Detects keypoints (corners)
4. Algorithm 4: Histogram of Oriented Gradients (HoG) OpenCV implementation
   1. Created by Robert K. McConnell is 1986 patent but first used in 1994 by Mitsubishi Electric Research Labs and became widespread in 2005.
   2. Creates histograms of edge orientations from patches in images
   3. Then used with machine learning algorithm (such as classical support vector machines?) to train a classifier to distinguish objects
   4. Con: Generally used in pedestrian tracking, can be used to track other objects but results vary.
   5. There is an HOG class present in OpenCV but unsure of usage/if it’s open source.