EECE5639 Computer Vision Term Project Report

Yu Shun Lin,
Electrical and Computer Engineering, lin.yus@husky.neu.edu

I. Abstract:

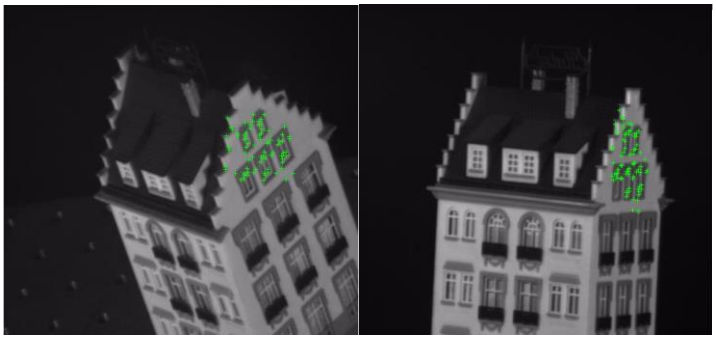
Assuming there is a 3D pattern rotate in the video frame which can be separate to the sequence of the pictures. We could apply Harris corner detector to these pictures to track the specific part when it changes its position. The images sequence is supposed to have two different angles of camera catching. However, the source only provides one of the cameras catching. Thus, we should assume each tracking distance in the original frame (first frame).

II. Description of Algorithms:

1. Read the image sequence from the source folder.
2. Create the video stream from the sequence of images.
3. Select a specific rectangle for tracking.
4. Apply Harris corner detector to the specific part.
5. Apply tracking toolbox to the selected rectangle.
6. Create the video player in MATLAB.
7. Apply the tracker and video player to each frame of video until it ends.
8. Close the video player and reader.
9. Create the table of time and 3D coordinate of tracking points.
10. Transfer the 3D coordinate points to point Cloud object.
11. Create the ASCII PLY File to store the information of coordination.

III. Experiments:

Traking of harris points:



These three figures show the harris point of the specific region, using the KLT tracker for the tracking.

IV. Observations:

The tracker has tracked the points during the frame plays.

EECE5639 Computer Vision Term Project Report

V. Conclusions:

Originally, we should have two different frames of the picture to estimate the depth of the camera. However, KLT tracker could not estimate the depth of the frame if we did not provide the information of the camera and 3D model true dimensions.

VI. References:

R. Szeliski, Springer. (2010) “Computer Vision Algorithms and Applications”
http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf
Matlab Documentation
https://www.mathworks.com/help/

VII. Appendix:

Matlab Codes:

Tracking

```
videoFileReader = vision.VideoFileReader('frame.avi');
videoPlayer = vision.VideoPlayer('Position',[100,100,680,520]);
objectFrame = videoFileReader();
objectRegion = [320,160,100,100];
objectImage =
insertShape(objectFrame,'Rectangle',objectRegion,'Color','red');
figure;
imshow(objectImage);
title('Red box shows object region');

points = detectHarrisFeatures(rgb2gray(objectFrame), 'ROI',objectRegion);

pointImage = insertMarker(objectFrame,points.Location,'+','Color','white');
figure;
imshow(pointImage);
title('Detected interest points');

tracker = vision.PointTracker('MaxBidirectionalError',1);

initialize(tracker,points.Location,objectFrame);

while ~isDone(videoFileReader)
```

```
        frame = videoFileReader();
        [points,validity] = tracker(frame);
        out = insertMarker(frame,points(validity, :),'+');
        videoPlayer(out);
end

release(videoPlayer);
release(videoFileReader);
```

Generate the video frame:

```
clear all; close all;

image_folder = 'C:\Users\arsen\Documents\GitHub\EECE5639_Computer-
Vision\Extra\hotel';
file_names = dir(fullfile(image_folder,'*.png'));
total_images = numel(file_names);
img = cell(1,total_images);


for k = 1:total_images
    File = fullfile(image_folder,file_names(k).name);
    img{k} = imread(File);
end
% for k = 1:total_images
%     imshow(img{k})
% end
outputVideo =
VideoWriter(fullfile('C:\Users\arsen\Documents\GitHub\EECE5639_Computer-
Vision\Extra','frame.avi'));
outputVideo.FrameRate = 5;
open(outputVideo)
for i = 1:total_images
   writeVideo(outputVideo,img{i})
end
% pointTracker = vision.PointTracker;
% [points,point_validity,scores] = pointTracker(outputVideo);
close(outputVideo)
%[interest_y,interest_x,m] = harris(img{k},1000);
```