Implementation of Stereo Odometery

Using Careful Feature Selection and Tracking

Mayank Mittal, **Ritwik Bera**, Prof. Gaurav Pandey

Indian Institute of Technology, Kanpur

Table of contents

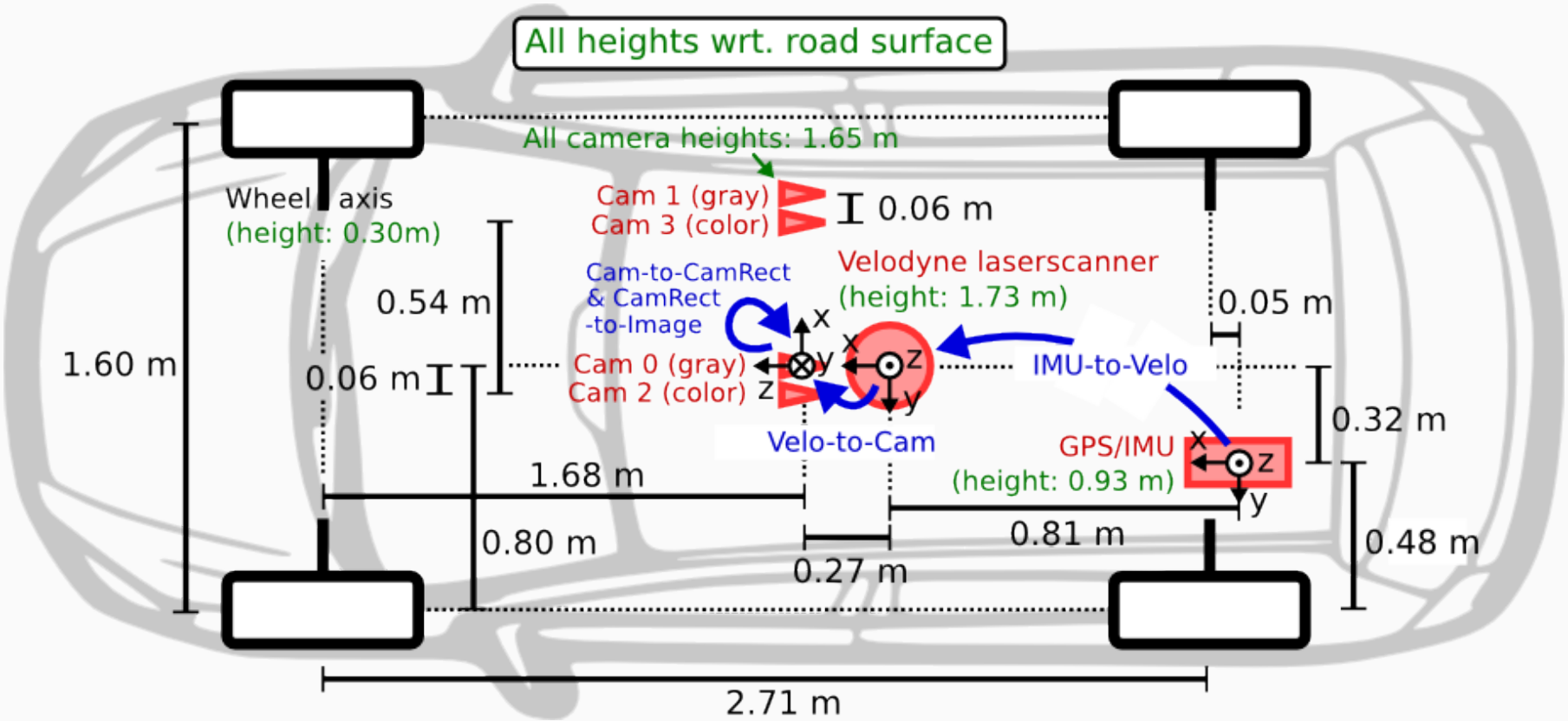
Problem Formulation

What is Stereo Odometry?

- **Odometry** is the process of estimating a robot's pose incrementally, relative to its surrounding
- Most common implementation: using *wheel encoders*
- If surrounding is texture- rich, **visual odometry** is considered the most accurate method
- Using a pair of calibrated cameras for odometry is called **stereo odometry**

KITTI Sensor Setup (Top View)

- 2 Grayscale cameras, 1.4 Megapixels: Point Grey Flea 2 (FL2-14S3M-C) [4]



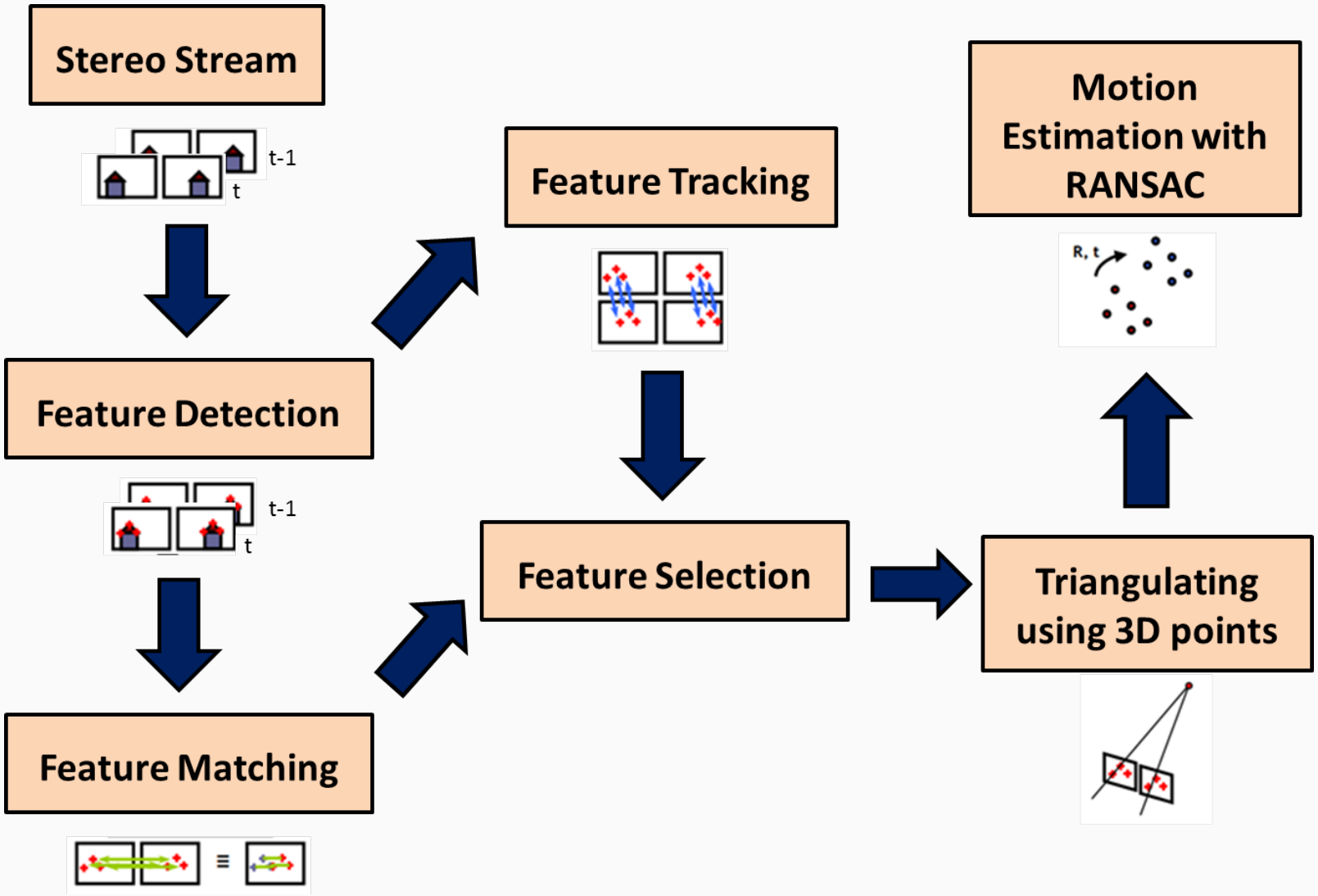
Known Parameters

Before running the stereo odometry algorithm, the following parameters need to be known using calibration:

- Intrinsic parameter matrices for the stereo pair: K_l, K_r
- Extrinsic parameter matrix for the stereo pair: $[R_{rl}|\, T_{rl}]$

Note: The KITTI Dataset used provides synced rectified images. In order to implement real time, we need to consider the radial and tangential distortion for each camera in the stereo pair as well

Overview of Algorithm

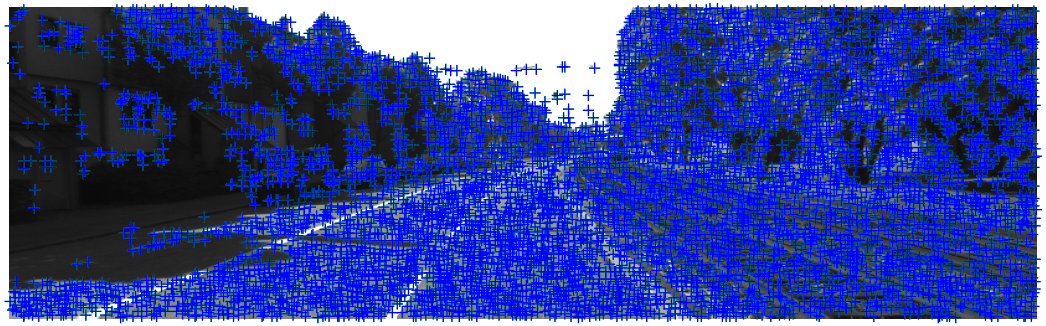


Feature Tracking and Selection

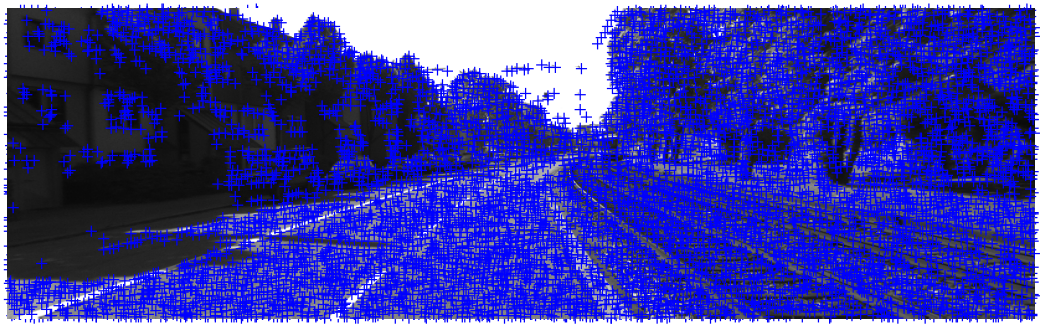
Feature Detection

- Features in an image can be points, edges or blobs, i.e. locations in an image which differ from the immediate surroundings
- **Minimum Eigenvalue Feature Detector** (by Shi- Tomasi)
  - Compute gradient (I_x, I_y) at each point of an image
  - Compute auto- correlation matrix $H = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$ from entries in gradient, where w is the weights set over a window W
  - Compute eigenvalues of H
  - Select points with response (λ_{min}) greater than threshold
  - Apply non- maximal suppression to attain points where λ_{min} is a local minima as features

Feature Detection



(a) Feature points detected in left camera at instant t



(b) Feature points detected in right camera at instant t

Feature Matching

- Correspondences between feature points in a given pair of images determined using Sum-of-Absolute-Differences (SAD)
- In conjunction with SAD, circular matching is done reject outliers
  - Each feature in starting frame needs to matched between the left and right images of t and $t-1$
  - If the circle is not closed i.e. feature is not correctly matched across all frames then it is considered an outlier

Feature Matching

Consider feature in left image at t frame

Figure: *Source:* StereoScan: Dense 3d in Real Time by Andreas Geiger

Feature Matching

Apply SAD over a window to match the feature in left image at t-1 frame

Figure: *Source:* StereoScan: Dense 3d in Real Time by Andreas Geiger

Feature Matching

Apply SAD over epipolar line to match the feature in right image at t-1 frame

Figure: *Source:* StereoScan: Dense 3d in Real Time by Andreas Geiger [3]

Feature Matching

Apply SAD over a window to match the feature in right image at t frame

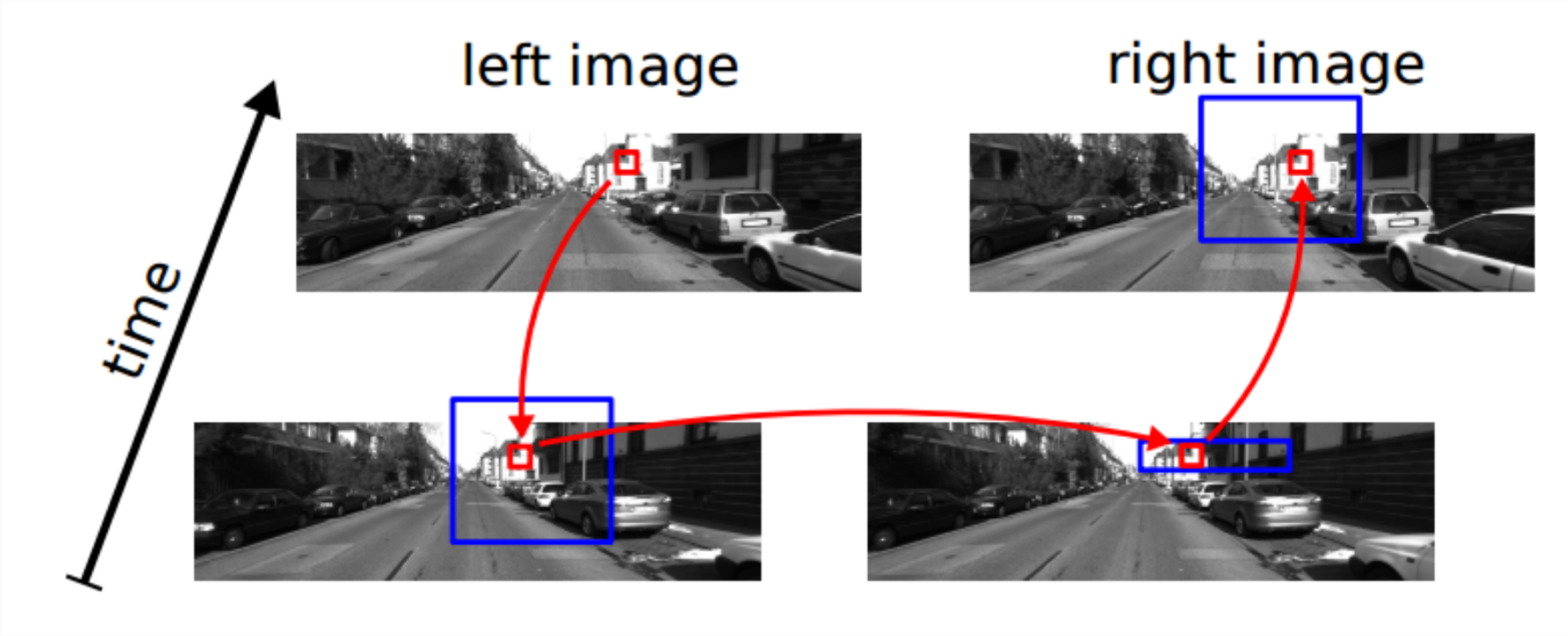


Figure: *Source:* StereoScan: Dense 3d in Real Time by Andreas Geiger [3]

Feature Matching

Apply SAD over epipolar line to match the feature in left image at t frame

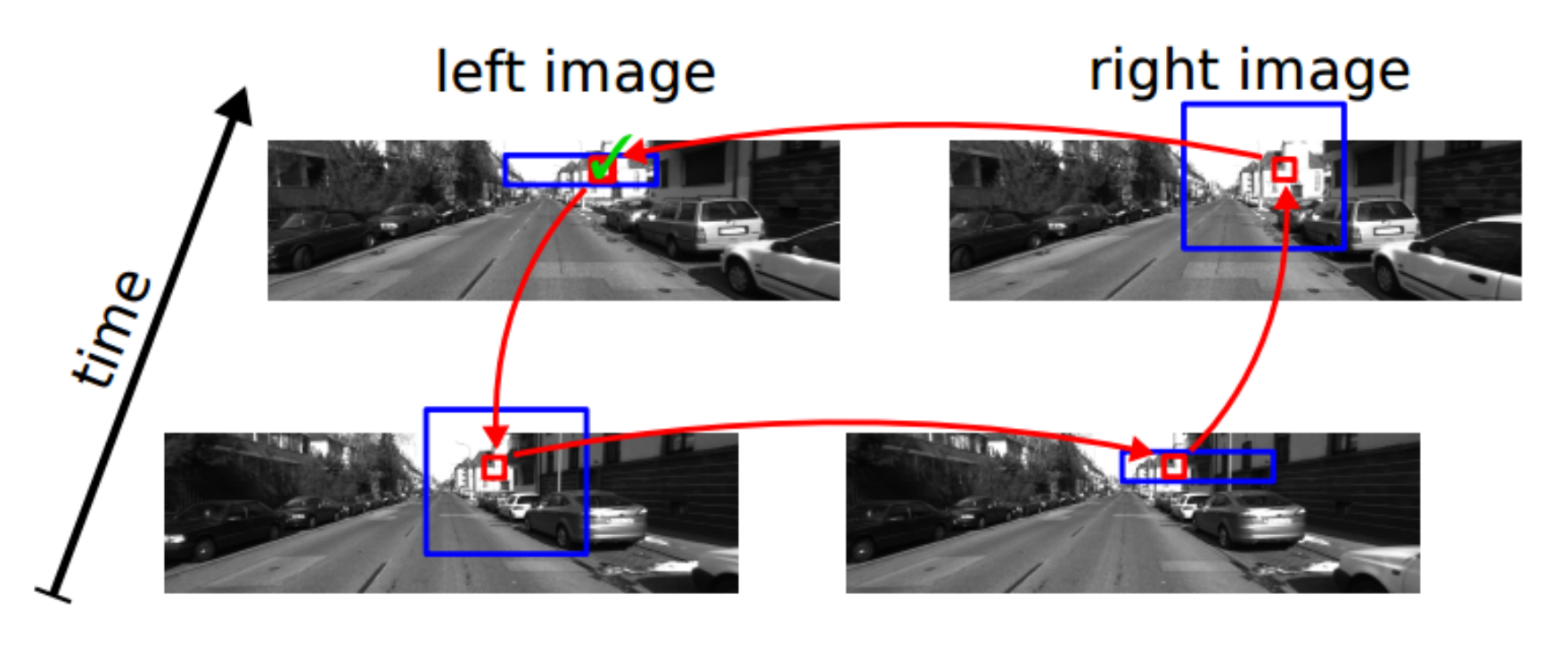


Figure: *Source:* StereoScan: Dense 3d in Real Time by Andreas Geiger [3]

Feature Matching

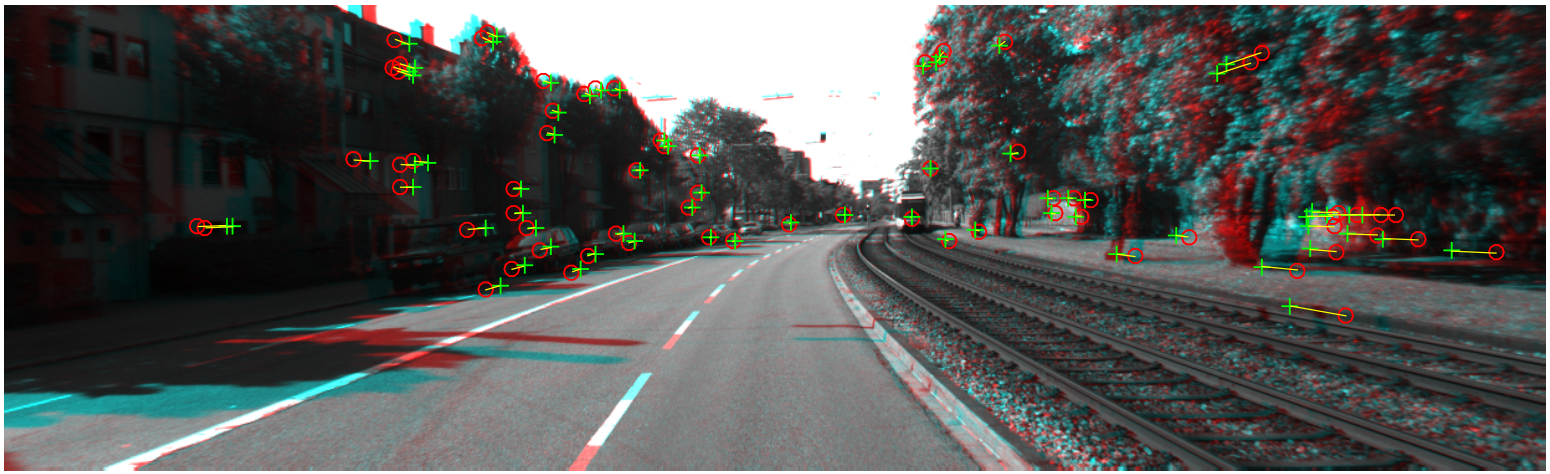


Figure: Matched features in left camera at instants t (features marked as green plus signs) and t-1 (features marked as red circles)

Feature Tracking and Selection

- Features tracked for longer duration are considered more reliable
  - Each feature is given an additional property *age* which is the duration for which the the feature is being tracked from its first occurrence
  - If a feature hasn't been tracked for more than three consecutive frames, it is discarded
- Using a subset of features for egomotion estimation provides better results and decreases computation time
  - Entire image is divided into 50×50 windows called buckets
  - In each bucket, only n of features selected on basis of feature strength and *age*

$$select(x, y) = \begin{cases} stronger(x, y), & \text{if } age(x) = age(y) \\ older(x, y), & \text{if } age(x) \neq age(y) \end{cases}$$

Feature Tracking and Selection

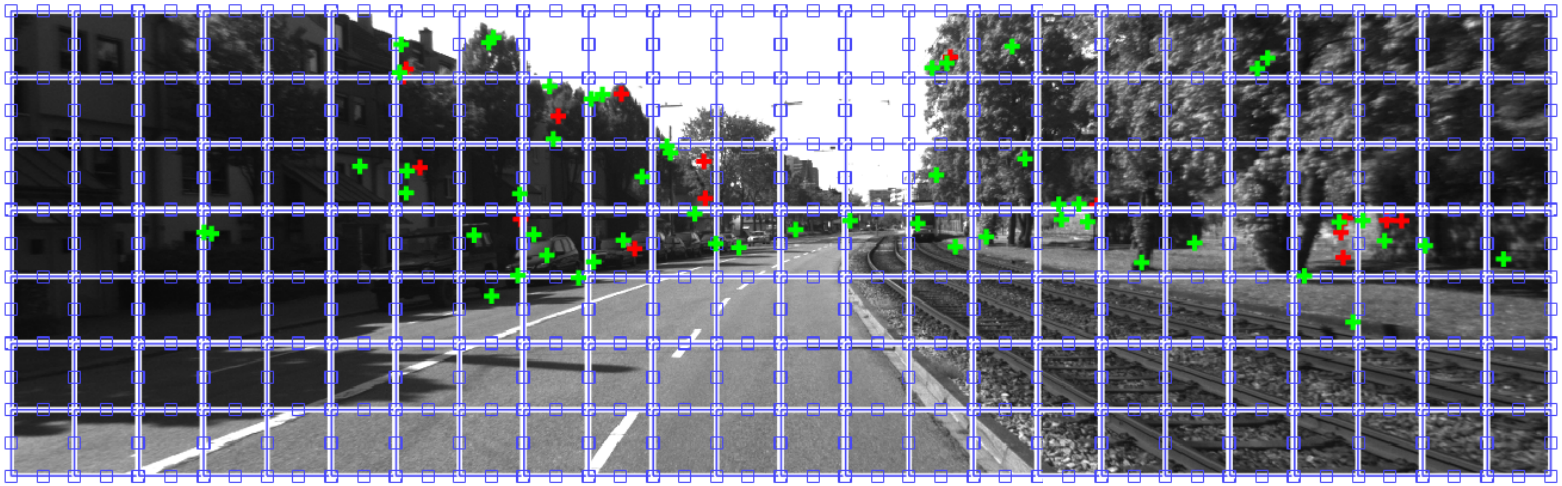


Figure: Feature location selected through bucketing are marked in green while the weak features in each bucket are marked in red

Egomotion Estimation

Rotation Estimation

- We assume that the surroundings (observed feature points) are static, and only the robot is under motion
- Rotation is estimated by applying Nister's Five Points Algorithm [5] in conjunction with RANSAC for the left camera only.
  - Five corresponding feature pairs from the left image at frame t-1 and frame t are selected randomly
  - Essential matrix E for this subset is computed using Nister's Five Point Algorithm
  - Essential matrix with maximum inliers is selected as best estimate $\hat{E}$
  - Rotation is calculated using singular value decomposition of estimated essential matrix $\hat{E}$

Translation Estimation

- Using the disparity map for the stereo pair images at frame t-1, 3D-point cloud is generated for each feature using triangulation

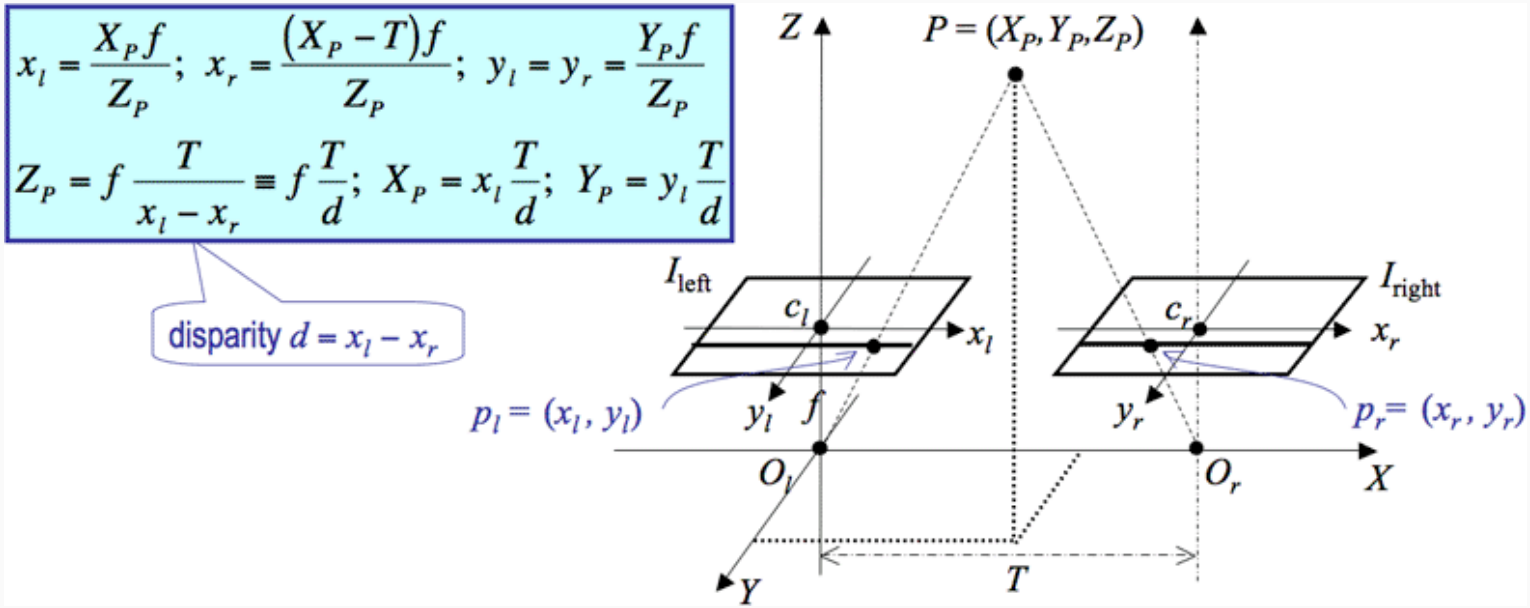
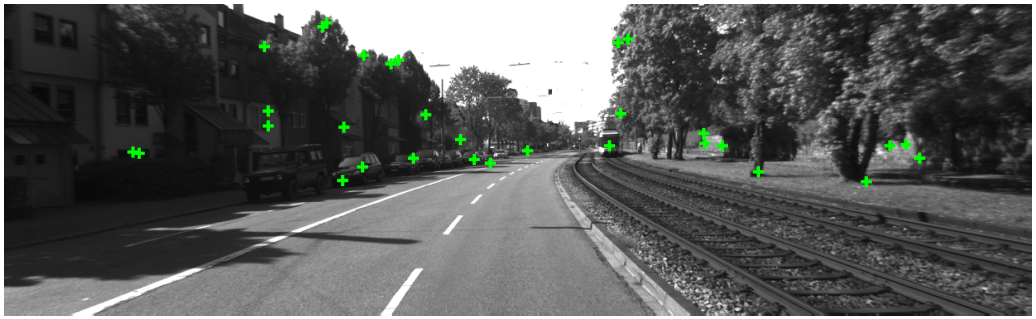


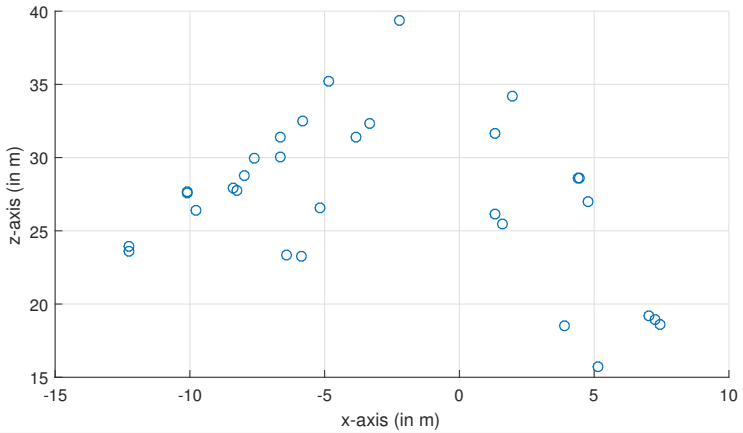
Figure: Triangulation of a 3D point in using camera's optical model

Source: `www.cs.auckland.ac.nz`

Translation Estimation



(a) Feature points detected in left camera at instant t-1



(b) Traingulated 3D points with respect to left camera frame at instant t-1

Translation Estimation

- We assume that the stereo pair (and hence, the robot) is static, and the surroundings is under motion
- 3D point could generated from the stereo pair at frame t-1 is reprojected to the image plane at frame t for each camera in the stereo pair, after a certain transformation (rotation and translation) which needs to be estimated [2].

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \pi(\boldsymbol{X}; R, t) = \begin{bmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Translation Estimation

- Error between reprojected 3D points onto the image planes of the stereo pair and the corresponding feature points observed in them at frame t is computed using L2 norm [1].

$$e = \sum_{i=1}^{n} \left\{ \|x_l - \pi(\boldsymbol{X_l}; R, t)\|^2 + \|x_r - \pi(\boldsymbol{X_r}; R, t)\|^2 \right\}$$

- Using the rotation matrix R estimated earlier, the error function needs to be minimized for finding the translation vector.
  - This is done iteratively using the quasi-newton method to find the minima of the error cost function.

Conclusion

Result

KITTI Dataset: City Sequence 01

Video: `https://youtu.be/AxtgUxlO3FY`

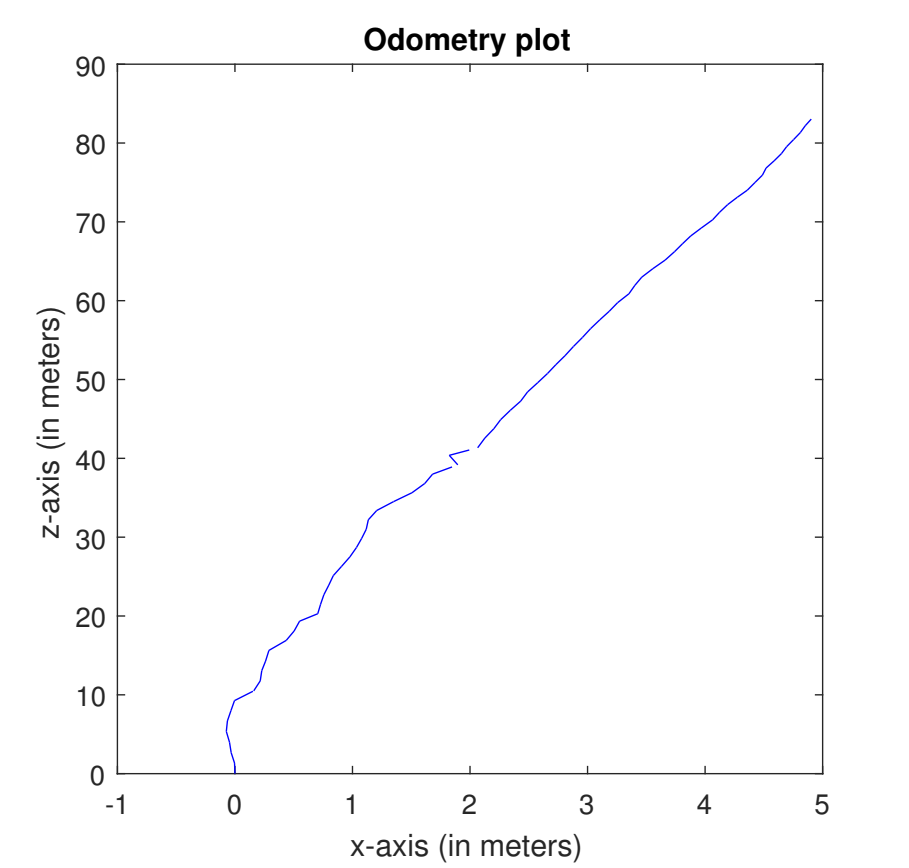


Figure: Estimated Trajectory of the left camera

Future Work

- Currently it takes of 6-7 seconds for computation in each time step. This can be made faster by implementing it on **OpenCV and ROS** (under process)
- Coupling stereo odometry with an IMU through Kalman Filter to increase the accuracy of rotation estimation

References I


I. Cvišić and I. Petrović.
Stereo odometry based on careful feature selection and tracking.
In *Mobile Robots (ECMR), 2015 European Conference on*, pages 1–6. IEEE, 2015.


A. Geiger, P. Lenz, C. Stiller, and R. Urtasun.
Vision meets robotics: The kitti dataset.
The International Journal of Robotics Research, 32(11):1231–1237, 2013.


A. Geiger, J. Ziegler, and C. Stiller.
Stereoscan: Dense 3d reconstruction in real-time.
In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. Ieee, 2011.

References II

B. Kitt, A. Geiger, and H. Lategahn.
Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme.
In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 486–492. IEEE, 2010.

D. Nistér.
An efficient solution to the five-point relative pose problem.
IEEE transactions on pattern analysis and machine intelligence, 26(6):756–770, 2004.

Questions?

Code Availability

MATLAB code available under MIT License

`github.com/Mayankm96/Stereo-Odometry-SOFT`

C++ code (under process)

`github.com/Mayankm96/SOFT_odom`

Feel free to take a look and contribute!

Nister's Five Point Algorithm

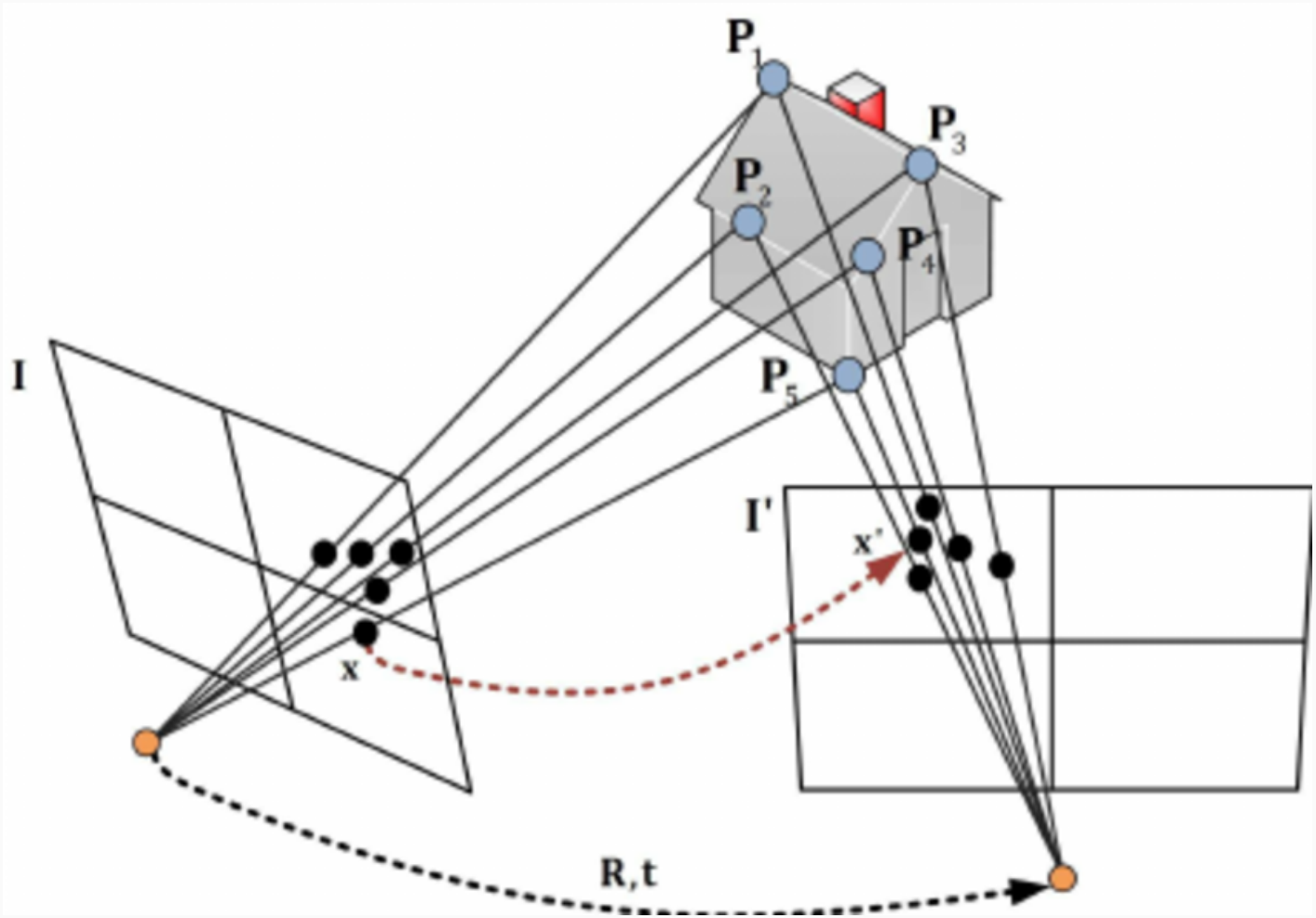


Figure: An illustration of the working of Nister's Five Point Algorithm

Nister's five point algorithm [5] exploits the following constraints to estimate the extrinsic parameters with just five pairs of points (instead of 8 in the eight-point algorithm).

$$2EE^TE - trace(EE^T)E = 0$$

$$det(E) = 0$$

$$x_l^T E x_r = 0$$

The R an t recovered from the resulting E matrix define the motion of the concerned camera between t and t-1 frames.

Stereo Disparity

Binocular (Stereo) disparity refers to the difference in image location of an object seen by the left and right cameras, resulting from the cameras' horizontal separation (parallax).

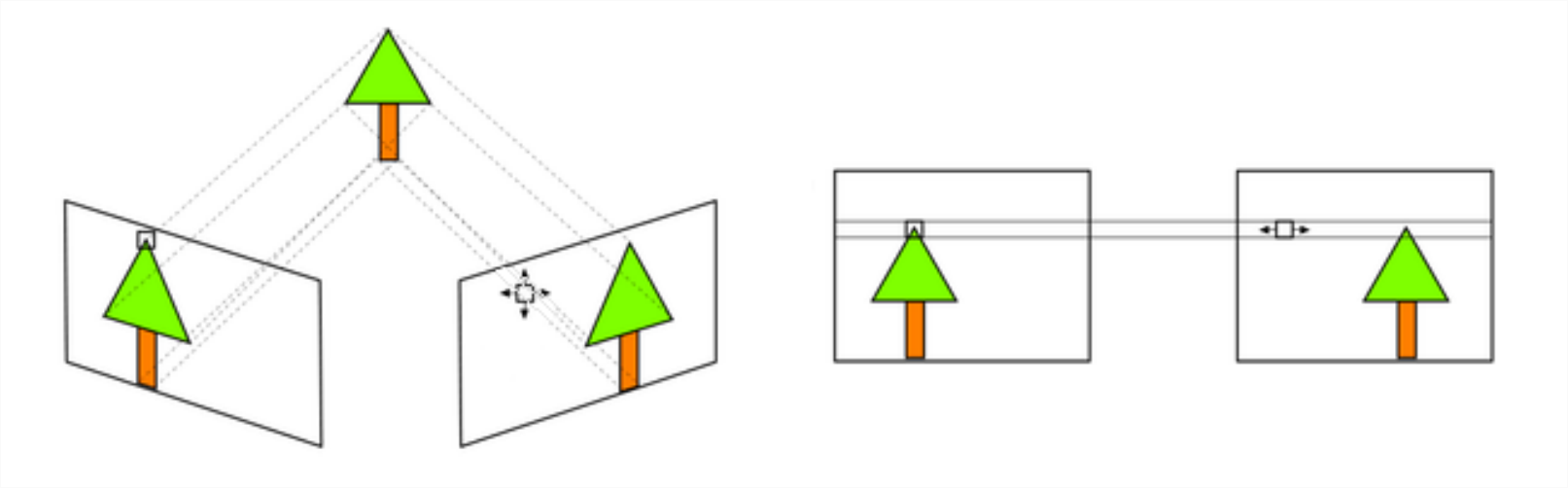


Figure: An example to showcase binocular disparity. Points closer to the cameras have higher disparity than the ones further away.