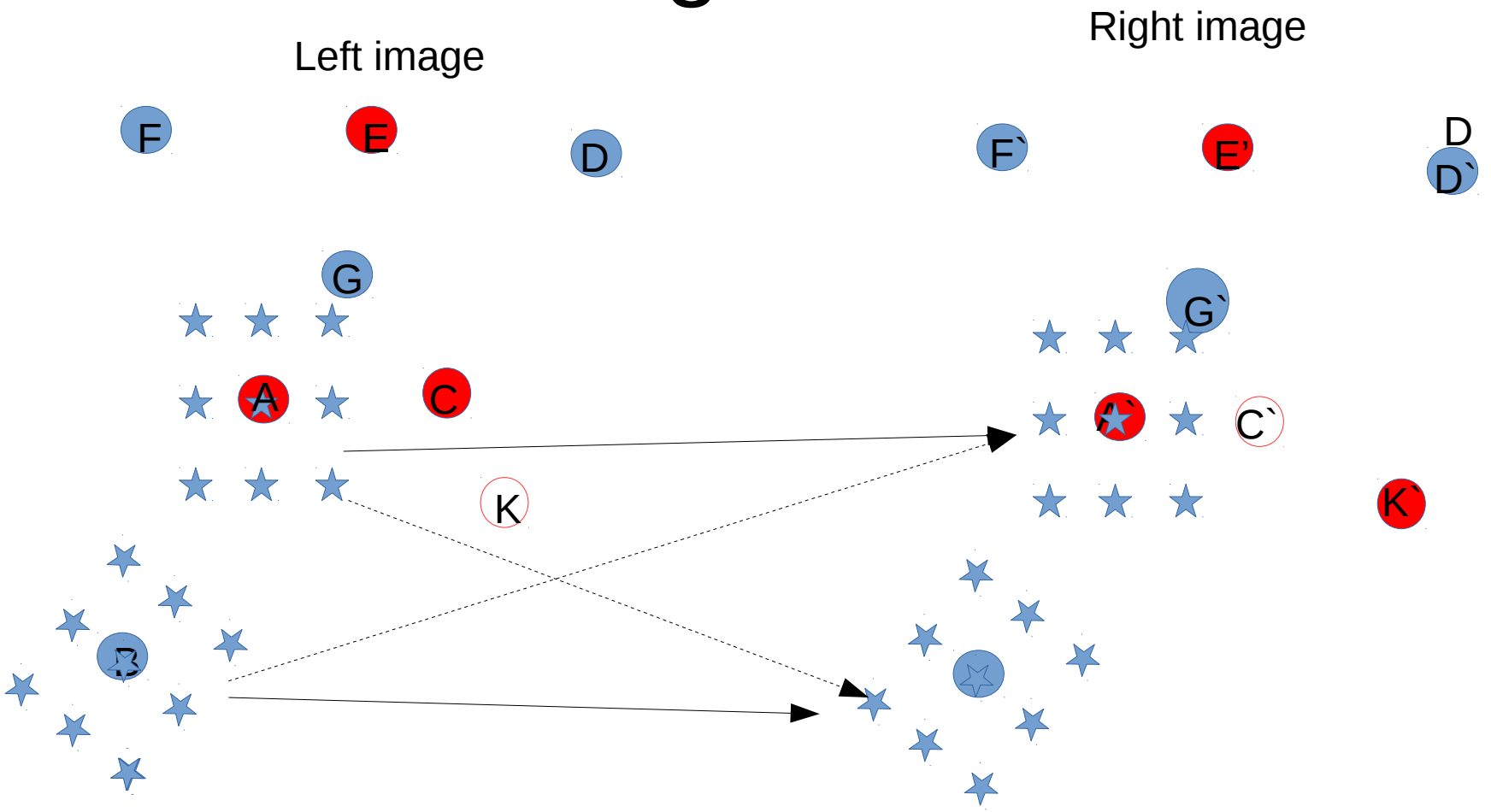
Samsonov Dima Haifa Israel 14 August 2017

Fast and accurate point registration by BOEV -Bigraph Oriented Edges Voting

The problem:

- Current image registration schemes have several problems:
- a) slow O(N^2) feature matching
- b) don’t use native local connectivity for small local affine deformation
- c) low precision

Regular feature extraction and matching scheme

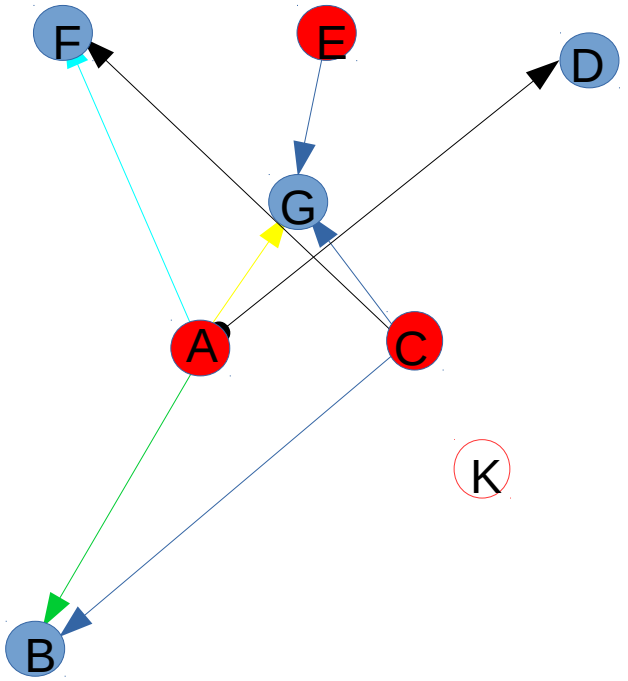


Find keypoints (vertices) on each image(or image pyramid),by local max DoG,LoG,DoH...etc
Sample ” ★“ featured on the some grid around point rotated by local gradient (and scaled)
Find closest (L1 or L2) featured vector on right side.

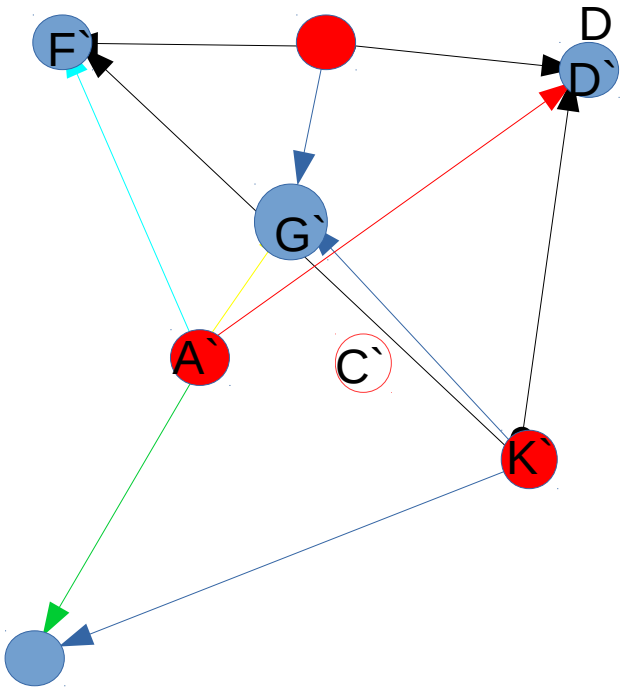
Proposed feature extraction and matching algorithm:

Per each image: Find keypoints(vertices) and make oriented edges

Left image

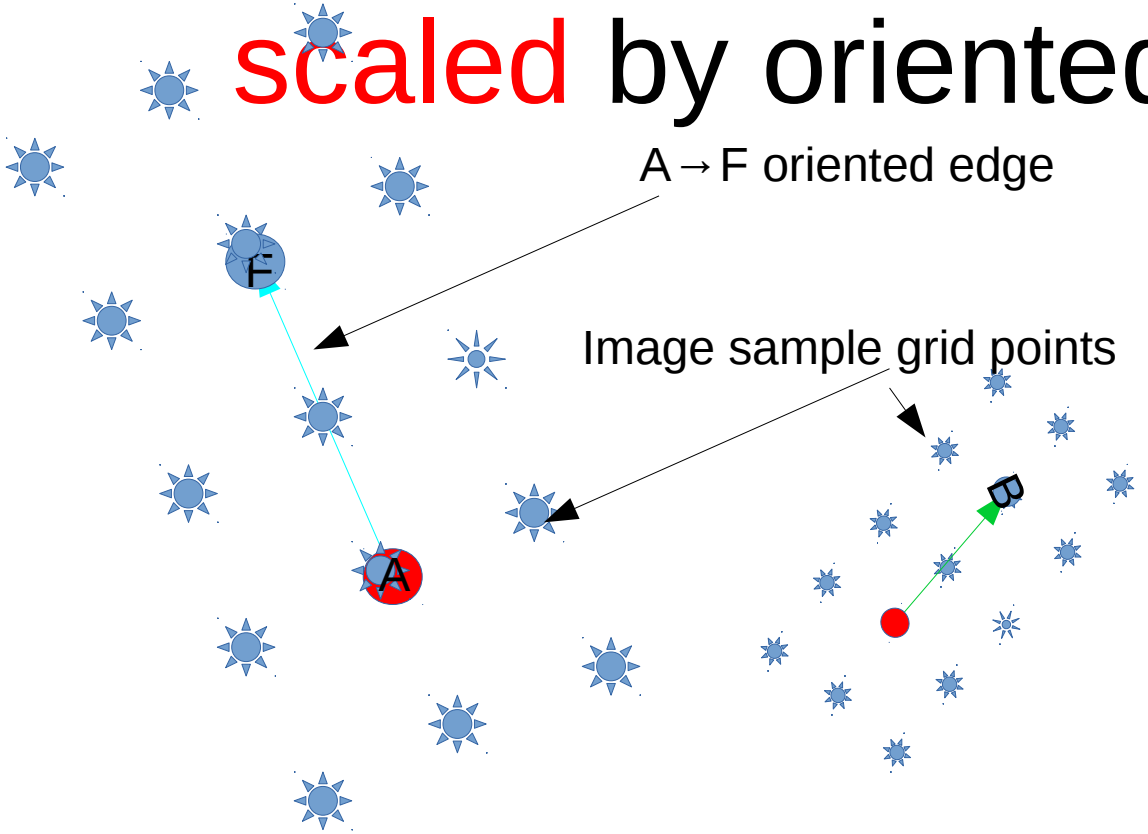


Right image



1 Find keypoints (vertices) on each image by local max DoG,LoG,DoH...etc
2 Set vertex color as “red” if DiffOfGaussians is positive or otherwise “blue”
3 Connect each red vertex with ~6-16 (L) closest blue vertices.
So we have array of ~N/2(red) + ~N/2(blue) verices and array ~(N/2*L) of oriented edges

Per each image:calculate features and hash on some grid aligned and scaled by oriented edge

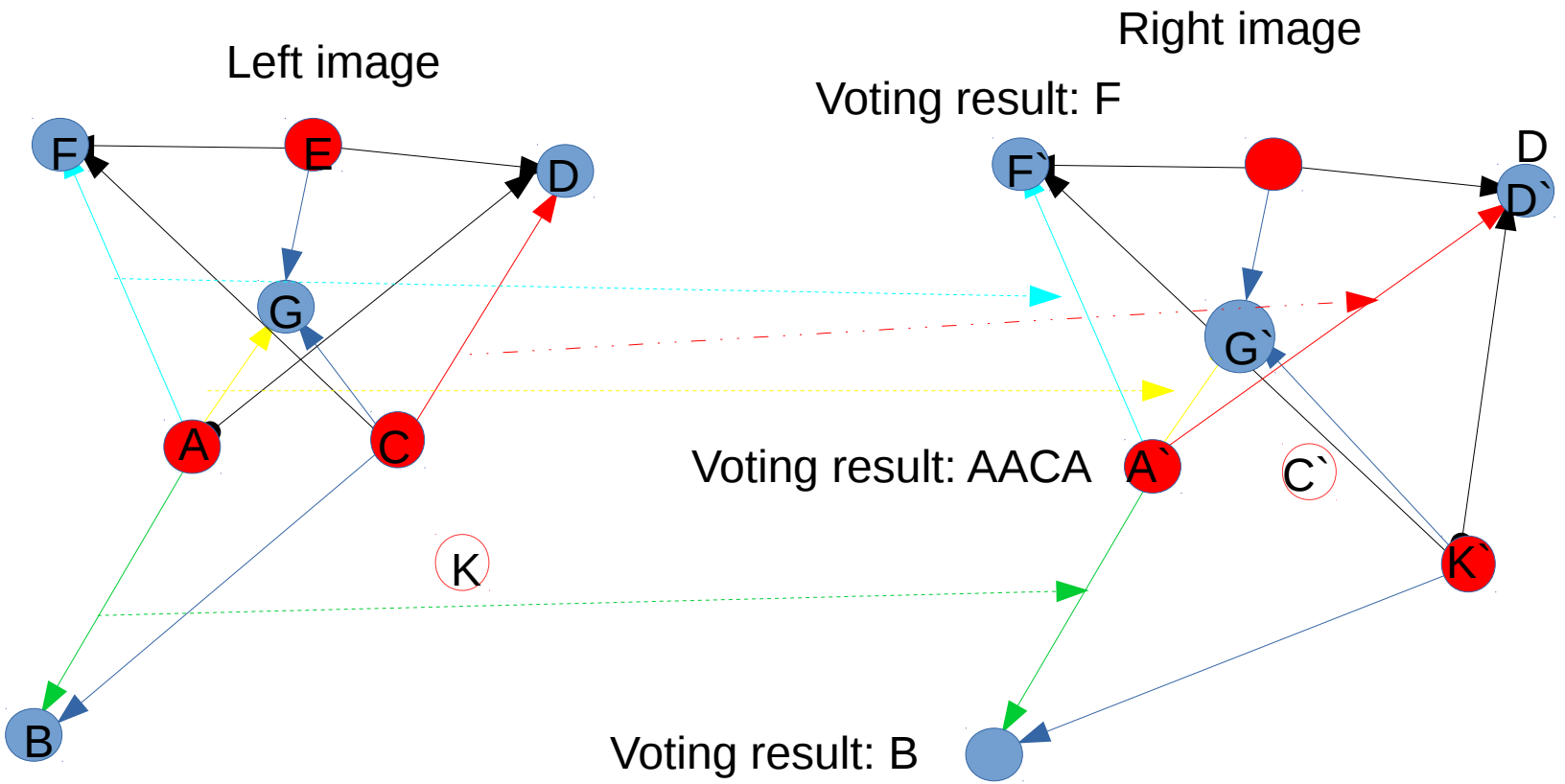


Hash value 0b11100...: vertices A F

Hash value 0b01001...: vertices G B

4 Bilinear sample on 9-16 scaled and rotated grid points from the smoothed image around each edge and calculate ~24bit hash value as sign of samples difference.
5 Reduce ~24 bit hash value to appropriate bit length through some hash16-hash12 crc function for future fast O(1) LUT search

Edges voting vertices matching



Hash value 0b11100: vertices A F

Hash value 0b01100: vertices A G

Hash value 0b01100: vertices C D

Hash value 0b01110: vertices A B

Hash value 0b11100: vertices A` F`

Hash value 0b01100: vertices A` G`

Hash value 0b01100: vertices A` D`

Hash value 0b01110: vertices A` B`

6 For each edge from left image put to vertices of edge(s) with same hash on right image indices of edge vertices
So we have point A` from the Right image paired to Point A from the Left image by voting 3 times for A and one time for C

Example of image registration L=12 without outliers filtering

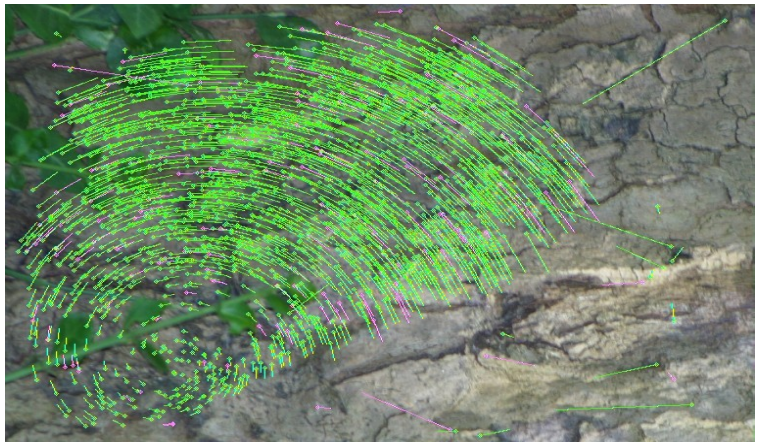
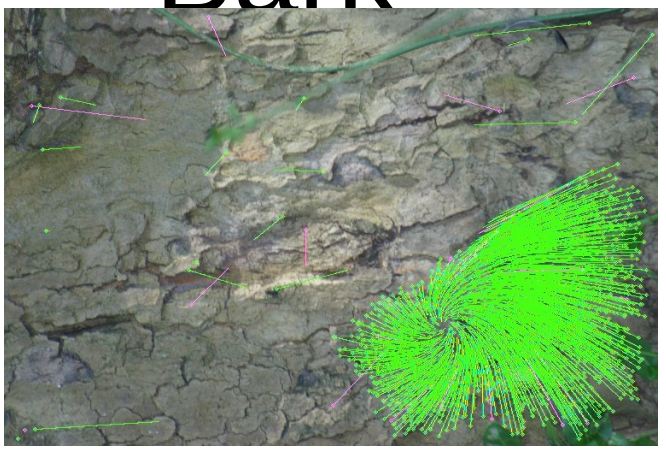
Example of image registration without outliers filtering L=12 fullHD 40 fps

Benchmark images

- Each group of 6 images processed (use 2 cores intel 3.8GHz)
- 6 detections and 5 matchings within 0.9 seconds (not final numbers,need more tuning)

- Images set from: K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky,
- T. Kadir and L. Van Gool, “A comparison of affine region detectors,” Int. Journal of
- Computer Vision, vol. 65, no. 1/2, pp. 43-72, 2005.

Example of image registration,no outliers filtering,motion scaled, L=48 Bark

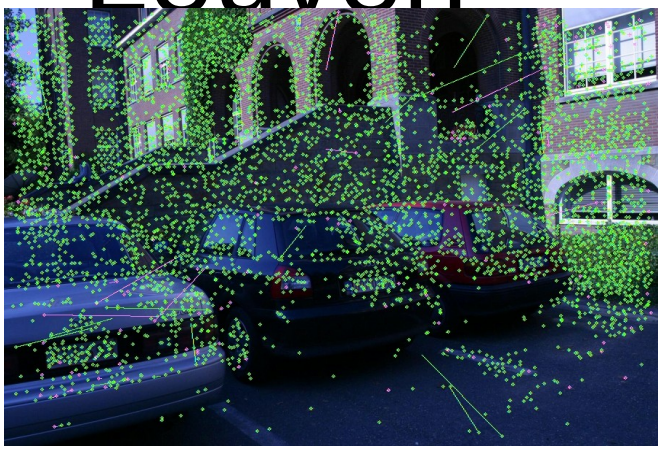



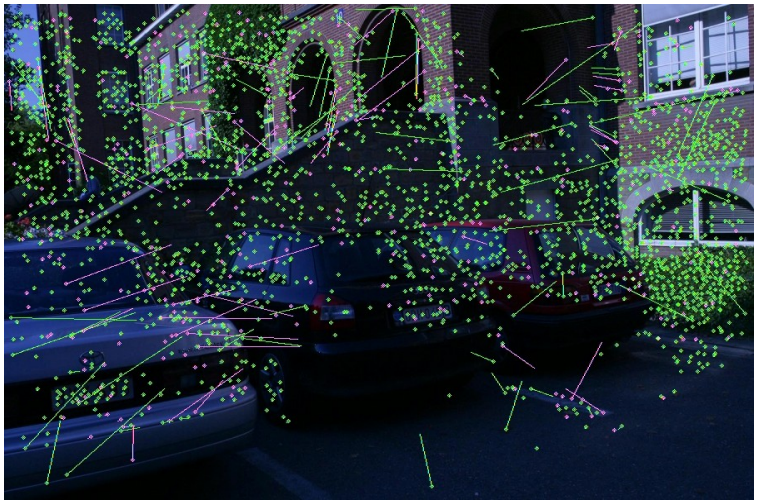

Example of image registration,no outliers filtering,motion scaled, L=48 Bikes





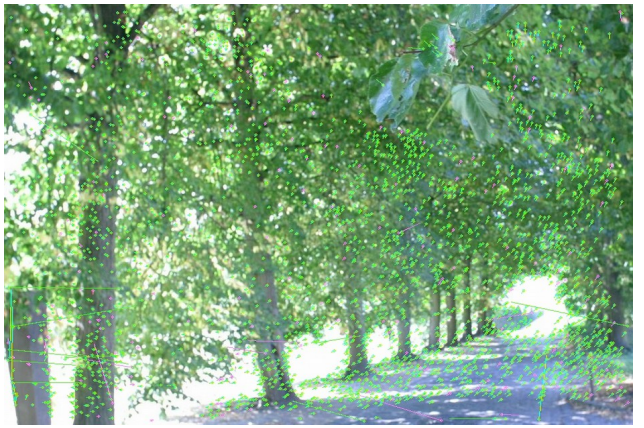
Example of image registration,no outliers filtering,motion scaled,L=48 Leuven

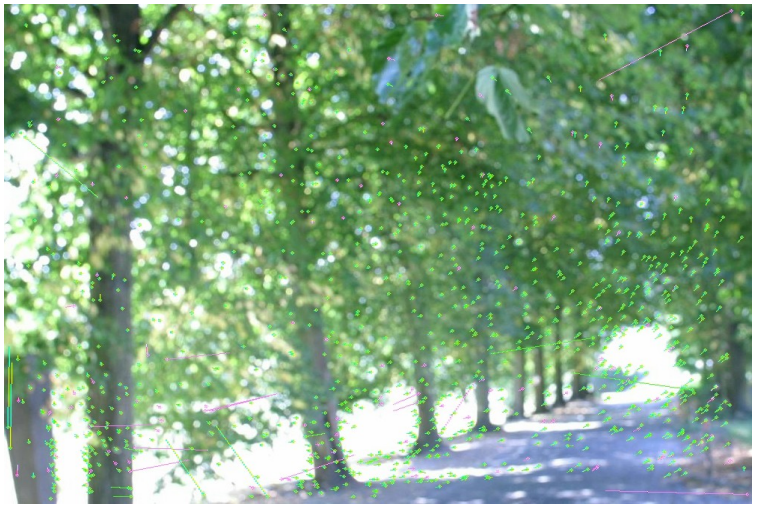





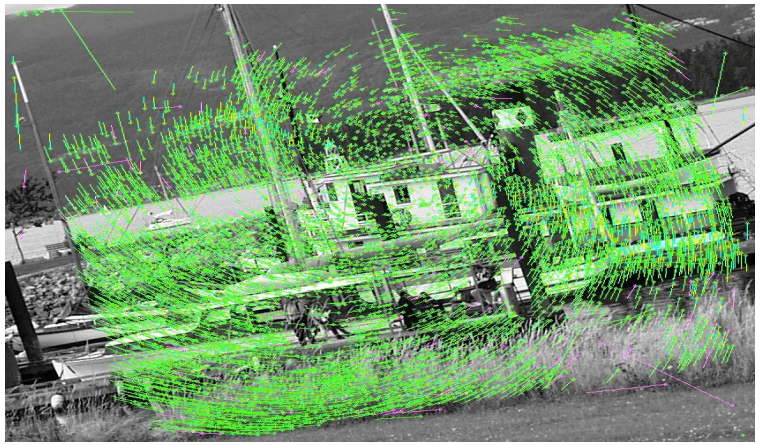
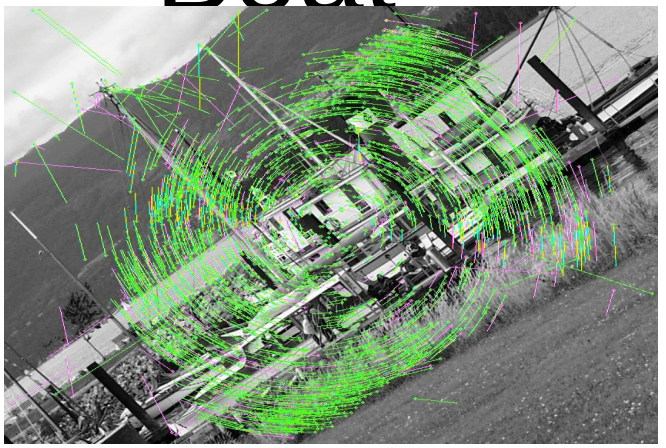
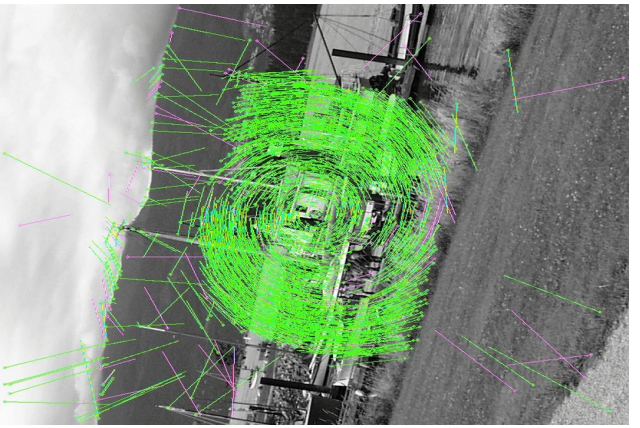

Example of image registration,no outliers filtering,motion scaled,L=48 Trees

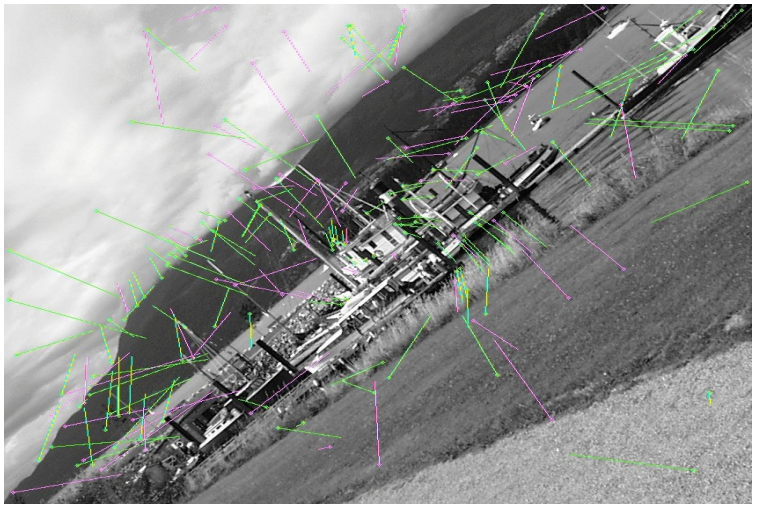




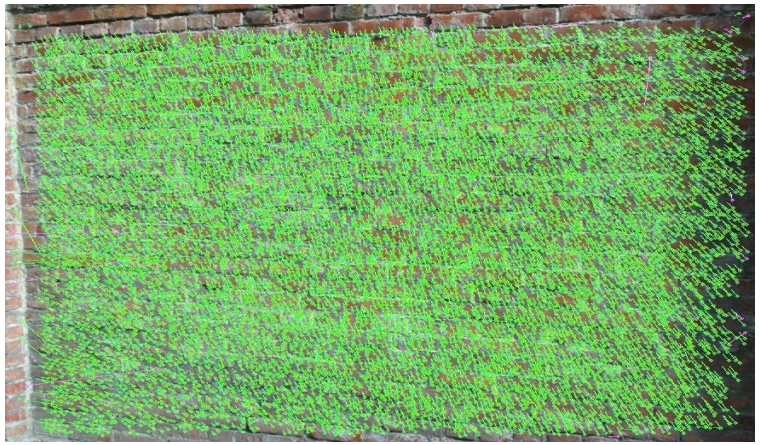


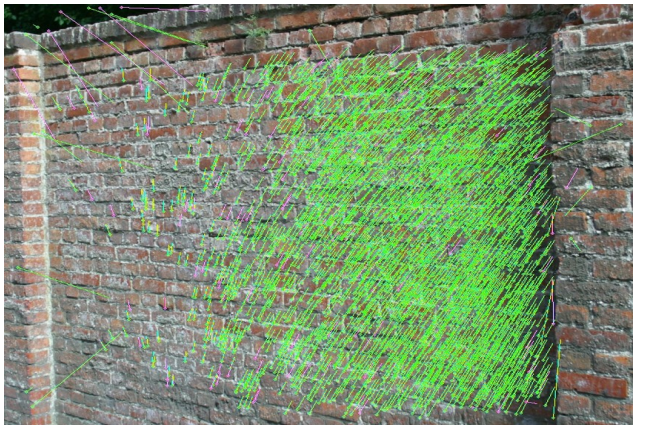
Example of image registration,no outliers filtering,motion scaled,L=48 Boat



Example of image registration,no outliers filtering,motion scaled,L=48 Wall



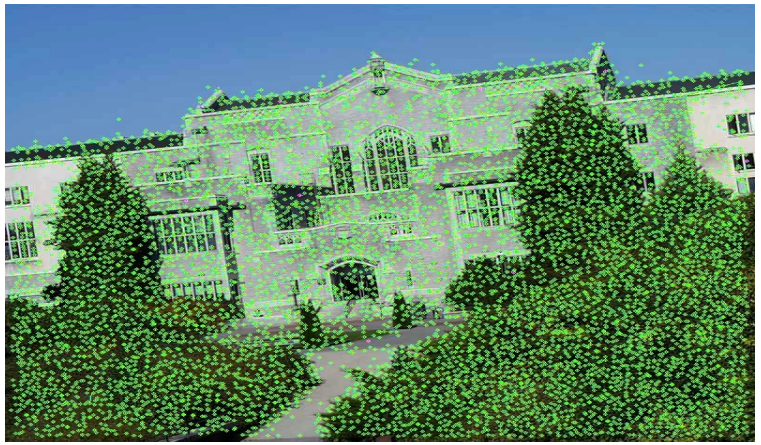
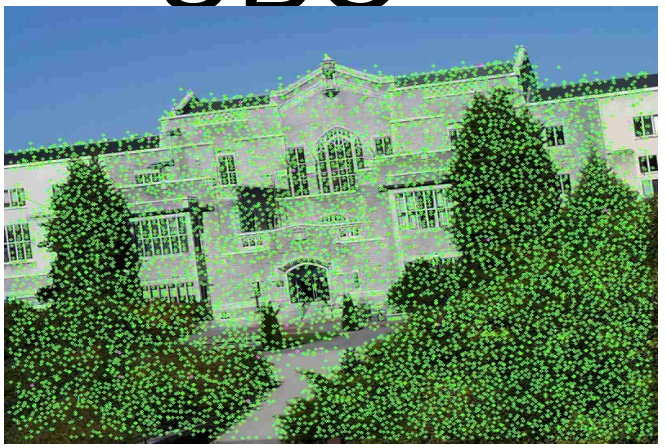


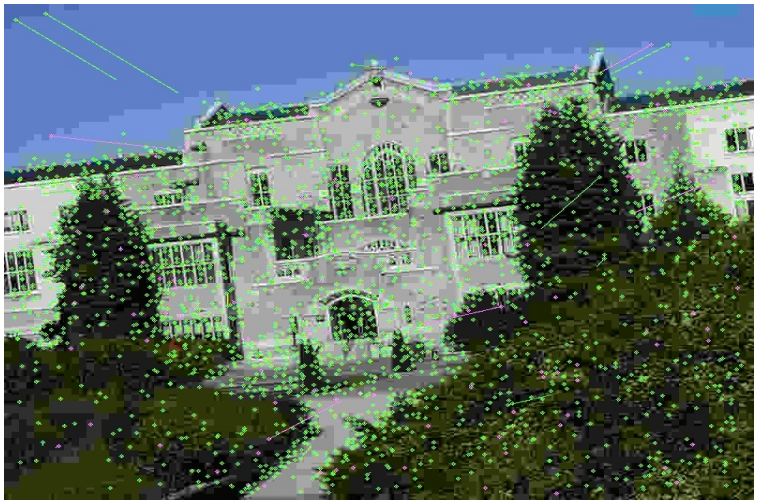
Example of image registration,no outliers filtering,motion scaled,L=48 Graff





Example of image registration,no outliers filtering,motion scaled,L=48 UBC




Conclusion:

Pros:

- Full Rotation and Shift invariant
- Simple code
- Mostly no outliers (due voting scheme)
- Tunable: processing time vs accuracy
- Fast matching (registration) suitable for real time FullHD 30FPS video ,with about 2000-10000 (6000 paired in 6 ms) for vertices per frame on modern dual core cpu or 640x480 30 fps video on rasberrypi3.

Cons:

- Relative sense to scale (but can be modified to work with pyramid)
- Sense to significant projective distortion

Applications

- SLAM and 3d video reconstruction
- Super resolution(due sub-pixel precision)
- Video De-noising
- Video compression
- Image recognition
- Face recognition
- 2d and 3d x-ray registration

Link to open source C++ implementation

- Complete source code in https://github.com/sdima1357/bigraph_image_registration_demo

Links

- SIFT https://en.wikipedia.org/wiki/Scale-invariant_feature_transform
- SURF https://en.wikipedia.org/wiki/Speeded_up_robust_features
- SLAM https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping
-
-