Comments from Dr. Consi:

1. Overall a good report and a good project
2. Explain Fig. 3
3. Explain lower panel of Fig. 6. What is color code? Needs axes labels.
4. Fig. 7: which is stereo depth matching and which is Lidar? What is color code? Needs axes labels.
5. Fig. 8: left and right panels look the same to me.
6. In general, refer to figures in your text.

Grade: 63/65

VISUAL ODOMETRY

USING KITTI DATASET

PRESENTED BY

SHREEJIT DESHMUKH

KANDIRAJU VENKATA SAI ADVAITH

KINGSLEY NWOSU

BASIL REJI

KEVIN SANI

PRUDHVI RAJASEKHAR NAIDU KONTHALA

Contents

INTRODUCTION

Our project aims to perform visual odometry using Kitti dataset, i.e., we will calculate poses using stereo matching and then compare the pose with ground truth to estimate the error. We are also performing LIDAR corrections in the dataset to see the improved accuracy. The data collected by Karlsruhe Institute of Technology was done by using the autonomous driving platform Annieway. Sensors used for collecting this data are stereo cameras- grayscale and RGB, Velodyne, IMU/GPS pair. The setup and configurations for each are shown in the figure below.

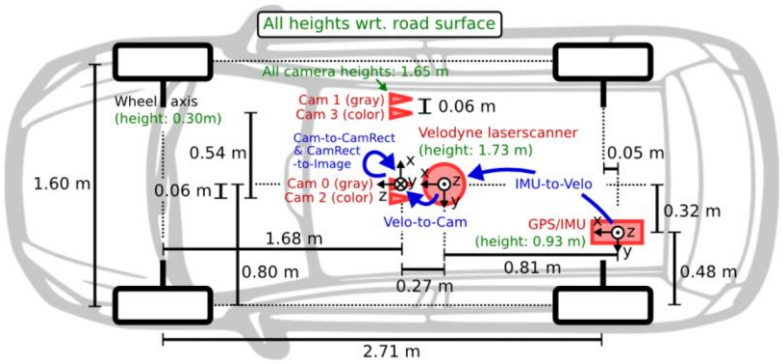

Table 4. RT common specifications

Parameter	Specification
Calculation latency	1 ms
Operating temperature[1]	-40° to 70 °C
Vibration	0.1 g^2/Hz 5–500 Hz
Shock survival	100 *g*, 11 ms
Internal storage	32 GB

Sensor:	• 64 lasers • 360 degree horizontal field of view (azimuth) • 0.09 degree angular resolution (azimuth) • 26.8 degree vertical field of view (elevation) • <5 cm resolution (distance) • 5-15 Hz field of view update (user selectable) • 50 meter range for pavement (~0.10 reflectivity) • 120 meter range for cars and foliage (~0.80 reflectivity) • >1m points per second • <0.05 miliseconds latency
Laser:	• Class 1M - eye safe • 4 x 16 lasers assemblies • 905 nm wavelenth • 5 nanosecond pulse • Adaptive power system for minimizing saturations and blinding • Beam Shape: Lower Block @ 100 feet: 6" x .8" (w x h) Upper Block @ 100 feet: 4" x .8" (w x h)
Mechanical:	• 12-16V input @ 4 Amps (max) • 10" tall cylinder of 8" OD radius
Output:	• 100 MBPS Ethernet with UDP

Point Grey Research Flea2 FL2-14S3M Specifications

Technical Specifications	
Scan Type	Area scan
Max Frame Rate	15.6 Hz
Area Scan Type	Progressive area scan
Video Color	Monochrome
Interface Type	IEEE 1394b
Digital Video Pixel Depth	12 bits/ch
Sensor Type	CCD
Support Status	NI Supported
Camera Status	Current

Fig. 1. – Setup of sensors from left to right i. Co-ordinate frame ii. Inertial measurement system iii. LIDAR iv. Grey scale image

The data is collected in midsize city Karlsruhe in Germany, and it is collected in sequences of twenty-two for convenience of data management. The ground truth data is available for the first eleven datasets, out of which we are using '02' dataset for our analysis.

COORDINATE FRAME AND PROJECTION MATRICES

The data collected from individual sets of sensors must be transposed to the global co-ordinate frame, which in our case is the left greyscale camera (position can be seen in Fig. 1). For that we need projection matrices which are provided in the dataset. These are in the form of homogenous matrices.

0	1	2	3	4	5	6	7	8	9	10	11	12
P0:	718.856000	0.000000	607.192800	0.000000	0.000000	718.856000	185.215700	0.000000	0.000000	0.000000	1.000000	0.000000
P1:	718.856000	0.000000	607.192800	-386.144800	0.000000	718.856000	185.215700	0.000000	0.000000	0.000000	1.000000	0.000000
P2:	718.856000	0.000000	607.192800	45.382250	0.000000	718.856000	185.215700	-0.113089	0.000000	0.000000	1.000000	0.003780
P3:	718.856000	0.000000	607.192800	-337.287700	0.000000	718.856000	185.215700	2.369057	0.000000	0.000000	1.000000	0.004915
Tr:	0.000428	-0.999967	-0.008084	-0.011985	-0.007211	0.008081	-0.999941	-0.054040	0.999974	0.000486	-0.007207	-0.292197

Fig. 2. – Projection matrices used for transforming data

THE APPROACH

So, to calculate the pose change in frames, we calculate the feature position change in between frames – the angles and the distance in terms of transformations. We inverse these transformations to estimate our car's position change. To better estimate this change we need the depth of our features which we calculate using disparity matching using SGBM algorithm. After calculating depth using

Stereo matching, we extract the scale invariant features using SIFT algorithm. Then we figure out the transformations of the features between consecutive frames to estimate the pose change in our vehicle. The techniques and algorithms used are described below.

SGBM (SEMI GLOBAL BLOCK MATCHING):

This example shows how to compute disparity between left and right stereo camera images using the Semi-Global Block Matching algorithm, which is suitable for implementation on an FPGA. Distance estimation is an important measurement for applications in Automated Driving and Robotics. A cost-effective way of performing distance estimation is by using visual odometry. With a stereo camera, depth can be inferred from point correspondences using triangulation. Depth at any given point can be computed if the disparity at that point is known. The example computes disparity using the SGBM method, which is an intensity-based approach that generates a dense and smooth disparity map for good 3D reconstruction.

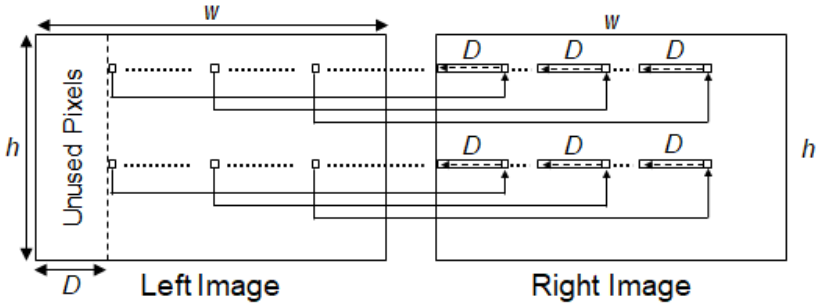

Fig. 3 – Reference for disparity

SIFT (SCALE INVARIANT FEATURE TRANSFORM):

Scale-Invariant Feature Transform (SIFT) is a computer vision technique for identifying and describing local features in pictures. In applications like object recognition and image matching, it enables the detection of localized characteristics in pictures, which is crucial. Utilizing the pixel depth, it recognizes characteristics in the first picture and compares them with matching features in the second image to rebuild the 3D location of the item in the first camera frame. SIFT can carry out feature identification independently of the image's perspective, depth, and scale, in contrast to the Harris Detector, which depends on these factors. The picture data is transformed into scale-invariant coordinates to do this.

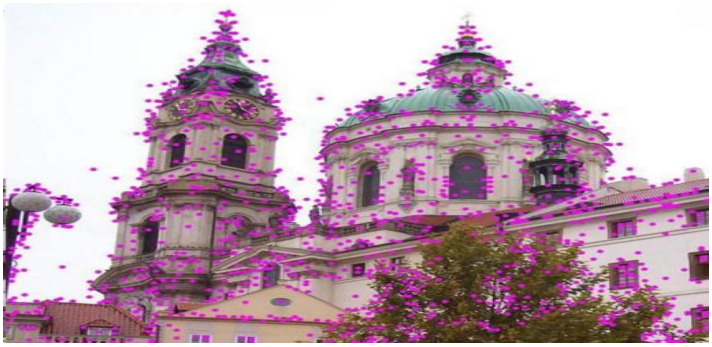

Fig. 4. – Key points identified by Sift in a program

Key points and interest points are the same thing. What makes a picture intriguing or stand out are specific spatial positions or spots within the image. They are resistant to changes in picture size, rotation, translation, distortion, and other factors. Using a technique called the Difference of Gaussian (DoG) pyramid, SIFT critical points are sought across several picture sizes. By choosing local maximum points in the 3D vicinity of the picture scale pyramid, key spots are chosen. To achieve rotation invariance, the prevailing orientation for each key point is identified. To describe the local appearance of each chosen key point, SIFT descriptors are produced as a histogram of picture gradients around the key points. Because of efficiency and application of our project we have used this algorithm for feature detection and stereo depth estimation.

RESULTS

The results are divided into two sections:

1. Individual algorithm analysis - where algorithms like SGBM, SIFT feature detection are implemented in the consecutive frames to analyze the efficiency.

2. Results and conclusion – where the results for the whole sequence are calculated and then compared with the ground truth poses for error estimation.

SECTION 1

The disparity is calculated between 1st pair of left and right image and shown as below –

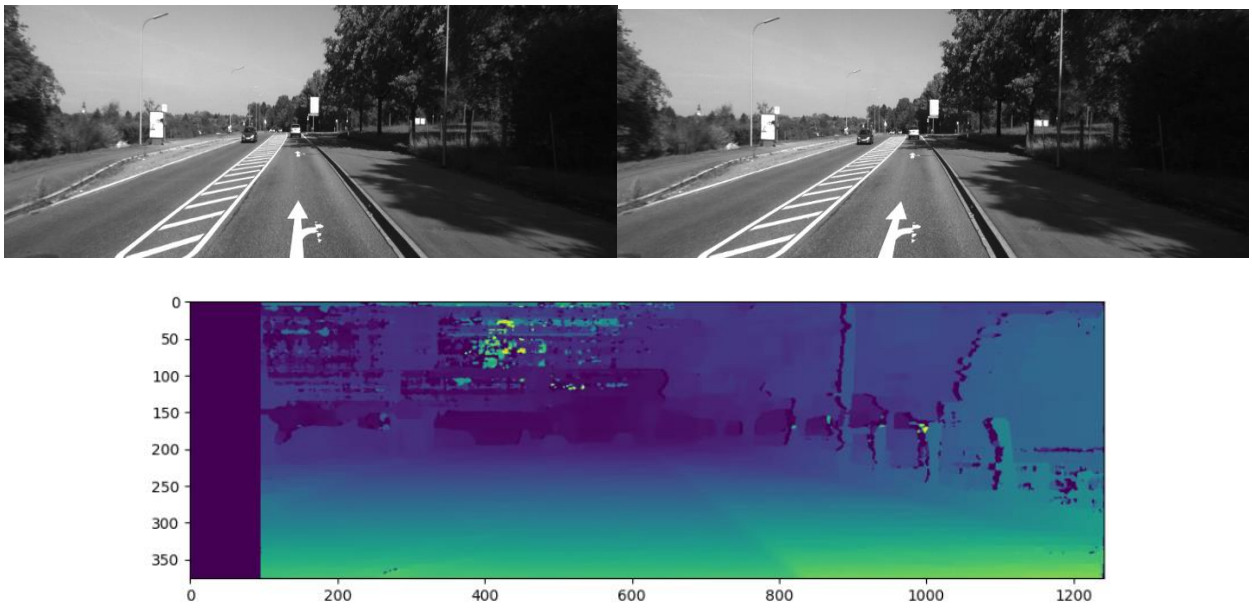

Fig. 6. – Disparity matching using SGBM

The depth is calculated using Stereo Matching. After depth calculation using Stereo matching, we match the results with LIDAR for visual error estimation for the same frame of reference.

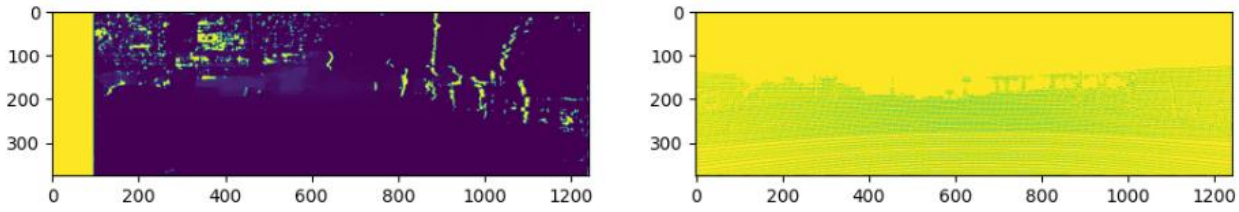

Fig. 7 – LIDAR and Stereo depth matching comparison.

```
location: [ 127 1220] stereo/lidar depth: [12.53208276 11.82472166]
location: [ 127 1224] stereo/lidar depth: [12.53208276 11.83043835]
location: [ 127 1228] stereo/lidar depth: [12.45628387 12.30383734]
location: [ 128 1207] stereo/lidar depth: [13.14535489 12.46581184]
location: [ 128 1210] stereo/lidar depth: [12.87149333 12.42755454]
location: [ 128 1214] stereo/lidar depth: [12.73879753 12.44825056]
location: [ 128 1216] stereo/lidar depth: [12.6088098  12.33368438]
location: [ 128 1232] stereo/lidar depth: [12.45628387 12.63933096]
location: [ 128 1235] stereo/lidar depth: [12.45628387 13.07676327]
location: [ 128 1239] stereo/lidar depth: [12.43122093 13.37027506]
location: [ 129 1171] stereo/lidar depth: [13.00698274 12.67090896]
location: [ 129 1175] stereo/lidar depth: [13.03442363 12.68862293]
location: [ 129 1179] stereo/lidar depth: [13.06198055 12.62237577]
location: [ 129 1186] stereo/lidar depth: [12.84473347 12.78569539]
location: [ 129 1190] stereo/lidar depth: [12.87149333 12.68347056]
location: [ 129 1193] stereo/lidar depth: [12.92534895 12.76712945]
location: [ 129 1195] stereo/lidar depth: [12.92534895 12.82444416]
location: [ 129 1199] stereo/lidar depth: [12.95244612 12.68425236]
location: [ 129 1203] stereo/lidar depth: [13.14535489 12.65098168]
location: [ 130 1169] stereo/lidar depth: [13.06198055 12.78448432]
location: [ 130 1182] stereo/lidar depth: [13.03442363 13.65244003]
location: [ 131 1124] stereo/lidar depth: [13.34409676 13.15099901]
location: [ 131 1125] stereo/lidar depth: [13.34409676 13.15934796]
location: [ 131 1129] stereo/lidar depth: [13.31533793 13.093095  ]
```

From the table we can see that there is difference in the values of LIDAR and stereo depth estimation. As we know that LIDAR depth values can be more trusted since they are 1st order values, but stereo matching depth is 2nd order derived from stereo matching. We'll estimate the pose using

1. Without LIDAR correction
2. Using LIDAR correction

And estimate the error difference between the two.

SECTION 2

PART A - We calculate the pose estimate using our Stereo matching data only

PART B – Calculating pose estimation after integrating LIDAR data

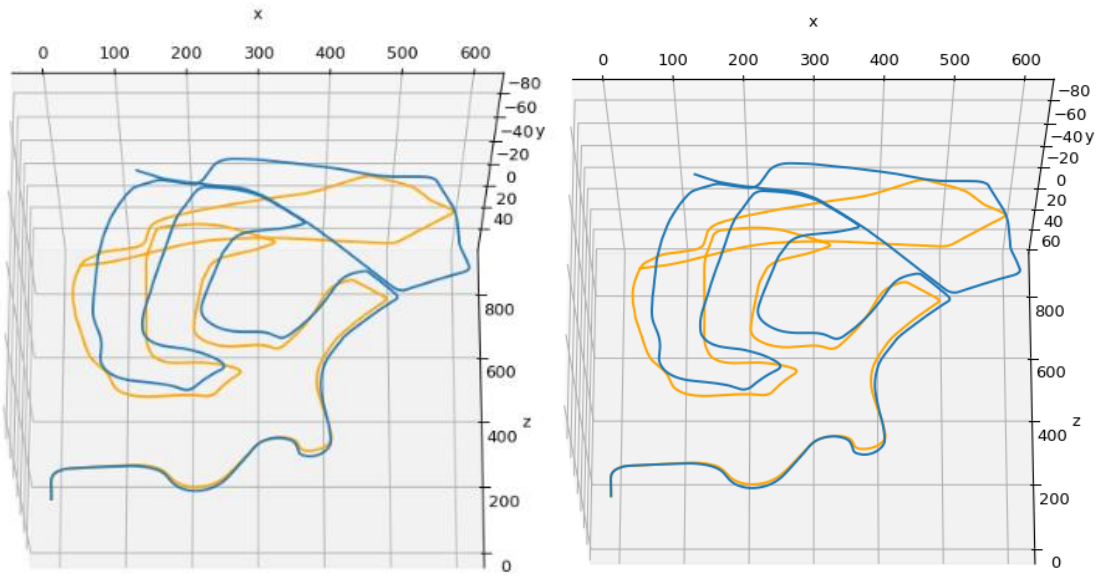

Fig. 8 – Part A (left) and Part B (right)

Ground Truth in dark blue

Estimated pose in orange

All units in meters

{'mae': 50.69206348100899, 'rmse': 62.34278295009286, 'mse': 3886.62258596238

mae: Mean Error Estimate

rmse: Root Mean Square Error

mse: Mean Square Error

{'mae': 50.65598853114764,
'rmse': 62.26798451041466,
'mse': 3877.3018949892394}

mae: Mean Error Estimate

rmse: Root Mean Square Error

mse: Mean Square Error

The results with LIDAR infused are better than we expected. The correction in value is not huge, but conclusive to our assumption. The approach adopted for visual odometry was simple but from scratch hence the error which is around ~60 meters is not applicable in real world. However, we can increase the accuracy using advanced filtering techniques for LIDAR and Stereo matching, for example, Particle filtering, Extended Kalman filtering, etc. The accuracy can be further increased by loop closures and implementing corrections.

Our project aimed to estimate the localization of our vehicle using classical methods of stereo matching and implementing corrections using LIDAR data and observe the accuracy of our implementation. We can conclude that this preliminary approach needs additional filtering for real world applications

REFERENCES

- P. Alcantarilla, L. Bergasa, and F. Dellaert. Visual odometry priors for robust EKF-SLAM. In ICRA, 2010.
- S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. IJCV, 92:1–31, 2011.
- S. M. Bileschi. Streetscenes: Towards scene understanding in still images. Technical report, MIT, 2006.
- J.-L. Blanco, F.-A. Moreno, and J. Gonzalez. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. Auton. Robots, 27:327–351, 2009
- G. Bradski. The opencv library. Dr. Dobb's Journal of Software Tools, 2000.
- T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In ECCV, 2004.
- T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. PAMI, 33:500–513, March 2011
- M. E. C. G. Keller and D. M. Gavrila. A new benchmark for stereo-based pedestrian detection. In IV, 2011
- J. Cech, J. Sanchez-Riera, and R. P. Horaud. Scene flow estimation by growing correspondence seeds. In CVPR, 2011.
- J. Cech and R. Sara. Efficient sampling of disparity space for fast and accurate matching. In BenCOS, 2007.