

## Assignment 2: Feature Matching for Location/Object Recognition

Gth749a : Kihwan Kim / Gth850c: Sangmin Lee

### 1. Objective

: With 30 different reference images which contain different object/location, we can find out the location of query image by feature matching algorithm. Features from each image are extract directly from SIFT detector. With these extracted features, we will make algorithms for similarity computation, voting and image matching.

### 2. How can we select the best matching result ?

: For the threshold, instead of using certain value to pick the inliers, on each feature of query image, we have picked three DB features which have closest distance to the query feature. On voting scheme, we weighted each vote depending on how close the distance is. For example, suppose we have picked three DB features, we will vote 3 on the image with feature of closest distance, 2 on image with feature of second closest distance, and 1 on image with feature third closest distance.

The reason why we didn't directly apply the threshold to select closest features is that applying certain threshold on each feature-set could be very heuristic procedure.

### 3. Reference Images

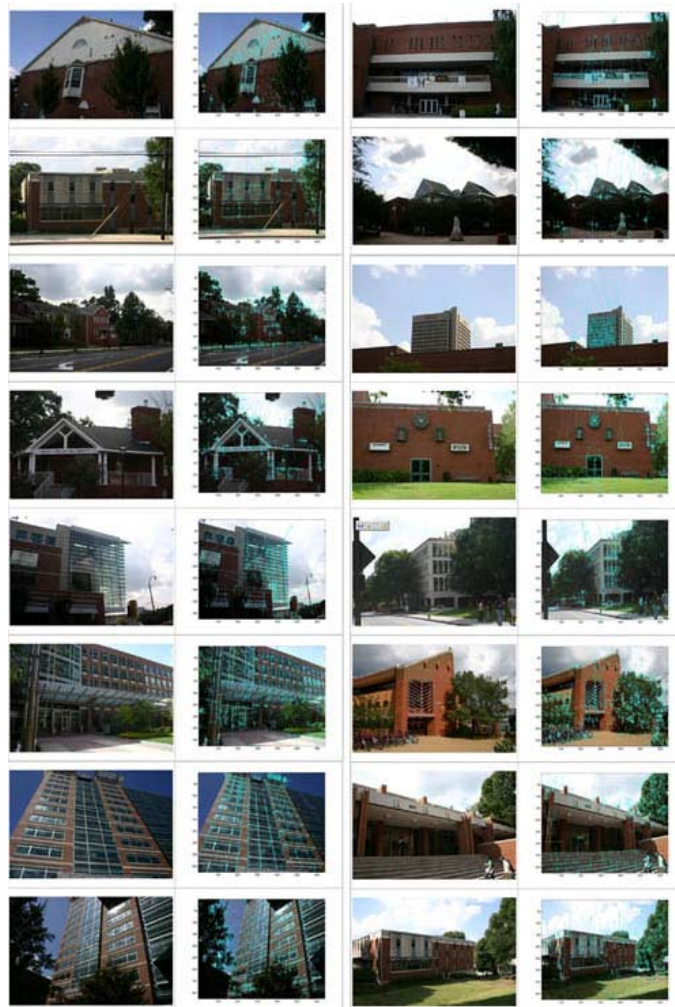
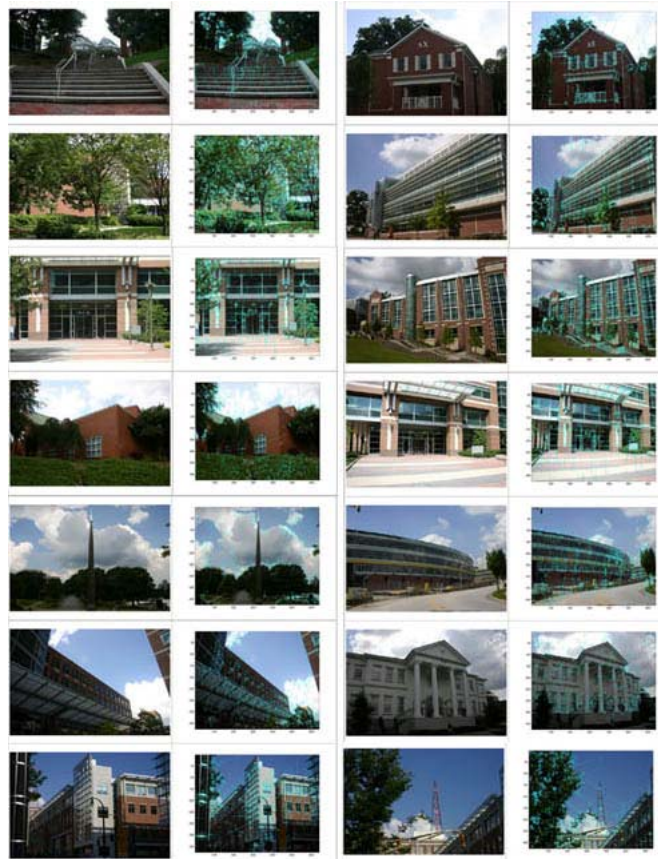
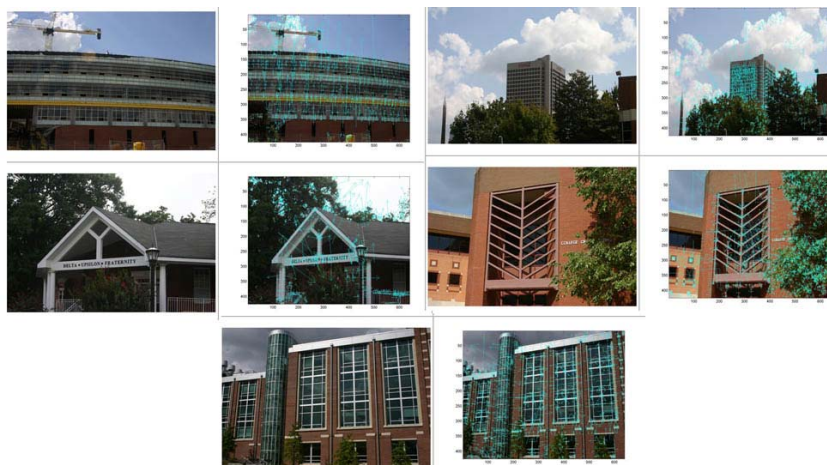


Fig1. Ref' Images -1



**Fig2. Ref' Images -2**

**4. Query Images**



**Fig3. Query Images**

## 5. Similarity Matrix

: Distance between same image but different resolution.  
We did it in two different ways

(1) We used two images (13.jpg and 1mod.jpg) where 1mod.jpg is same image with different resolution.

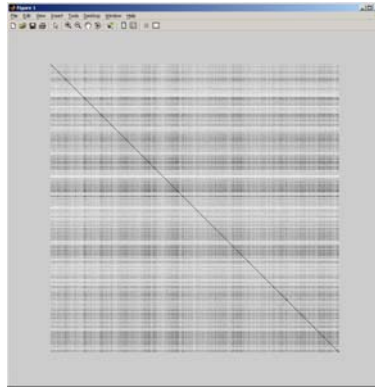


Fig4. Similarity Matrix Image #1

(2) We had sift function to change the ratio of the image. i.e we only used one image in this case

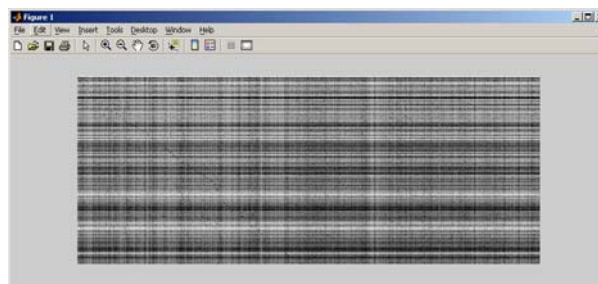
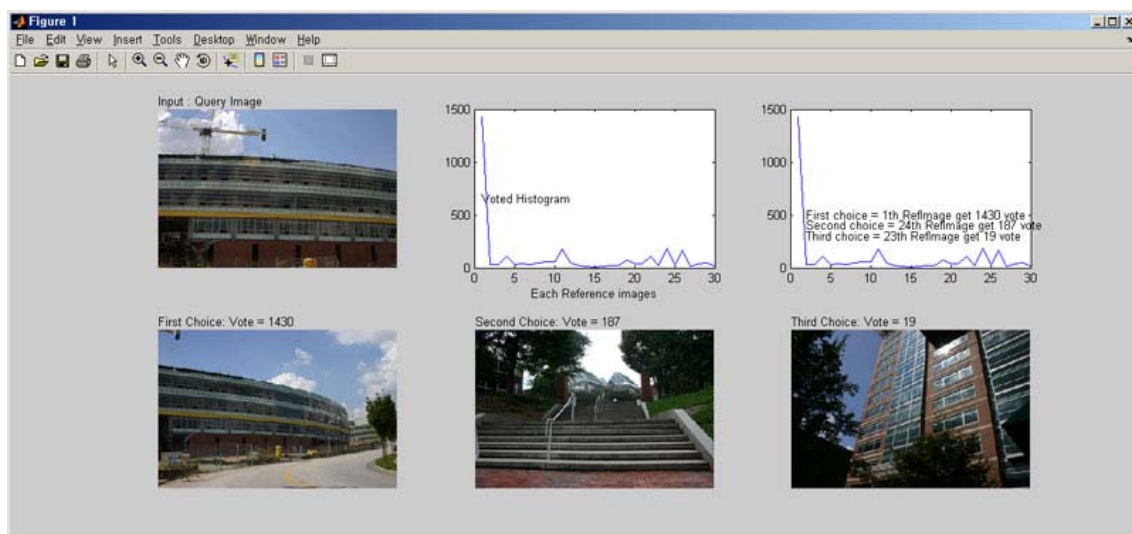
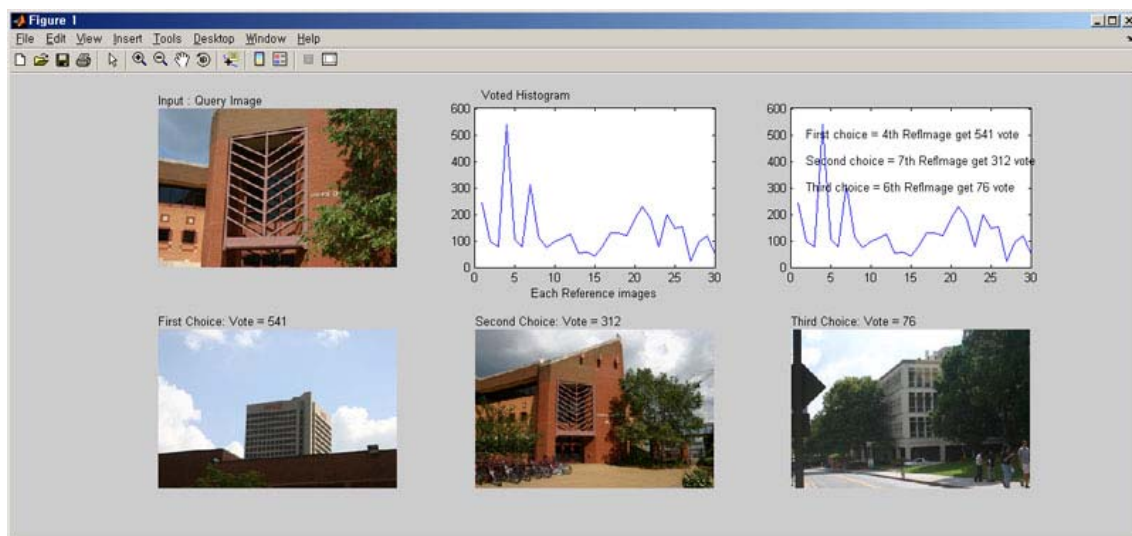
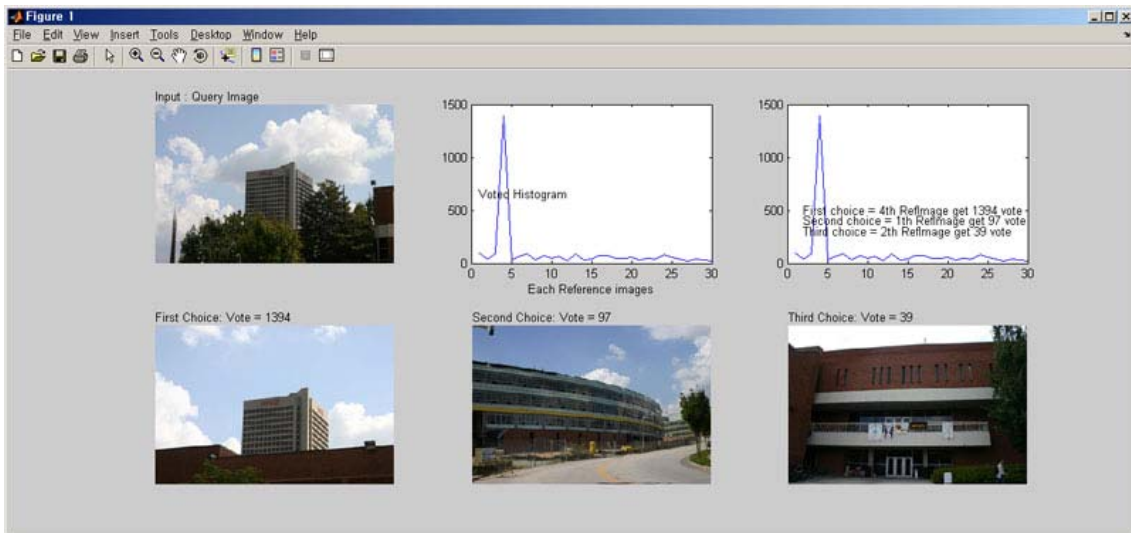
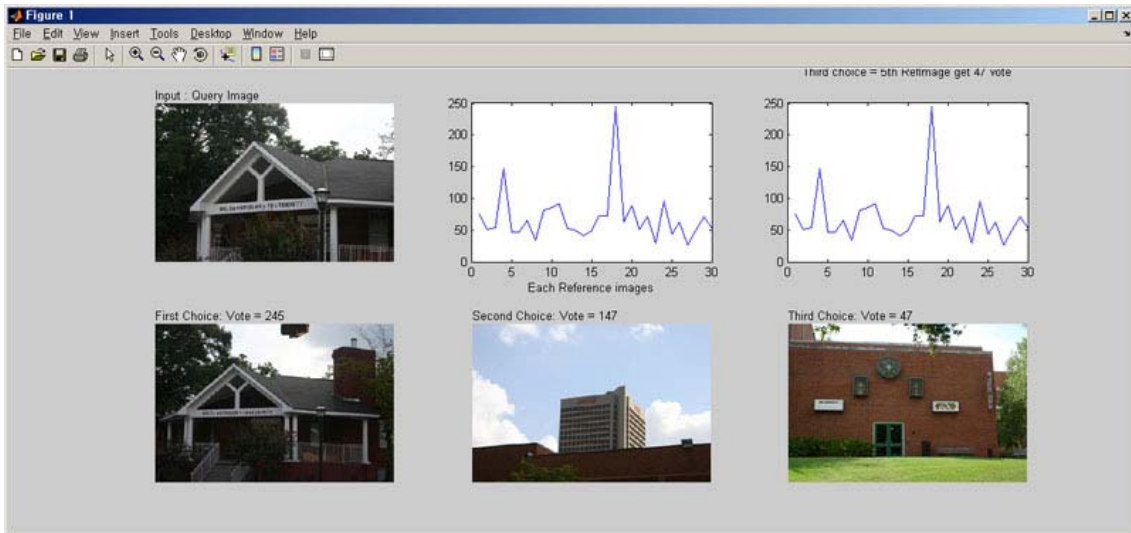


Fig5. Similarity Matrix Image #2

## 6. Result Images & Voting Histograms & Selected Images





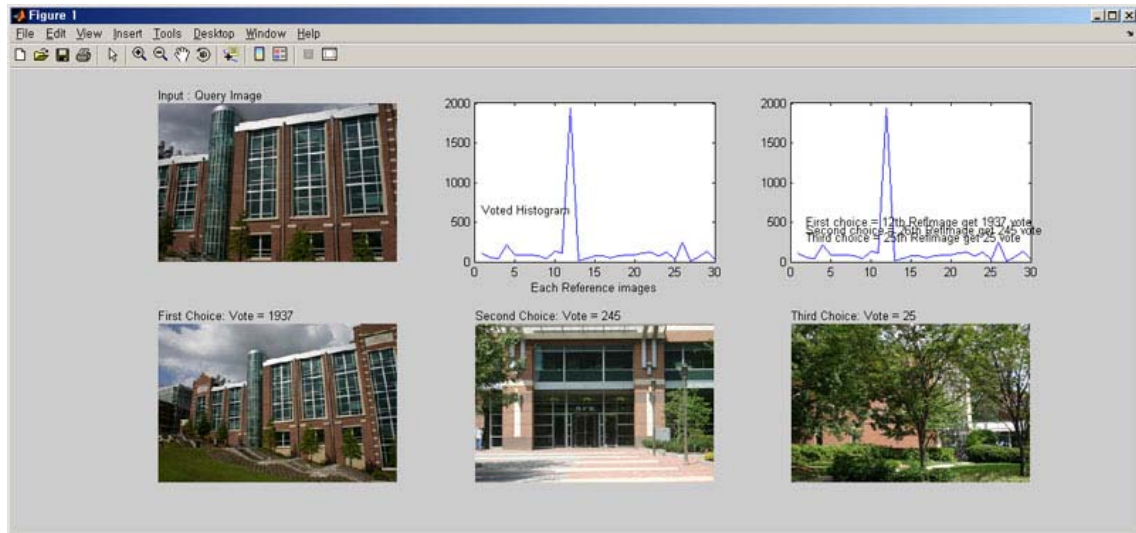


Fig6. Result Images

## 6. Result Analysis

: As shown above, except query4 image, most of the result is good. They found exact locations in the DB(reference images). Most of them got the relatively high voting Numbers from voting process. However, when we look at the 4<sup>th</sup> query image( CoC building ) below, we can find out that they picked the CoC building as a second selection.

The reason for this query image giving us the wrong match would be as follows (we presume)  
 First of all the database image has too much zoomed out view that most significant feature within query image (we think it as V-shaped structure), was not so well detected in the database image. So, it could reduce the possibility to gain the less voting number compare with other 1<sup>st</sup> choice-image.  
 Second the V-shaped structure within query image has similar orientation with edge of building in most voted image or third most voted image.

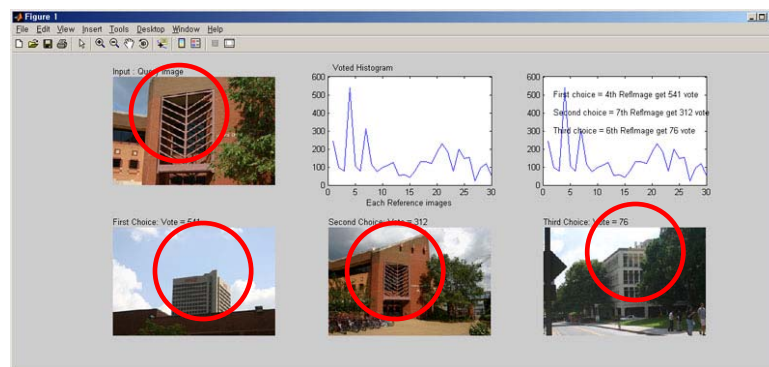


Fig7. Mismatched Image

## 7. Execution Procedure ( from Readme.txt )

### \* Feature Matching algorithm

- (1) Place the Ref Images(1.jpg~30.jpg) on the same directory
- (2) Place the Query Images(que1.jpg~que5.jpg) on the same directory
- (3) Execute "**Generate\_DataOfRefImages.m**" to generate feature data of Ref Images ( feat1.mat ~ feat30.mat )
- (4) Execute "**Generate\_DataOfQuerImages.m**" to generate feature data of Query Images ( feat\_ref1.mat ~ feat\_ref5.mat )
- (5) Execute "**first\_step.m**"

- Load features from 30 ref-images and make feature DB(library)
- Save the result of DB in RefImgDB.mat
- making FeatureIdx matrix(look up table) : same N of row with DB.
- (6) Execute "**second\_step.m**" - Calculate Distance matrix(Similarity Matrix)
  - Take 3 candidate of each Quer-feature to Ref-feature matching
  - Weighting 3 candidate each
  - Summing up for voting and accumulate in Voting Histogram.
  - Select 3 ref images ( different from weighted 3 candidate above. )
- (7) Execute "**third\_step.m**" to display the every result.
  - Display 'query image','histogram', and selected 3 reference images.

**\* Similarity Matrix**

: Distance between same image but different resolution. We did it in two different ways

- (1) We used two images (13.jpg and 13mod.jpg) where 1mod.jpg is same image with different resolution.
  - Place the images 1.jpg and 1mod.jpg on same directory
  - Execute "**Gen\_mod.m**" to generate feature data of 1.jpg and 1mod.jpg.
  - Execute "**Sim1.m**" to generate and display similarity matrix between 1.jpg and 1mod.jpg.
- (2) We had sift function to change the ratio of the image.
  - i.e we only used one image in this case
  - Place the images 1.jpg on same directory
  - Execute "**Gen\_mod1.m**" to generate feature data of 1.jpg and changed form of 1.jpg.
  - Execute "**Sim1.m**" to generate and display similarity matrix between 1.jpg and changed form of 1.jpg.

**8. Source code for each algorithm**

**\* Feature Matching algorithm**

**-Generate\_DataOfQuerImages.m**

```
clear all;
for j = 1:5
    str = sprintf('que%d.jpg',j);
    str2 = sprintf('featque%d',j);
    [image, descripts, locs] = sift(str,1);
    save(str2);
end
disp('Feature data of 5 query images are generated!');
```

**-Generate\_DataOfRefImages.m**

```
clear all;
for j = 1:30
    str = sprintf('%d.jpg',j);
    str2 = sprintf('feat%d',j);
    [image, descripts, locs] = sift(str,1);
    save(str2);
end
disp('Feature data of 30 refrence images are generated!');
```

**- first\_step.m**

```
%%%%%%%%%%
% First Step
%%%%%%%%%%
clear all;
%%%%%%%%%%
% Ref Images & Data of each Ref Images
% SHOULD be on the same directory of this file !!!!!!!
% ie) feat1.mat ~ feat30.mat
%%%%%%%%%%
nTotalRef = 30;

%%%%%%%%%%
% Load features of every reference images , then put this data into
% DescriptsDB
%%%%%%%%%%
```



```

nUpperCand = 3; % How much we will choose the best candidate in voting histogram
MinDistFeatureCand = [0]; % save the distances of the most 'nUpperCand'st candidates
MinDistFeatNo = [0]; % save the Idx(Ref) number of the most 'nUpperCand'st candidates
SelectedDistanceMatrix = [0]; %nUpperCand(threshold) by nquery(feature)
SelectedFeatureNoMatrix = [0];
RefVoteHist = zeros(nTotalRef,1); %image voting histogram.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculating Distances and get similarity matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ztemp = (sqrt(D))*P*QueryDescripts';
ztemp2 = (sqrt(D))*P*DescriptsDB';
distQtoRef = distance(ztemp, ztemp2) ;
SimilarityMatrix = distQtoRef;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sub-Sorting for getting candidate corresponding nUpperCand
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for nquery = 1:sizeQuery
    distV = SimilarityMatrix(nquery,:);
    for i = 1:nUpperCand % just find upper nUpperCand number of minimums.
        nUpperCand == thresh hold?
            % select upper 3 minimums
            [c, where] = min(distV);
            IndexNo = FeatureIdx(where,1);
            if i==1
                MinDistFeatureCand = c; % Minimum Dist( Q-DB ) Candidate matrix
                MinDistFeatNo = where; % Matched Candidate no in DBz
                RefVoteHist(IndexNo,1) = RefVoteHist(IndexNo,1) + 3; % Weight for vote
            else
                MinDistFeatureCand = [MinDistFeatureCand( 1:size(MinDistFeatureCand) ) ; c];
                MinDistFeatNo = [ MinDistFeatNo(1:size(MinDistFeatNo)) ; where ];
                if i==2
                    RefVoteHist(IndexNo,1) = RefVoteHist(IndexNo,1) + 2 ; % Weight for vote
                else
                    RefVoteHist(IndexNo,1) = RefVoteHist(IndexNo,1) + 1 ; % Weight for vote
                end
            end
        end

        if where<2 % if Minimum exist 1st row of disQtoRef
            distV = [ distV(2:size(distV)) ];
        else
            distV = [ distV(1:where - 1) ; distV( (where + 1):size(distV) )];
        end
    end
end

if nquery == 1
    SelectedDistanceMatrix = MinDistFeatureCand;
    SelectedFeatureNoMatrix = MinDistFeatNo;
else
    SelectedDistanceMatrix = [ SelectedDistanceMatrix MinDistFeatureCand ];
    SelectedFeatureNoMatrix = [ SelectedFeatureNoMatrix MinDistFeatNo ];
end
end
end

```

### - third\_step.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Get 3 most much accumulated Idx
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TempRefVoteHist = RefVoteHist;
for i = 1:3
    if i == 1

```



```

    [maxAccum, FirstCandidate] = max(TempRefVoteHist);
elseif i == 2
    [maxAccum, SecondCandidate] = max(TempRefVoteHist);
    if FirstCandidate <= SecondCandidate
        SecondCandidate = SecondCandidate + 1;
    end
else
    [maxAccum, ThirdCandidate] = max(TempRefVoteHist);
    if (FirstCandidate <= ThirdCandidate)&&(SecondCandidate <= ThirdCandidate)
        ThirdCandidate = ThirdCandidate + 2;
    elseif (ThirdCandidate <= FirstCandidate)||((ThirdCandidate <= SecondCandidate)
        ThirdCandidate = ThirdCandidate + 1;
    else
        aaa == 1;
    end
end

if FirstCandidate == 1
    TempRefVoteHist = [ TempRefVoteHist(2:size(TempRefVoteHist)) ];
else
    TempRefVoteHist = [ TempRefVoteHist(1:FirstCandidate - 1) ;
TempRefVoteHist( (FirstCandidate + 1):size(TempRefVoteHist) )];
end
end

%%%%%%%%%%%%%%
% Result Display
%%%%%%%%%%%%%%
str1 = sprintf('%d.jpg',FirstCandidate);
str2 = sprintf('%d.jpg',SecondCandidate);
str3 = sprintf('%d.jpg',ThirdCandidate);

str11 = sprintf('First Choice: Vote = %d',RefVoteHist(FirstCandidate) );
str22 = sprintf('Second Choice: Vote = %d',RefVoteHist(SecondCandidate) );
str33 = sprintf('Third Choice: Vote = %d',RefVoteHist(ThirdCandidate) );

img1 = imread(str);      % Query Image
img2 = imread(str1);     % First candidate Ref Image
img3 = imread(str2);     % Second candidate Ref Image
img4 = imread(str3);     % Third candidate Ref Image

subplot(2,3,1);
imshow(img1);
text(1,-10,'Input : Query Image');
subplot(2,3,2);
plot(RefVoteHist);      % Display Voting Histogram
text(1,650,'Voted Histogram ');
xlabel('Each Reference images');

subplot(2,3,3);
plot(RefVoteHist);
str333 = sprintf('First choice = %dth RefImage get %d vote
',FirstCandidate,RefVoteHist(FirstCandidate) );
str444 = sprintf('Second choice = %dth RefImage get %d vote
',SecondCandidate,RefVoteHist(SecondCandidate) );
str555 = sprintf('Third choice = %dth RefImage get %d vote
',ThirdCandidate,RefVoteHist(ThirdCandidate) );
text(2,500,str333);
text(2,400,str444);
text(2,300,str555);

subplot(2,3,4);
imshow(img2);

```

```
text(1,-10,str11);  
text(1,-10,str11);
```

```
subplot(2,3,5);  
imshow(img3);  
text(1,-10,str22);  
subplot(2,3,6);  
imshow(img4);  
text(1,-10,str33);
```