CSE3018 : Content Based Image and Video Retrieval

Lab - 13

Naga Harshith Bezawada
19BCE1547

SURF Code

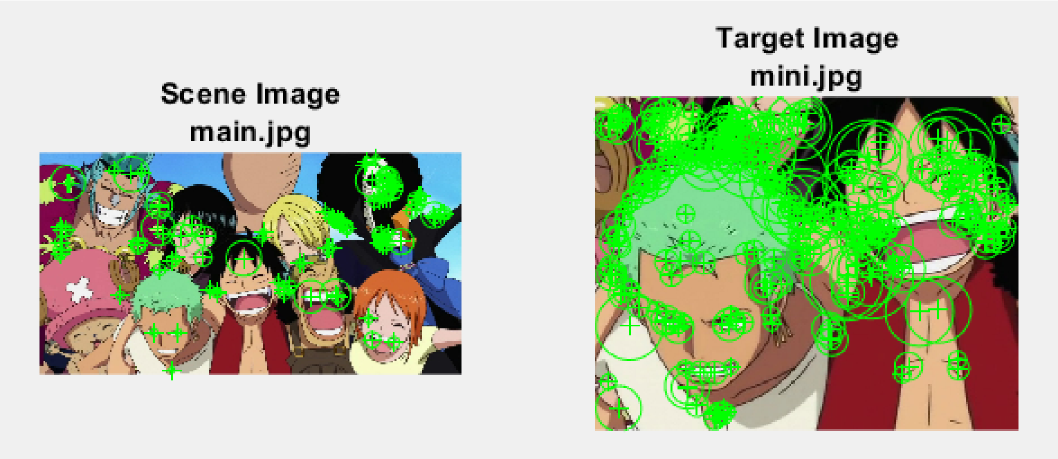
```
clc
close all
clear all
set(gcf, 'Name','SURF keypoints','NumberTitle','off');
%% Select Scene image
[scene_name, file_path] = uigetfile('*.*', 'Select the scene image');
scene_img_path = sprintf("%s%s", file_path, scene_name);
if scene_name
scene_img = imread(scene_img_path);
scene_img_gray = im2gray(scene_img);
else
disp('No scene image selected!');
return
end
% Extract SURF features for scene image
points = detectSURFFeatures(scene_img_gray);
subplot(1, 2, 1);
imshow(scene_img); hold on;
title(sprintf('Scene Image\n%s', scene_name));
```

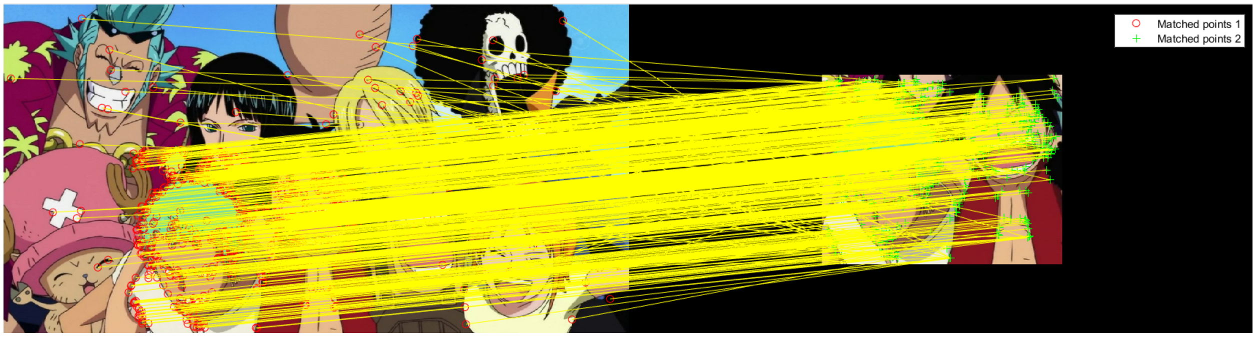
```
% Extracting top 100 points
plot(points.selectStrongest(100));


%% Select Target image
% Either pick a new image or rotate the scene image
%% For rotating the scene image, uncomment the line below
% target_img = imresize(imrotate(scene_img, -20), 1.2);
%% For picking the a new target image
[target_name, file_path] = uigetfile('*.*', 'Select the target image');
target_path = sprintf("%s%s", file_path, target_name);
if target_name
target_img = imread(target_path);
target_img_gray = im2gray(target_img);
else
disp('No target image selected!');
return
end
% Extract SURF features for target image
points2 = detectSURFFeatures(target_img_gray);
subplot(1, 2, 2);
imshow(target_img); hold on;
title(sprintf('Target Image\n%s', target_name));
% Extracting top 300 points
plot(points2.selectStrongest(300));
```

```
%% Matching images
% Extracting the features
[f1,vpts1] = extractFeatures(scene_img_gray, points);
[f2,vpts2] = extractFeatures(target_img_gray, points2);

% Retrieving the locations of matched points
indexPairs = matchFeatures(f1,f2) ;
matchedPoints1 = vpts1(indexPairs(:,1));
matchedPoints2 = vpts2(indexPairs(:,2));

% Displaying the matching points
figure('Name','Matched points','NumberTitle','off'); ax = axes;
showMatchedFeatures(scene_img, target_img, matchedPoints1,
matchedPoints2, 'montage','Parent',ax);
title(ax, 'Candidate point matches');
legend(ax, 'Matched points 1','Matched points 2');
```

OUTPUT

Key Points





SIFT CODE

```
clc;
clear all;
close all;
%% Select Scene image
[scene_name, file_path] = uigetfile('*.*', 'Select the scene image');
scene_img_path = sprintf("%s%s", file_path, scene_name);
if scene_name
```

```
scene_img = imread(scene_img_path);
scene_img_gray = single(im2gray(scene_img));
else
disp('No scene image selected!');
return
end

%% Single image SIFT feature extraction
figure('Name', 'SIFT frames & descriptors', 'NumberTitle',
'off');
imshow(scene_img); % Display image

% Compute the SIFT frames (keypoints) and descriptors
[f,d] = vl_sift(scene_img_gray);
perm = randperm(size(f, 2));
sel = perm(1:50);
h1 = vl_plotframe(f(:, sel));
h2 = vl_plotframe(f(:, sel));
set(h1, 'color', 'k', 'linewidth', 3);
set(h2, 'color', 'y', 'linewidth', 2);
h3 = vl_plotsiftdescriptor(d(:,sel), f(:,sel));
set(h3,'color','g');

%% Select Target image
% Either pick a new image or rotate the scene image
% For rotating the scene image, uncomment the line below
% target_img = imresize(imrotate(scene_img, -20), 1.2);
% For picking the a new target image
[target_name, file_path] = uigetfile('*.*', 'Select the target image');
target_path = sprintf("%s%s", file_path, target_name);
if target_name
target_img = imread(target_path);
else
disp('No target image selected!');
return
end

figure('Name', 'Selected Images', 'NumberTitle', 'off');
subplot(1,2,1); imshow(scene_img);  % Display scene image
title(sprintf('Scene image\n%s', scene_name));



%% Image Matching

% --- Resize target image ---
% scene_img_size = size(scene_img);
% target_img_size = scene_img_size(1:2);% => size(scene_img_gray)
target_img = imresize(target_img, size(scene_img_gray));
subplot(1,2,2); imshow(target_img); % Display target image
title(sprintf('Target image\n%s', target_name));
```

```
target_img_gray = single(im2gray(target_img));

% --- NOTE: Images have to be the same size ---
[fs, ds] = vl_sift(scene_img_gray);
[ft, dt] = vl_sift(target_img_gray);
[matches, scores] = vl_ubcmatch(ds, dt);

%% Visualization
figure('Name', 'Image Matching Visulization', 'NumberTitle',
'off');
ax = axes;
m1 = fs(1:2,matches(1,:));
m2 = ft(1:2,matches(2,:));
m2(1,:) = m2(1,:) + size(scene_img_gray,2) * ones(1,size(m2,2));
X = [m1(1,:);m2(1,:)];
Y = [m1(2,:);m2(2,:)];
c = [scene_img target_img];
imshow(c,[]);
hold on;

% Determine number of lines depending on dimensions of X and Y
line(ax, X, Y);
```

OUTPUT

SIFT Frames and Descriptors

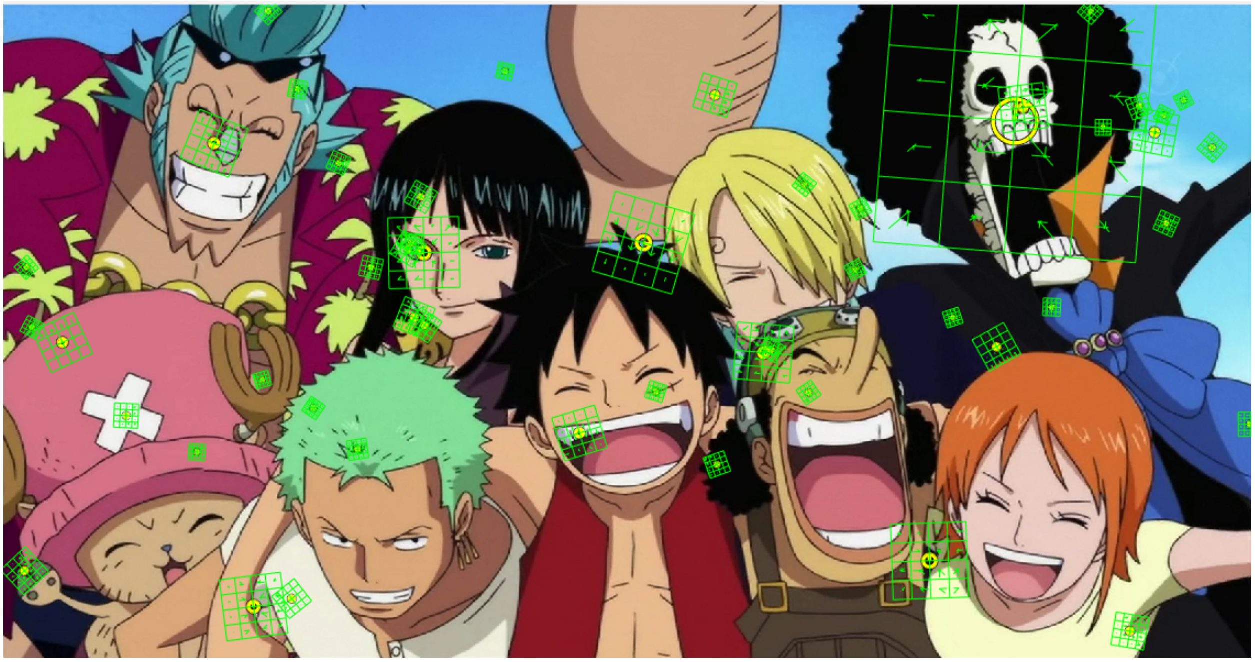


Image Matching Visualization