Hot Topics in Computer Vision Project B

Radar Group: Weibo Qiu, Rucha Thorat, Kalyani Jayaprakash
Ramapriya Janardhan, Maosheng Tang

July 28, 2018

1 Introduction

Our Radar group aims at real-timely processing the radar occupancy grid video in order to accomplish vehicle localization in the given ground truth map. The available resources include radar occupancy grid video (as Figure 1 shows), Ground truth map (as Figure 2 below indicates) and ego motion data. Notice that the radar video frame's resolution is 2000*2000, while ground truth map's resolution is 1000*1000. However as can be seen in Figure 1, the radar video frame contains most green area which is regarded as redundant data. Therefore it is important to capture the essential information from radar video frame and then apply matching with ground truth map. In addition to matching two figures, ego motion data also contributes to offering velocity, acceleration and rotation rate information in order to improve the accuracy of localization. However, the sample rate of the ego motion data differentiates from the radar video frame frequency, thus it is hard to accumulate ego motion data.

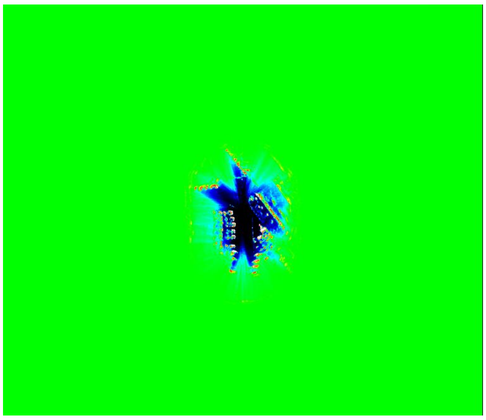

Figure 1: Radar occupancy grid video frame

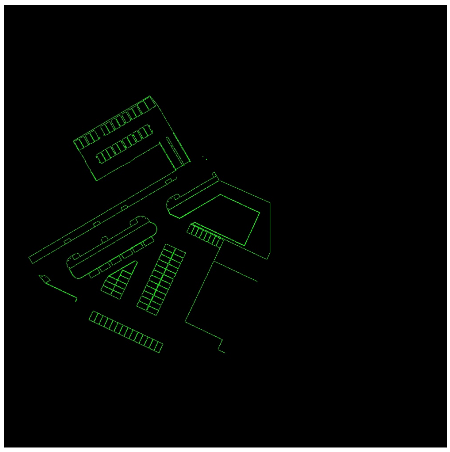

Figure 2: Ground Truth Map

After accomplishing radar data processing, the fusion group requires vehicle coordinates (in the ground truth map), orientation and covariance matrix from us. By collecting all the information from radar group, lidar group and camera group, the fusion group is capable of identifying more accurate car localization in the given ground truth map utilizing Kalman Filter.

2 Literature Review

From a imaging processing point of view, our task is to match the shape extracted from the radar image (a frame of the radar sensor video) to the ground truth map, so that we can obtain the location of the vehicle in the latter. Earlier, Hough Transform, invented by Duda and Hart [DH72], was used to detect curves in a binary image, by mapping the points on the curves to the parameter space, for example, a line can be represented by its intercept and slope and the 2d intercept-slope plain is the parameter space for detecting lines. The limit of this method lies in the fact that, it can not be used to complex situations, like a non-analytic shape in a gray-scale image. Later, D. Ballard proposed in [Bal87] an evolved version of the Hough Transform, which they call

"Generalized Hough Transform", that can be used universally to detect arbitrary shapes in a given image. The core of this algorithm is the construction of an R-table, in which the mapping from image space to general Hough transform space is stored. The algorithm can be briefly described as: for each boundary point in the query image, compute the gradient direction and the offset to a reference point, and store the offset as a function of the gradient direction. So, for each row in the R-table, it stands for a certain direction, and in each row, a series of offset vectors is stored. In practice, we divide a total of 2π possible gradient directions into n parts, and each computed direction is then found in one of these sections. The larger the value n, the more accurate the result, but the higher the computational cost. In recognition phase, for each edge pixel $\mathbf{x}$ in the ground truth image, increment all the points $\mathbf{x}+\mathbf{r}$ in an accumulate array, where $\mathbf{r}$ is the offset vectors found in the corresponding row of the R-table, the maxima in the accumulate array indicate the potential instances of the desired shape. To account the scale and orientation of the shape, the R-table should be multiplied by a scaling factor or be circular shifted and multiplied by a rotation matrix accordingly.

Not only conduct literature research on Generalized Hough Transform, we also applied it to our project and obtained some preliminary results. However, this method also requires the template picture(in our case radar video frame) being smaller than the test picture (in our case ground truth map). In order to evaluate the result of this algorithm, we manually crop the central part of one radar video frame which contains all the necessary information and tried to match it to the ground truth map. After implementing the algorithm, some trial results can be seen in Figure 3. It returns 4 possible results including the Scale, Angle and Position information. One of these outcomes might be a candidate since one line is perfectly match to the line in the ground truth map, as pointed in the circled area in Figure 3. But one of the problems is that we are not able to identify which one out of these four results is the most possible location of the car.

Additionally in this trial implementation, we have divided 360 degrees into 36 regions, each of which contains 10 degrees. In this case, 36 sets of R-tables were established to save all the offset vectors. The running time of one experiment is about 2.5 minutes with average memory consumption of 500MB. As the trial result indicates, the computational cost is relatively high, it is not applicable to use the algorithm in a real-time situation.

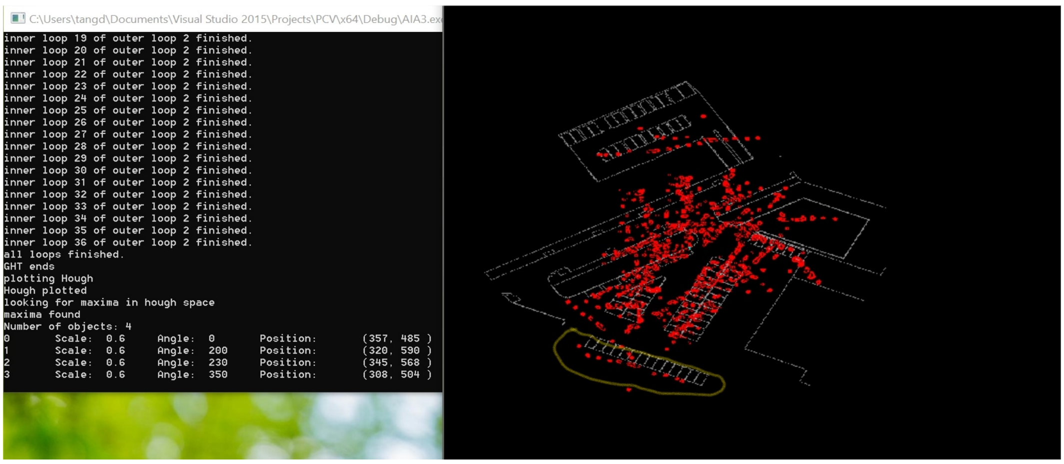

Figure 3: Generalized Hough Transform Results

As discussed, the inherited shortcoming of this algorithm is creating multiple R-tables as well as scale vector, which consumes relatively high computational effort. Inspired by the method and to overcome its drawback, a more direct way is to do template matching. This technique computes the cross correlation between the query image and the ground truth image, with the highest output being the area where the two images matches. The cross correlation computation can be quite fast in the frequency domain, for example, see [Lew95], fast processing speed is hence expected. This function has been implemented in OpenCV and is ready-to-use.

3 Implementation

We divided the workflow into several stages as illustrated in Figure 4: First we extract the radar images frame by frame from the given radar occupancy video data, followed by image pre-precessing in the hope of getting some shapes that will be helpful in the next stage, where the processed images are matched to the ground truth map. Eventually, We will obtain some positions being the possible location and orientation of the vehicle. As a final step, we integrate our work with the Cassandra platform so that senor fusion and other further development can be conducted.

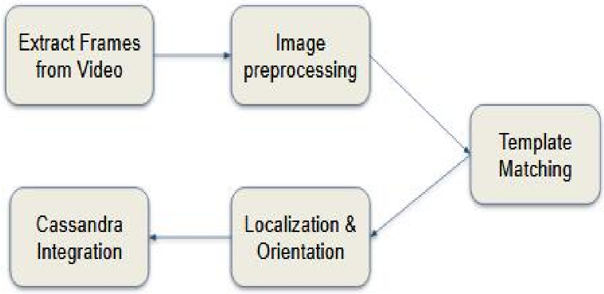

Figure 4: General Workflow

3.1 Dataflow through Cassandra

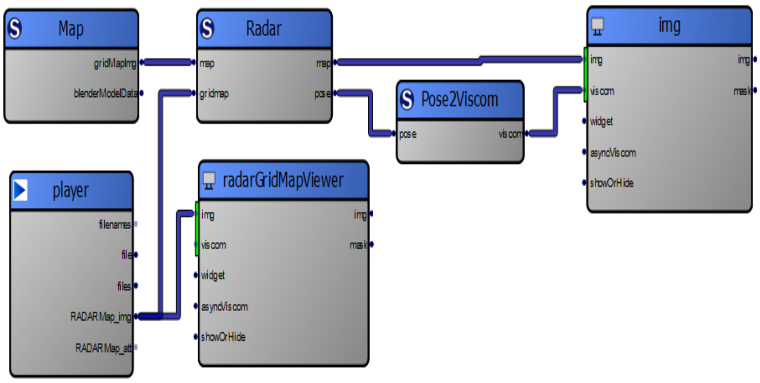

Figure 5: Cassandra graph for radar

The radar grip occupancy video is loaded in Cassandra and each radar frame extracted using player station (Figure 4). Such one radar frame and the ground truth map is given as a input to our implementation to process frame and matching it with ground truth map. Once matching is done, we come up with the co-ordinates of the car and orientation, which are then passed for sensor fusion through Pose2viscom station. In this way we integrate our implementation with Cassandra.

3.2 Image Preprocessing

The frames from radar video are preprocessed before performing localization. Cassandra sends video frames in CImage format. Before we begin processing the frame, we convert frame to Opencv Mat format. Since the frames extracted directly from the video contain a lot of irrelevant and misleading information, like the colors and noises etc., meanwhile, the images are larger than the ground truth map which is not possible for matching a larger image to a smaller one, therefore a sequence of image preprocessing is essential, before we "feed" to image to the template matching function.

Filter color - In the occupancy grid map, red pixels represent area where the obstacles are with the highest probability. We can thus get rid of other colors in the occupancy grid map to reveal the most relevant shape detected by the radar sensor. We identified range for red colored pixels and eliminated all the pixels which didn't belong to this range. The frame after filtering red color is shown in figure 7.

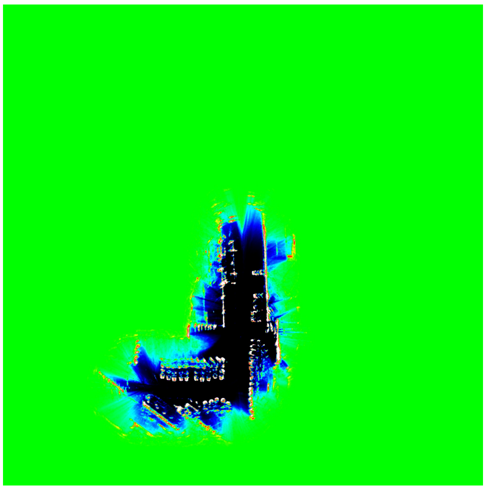

Figure 6: One frame extracted directly from the radar video data

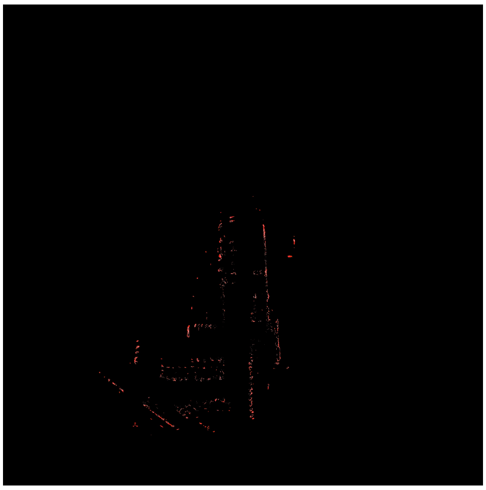

Figure 7: After removing irrelevant colors, the remaining gives us a somewhat concrete shape which will be further processed

Canny edge detection - In order to enhance the quality and appearance of the frame after removing unnecessary colored pixels, we used Canny edge detection algorithm. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. After enhancing frame using canny edge detection algorithm, we got frame as shown in figure 8.

Resizing - The resolution of ground truth map is 1000*1000 pixels and the resolution of radar frame is 2000*2000 pixels. But maximum part of the frame is blank (most of the pixels are not colored) except red colored pixels. In order to keep only the region of interest, that is part of frame which only covers all the colored pixels, we resized the frame to not include pixels which are not colored. This process reduced the frame size and prevented errors that could occur during template matching due to differences among resolutions of ground truth map and radar frame. We also calculated the new car location after resizing as we needed it to calculate car location in ground truth map (car's location was fixed in unprocessed frame (x=1000,y=1000)). The frame after resizing is shown in figure 9

Figure 8: Frame after canny edge detection

Figure 9: Frame after resizing

3.3 Template Matching

After preprocessing, radar frames are matched against ground truth map, to find the car location on ground truth map using Opencv's matchTemplate function. TM COEFF template match method was used to perform matching. TM COEFF was turned out as the best matching method when we compared it against other template matching methods such as SQDIFF NORMED,TM CCORR,TM CCORR NORMED and TM COEFF NORMED. The template matching method returned coordinates on ground truth map where the radar frame matched. From these co-ordinates, we determined the car location on ground truth map. The matching method also returned intensity value for matching.

The co-ordinates on ground truth map were not accurate when we did template matching without modifying the resized radar frame. Hence we decided to rotate the radar frames and perform template matching. Each radar frame was rotated 36 times with rotation degree of 10 in each rotation. When each rotation was performed, the rotated frame was matched against ground truth map. The coordinates of car and intensity of matching were recorded for each rotation. After 36 rotations, the frame with best intensity (maximum intensity), rotation angle, and coordinates were determined. A covariance matrix was also determined using rotations, intensities and coordinates of each frame. The results were then sent to fusion group for further processing.

4 Results and analysis

Our implementation yielded the localization of the car on the ground truth map with car's position as x and y co-ordinates. The orientation of the car is obtained by the rotation angle.Our implementation ensures to select one best possible match for a particular frame. As we are passing only single set of output for any frame, our co-variance matrix is static.

In the template matching method, we are incorporating the rotation of the radar frames. So, for every single frame rotated 36 times and processed for each rotation, a total time of 14 seconds/frame was observed. This compromise has been included for attaining more accuracy for matching radar image perfectly with ground truth map. The processing time could be reduced by minimizing the number of rotations but then there is a trade off with the accuracy of output.

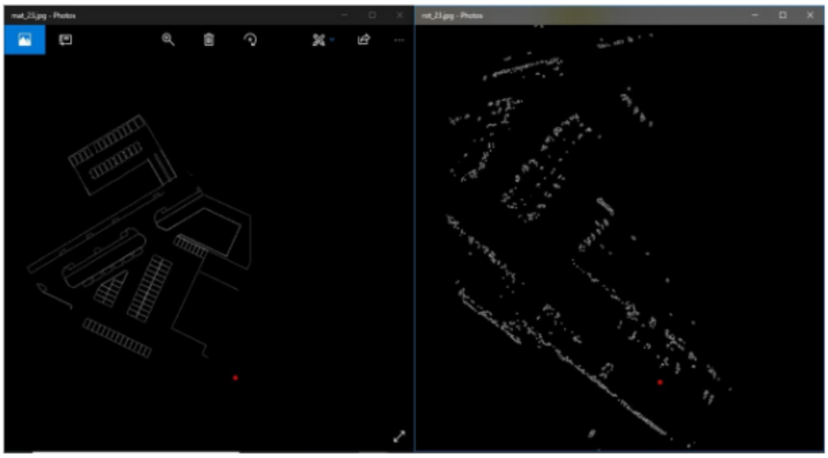

Figure 10: Localization using template matching

Figure 10 illustrates an example outcome of our implementation. One of the rotated radar frame (right) is matched with the ground truth map (left) to identify the location of the car. The angle at which the frame is rotated gives the orientation of the car. For visualization of the position of the car is marked in red.In this way we deliver the position and orientation of the car for every frame.

The radar video is noisy in the beginning and data from initial few frames are uncertain. This results in less reliability of radar sensor data in the early stage. This drawback is considered while doing sensor fusion. Even after the radar data in the video become stable, extracted occupancies from the frames contain noise due to which contouring does not give proper shape. This makes it difficult to guarantee the correctness of template matching.

As mentioned, the co-ordinates, orientation of the car in ground truth map as well as static co-variance matrix are passed for sensor fusion through Pose2viscom station. The further radar localization result in Cassandra has not been presented in this report, since the Fusion group has to merge all the results from Radar, Lidar and Camera groups in order to deliver a more accurate results (localization and orientation of the car).

5 Conclusion

Radar sensors have long been used in the military and industry to detect objects or obstacles. It is relatively cheaper than the more advanced Lidar sensors and reliable under different working conditions, such as in the dark, in which case a camera would be very likely fail to capture anything. However the shortcomings of radar sensor, including but not limited to, noisy imaging, not enough detection at early stage (first 5-10 seconds), are apparent. Additionally, the radar sensor has no clue about what kind of obstacles it has detected compared to Lidar. Due to these shortcomings, it is hard for us to use the radar imaging alone to accurately locate the car on the ground truth map via the template matching algorithm. Considering time limits, the processing speed and localization accuracy of this project are not optimal as well and can be further improved in the future.

References

[Bal87] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. In *Readings in computer vision*, pages 714–725. Elsevier, 1987.

[DH72] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, Jan 1972.

[Lew95] John P Lewis. Fast normalized cross-correlation. In *Vision interface*, volume 10, pages 120–123, 1995.