Pattern matching

# Node points generation

Simply corner points of an object are treated as node points. All corner points are computed using Harris Corner Detector.

# Matching #1 (Template Matching)

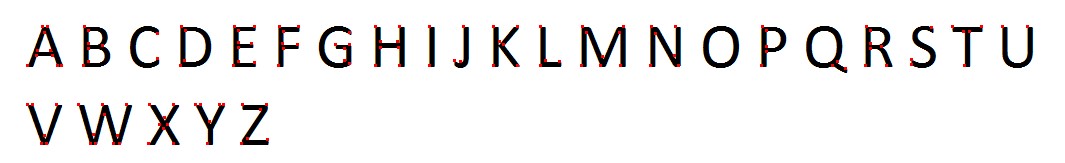
* Take query pattern points and model pattern points
* Matching is done using template matching the query pattern image template on model pattern template image

Drawbacks

Not scale invariant and not rotational invariant

## Results

Model Pattern Image



Query Pattern Image



Result image



# Matching #2 (Naïve 1)

* Make list of points for each query pattern points and model pattern points
* Choose two points each from query pattern list and model pattern list
* Search for the match of query points in the model pattern list basing on the Euclidean distance metric
* Once after the correct match found mark the matched points from the list of model pattern list and remove those matched from the model pattern list for next match

Drawbacks

False positives are coming and high time complex

## Result

Model Pattern Image



Query Pattern Image



Result Image



# Matching #3 (Naïve 2)

* Make list of points for each query pattern points and model pattern points
* Now calculate distance matrix each for query pattern points and model pattern points i.e. distance of a point to all other points in pattern
* Now match the distances from query pattern’s distance matrix with distances in model pattern’s distance matrix
* Once after finding matched point in model pattern point mark the point and remove from list for next finding next match

Drawbacks

False positives are coming and high time complex

## Result

Model Pattern Image Query Pattern Image

Result image



# Matching #4(Naïve 3)

* Choose three points from query pattern
* Loop through all possibilities to choose three points from model pattern
* Now find the affine transform matrix between chosen query reference points and model target points
* Now transform query pattern using obtained affine matrix
* Search for transformed pattern in model pattern using any of above defined matching methods or nearest neighbor search algorithm

# Drawbacks

More time complex and have choose correct thresholds for matching

# Result

Still under development

# Future plans

* The efficient matching of pattern points can be done using k-nearest neighbor search and threshold the distance efficiently evaluated by a kd-tree
* Compute transformation mappings of pattern points so that we can make it scale and rotation invariant
* Relative distance and angle information can be used to limit false positives