Drone Navigation based on Visual Inertial Odometry

SHUBHAM SINHA (21111409)

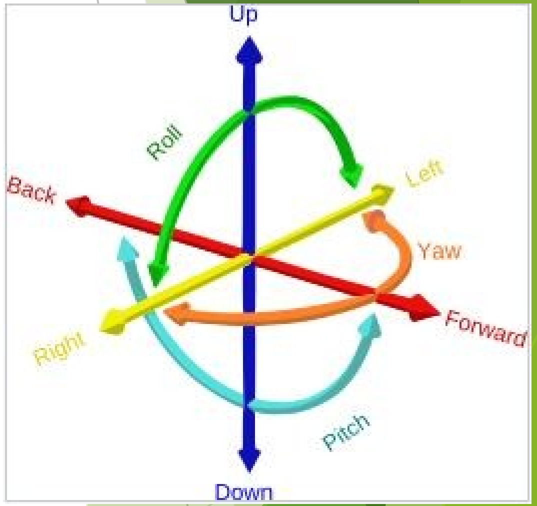
BINAY KUMAR SUNA (21111021)

Introduction

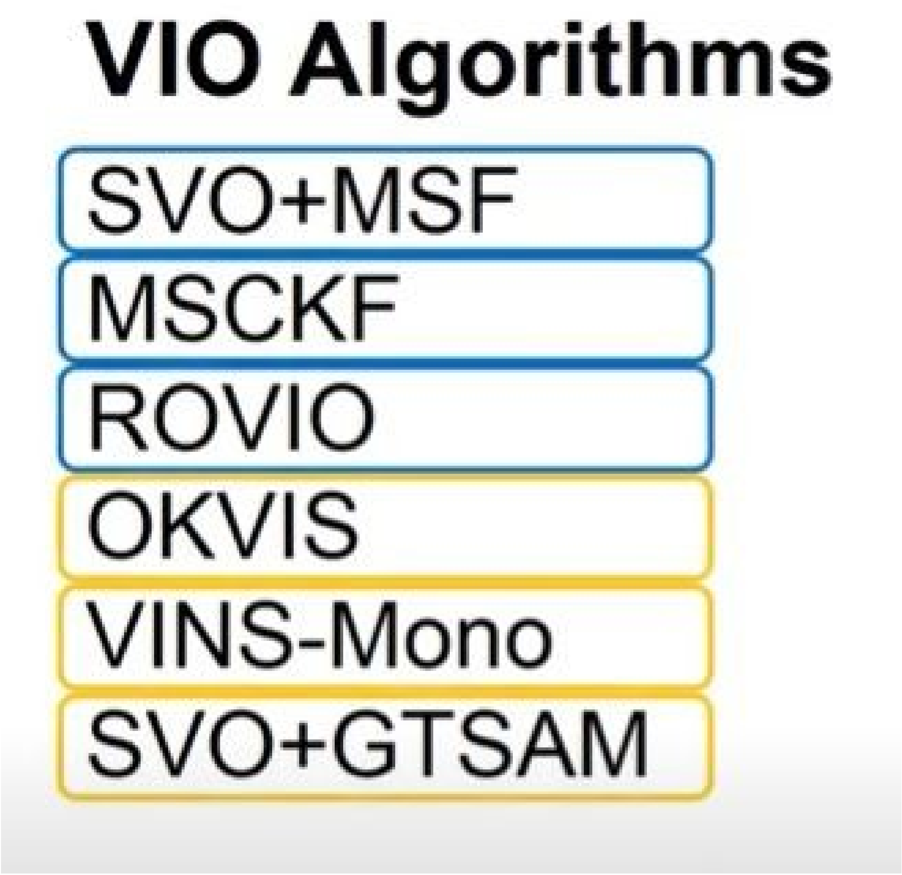
- Due to the increased rate of drone usage in various commercial and industrial fields, the need for their autonomous operation is rapidly increasing.
- Requires ability to operate safely in an unknown environment.
- GPS accuracy might be not suitable in some applications and this solution is not applicable to all situations.

Visual Odometry

- Odometry in Robotics is a more general term, and often refers to estimating not only the distance traveled, but the entire trajectory of a moving robot.
- In Visual Odometry we have a camera (or an array of cameras) rigidly attached to a moving object (such as a car or a robot), and our job is to construct a 6-DOF trajectory using the video stream coming from this camera(s).
- When we are using just one camera, it's called ***Monocular Visual Odometry***. When we're using two (or more) cameras, it's refered to as ***Stereo Visual Odometry***.
- The advantage of stereo is that you can estimate the exact trajectory, while in monocular you can only estimate the trajectory, unique only up to a scale factor.



Variations in VIO

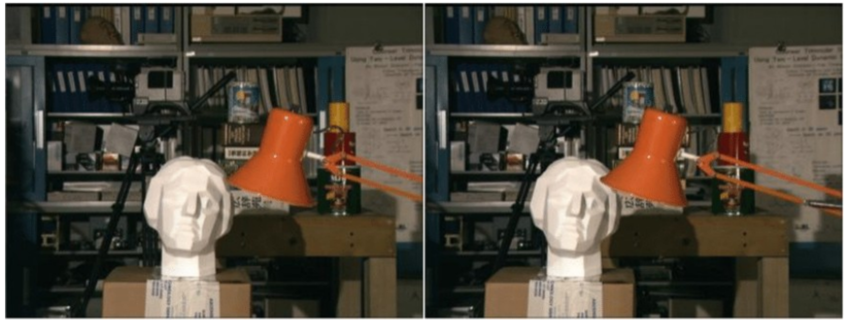


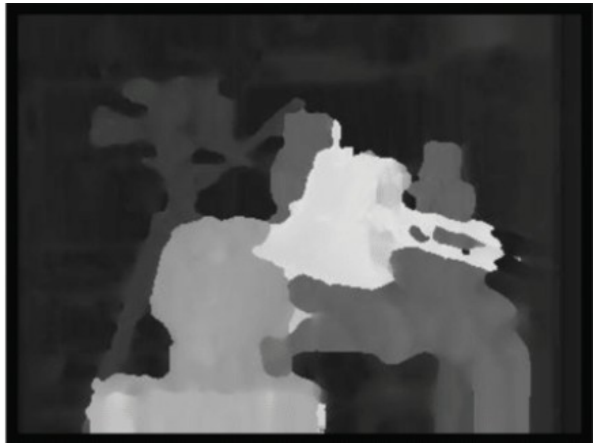
VIO Algorithms

- SVO+MSF
- MSCKF
- ROVIO
- OKVIS
- VINS-Mono
- SVO+GTSAM

Disparity Map Computation

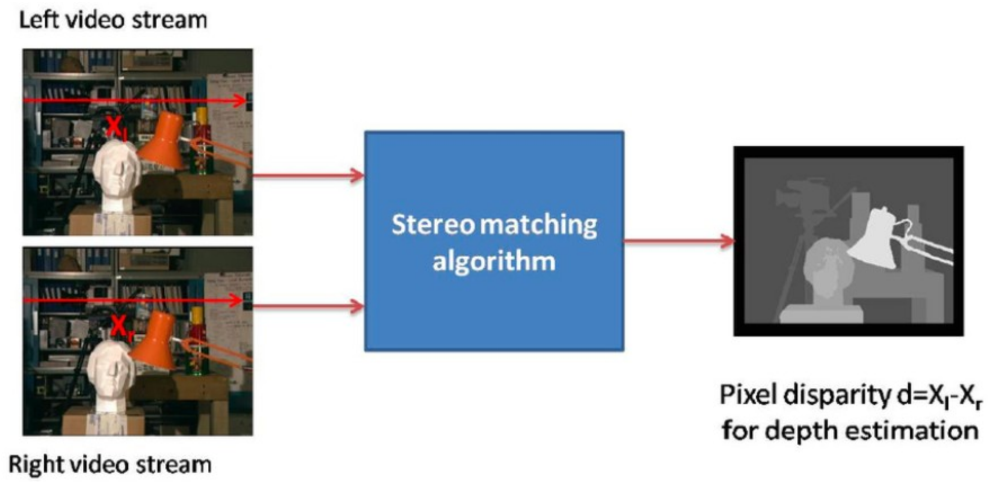
- Before computing the disparity maps, we must perform undistrortion and rectification.
- Suppose a particular 3D in the physical world F is located at the position (x,y) in the left image, and the same feature is located on (x + d,y) in the second image, then the location (x,y) on the disparity map holds the value d.





Block Matching Algorithm

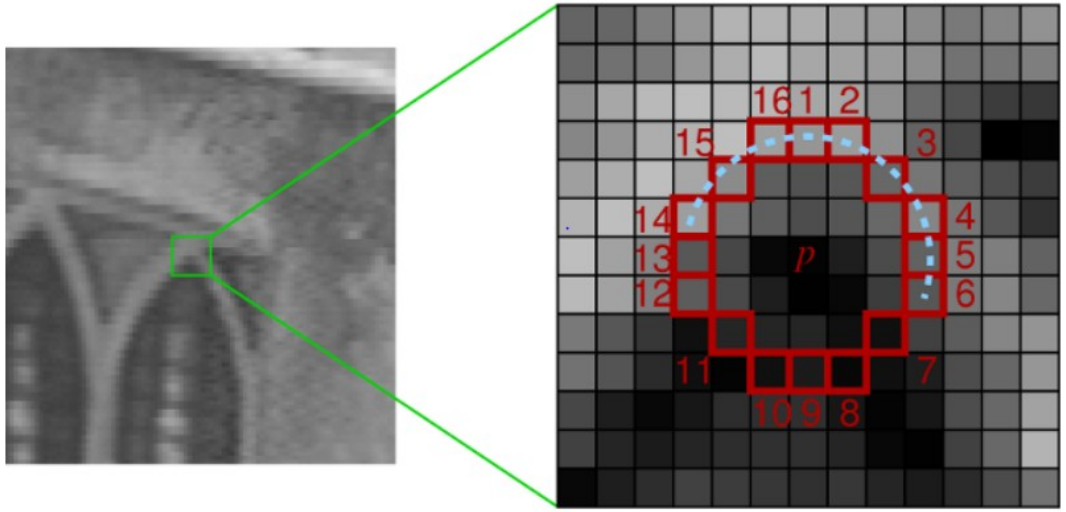
- Disparity at each point is computed using a sliding window. For every pixel in the left image a 15x15 pixels wide window is generated around it, and the value of all the pixels in the windows is stored.
- his window is then constructed at the same coordinate in the right image, and is slid horizontally, until the Sum-of-Absolute-Differences (SAD) is minimized.


Left video stream
X_l
X_r
Right video stream
Stereo matching algorithm
Pixel disparity $d=X_l-X_r$ for depth estimation

Stereo matching algorithm

Feature Detection

- Suppose there is a point P which we want to test if it is a corner or not. We draw a circle of 16px circumference around this point as shown in figure below.
- For every pixel which lies on the circumference of this circle, we see if there exits a continuous set of pixels whose intensity exceed the intensity of the original pixel by a certain factor I and for another set of contiguous pixels if the intensity is less by at least the same factor I. If yes, then we mark this point as a corner.

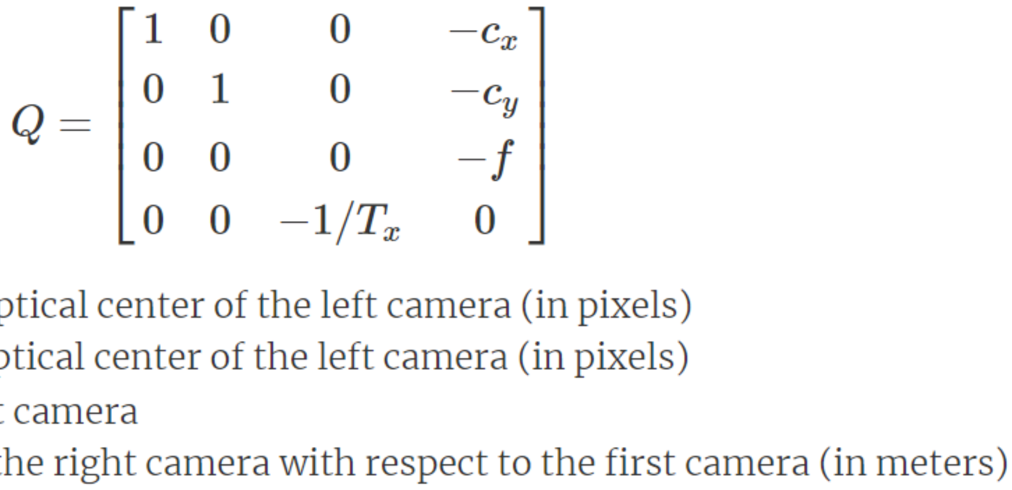


Feature Description and Matching

- The fast corners detected in the previous step are fed to the next step, which uses a KLT tracker (Kanade-Lucas-Tomasi).
- The KLT tracker basically looks around every corner to be tracked, and uses this local information to find the corner in the next image.
- The KLT tracks an object in two steps :
- It locates the trackable features in the initial frame, and then tracks each one of the detected features in the rest of the frames by means of its displacement
- The displacement of the specific feature is then defined as the displacement that minimizes the sum of differences. This is done continuously between sequential images so that all the features can be tracked.

Triangulation of 3D point cloud

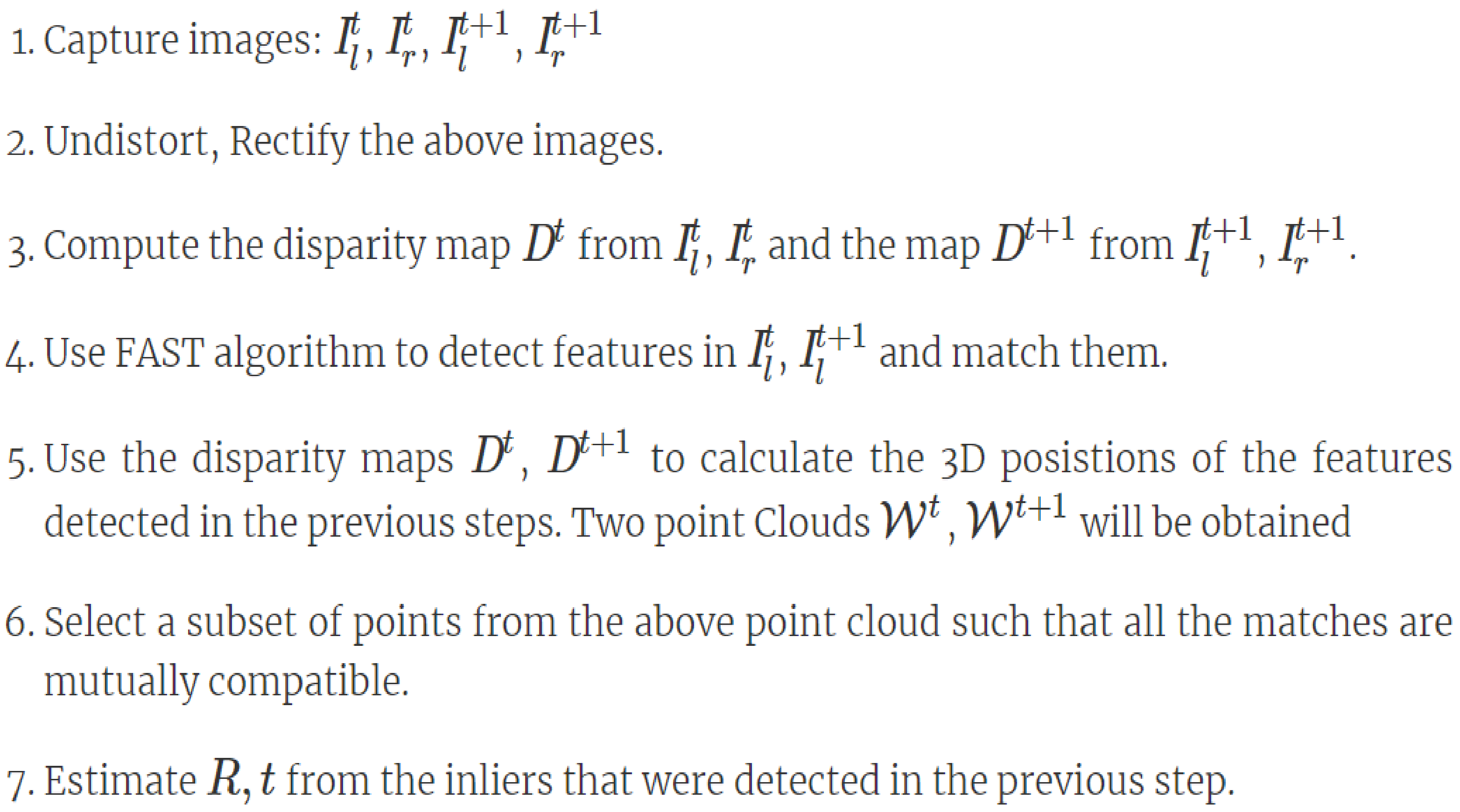
- The real world 3D coordinates of all the point in F(t) and F(t+1) are computed with respect to the left camera using the disparity value corresponding to these features from the disparity map, and the known projection matrices of the two cameras P1 and P2.
- We first form the reprojection matrix Q, using data from P1 and P2.



$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & -f \\ 0 & 0 & -1/T_x & 0 \end{bmatrix}$$

c_x = x-coordinate of the optical center of the left camera (in pixels)
c_y = y-coordinate of the optical center of the left camera (in pixels)
f = focal length of the first camera
T_x = The x-coordinate of the right camera with respect to the first camera (in meters)

The Algorithm



1. Capture images: I_l^t, I_r^t, I_l^{t+1}, I_r^{t+1}

2. Undistort, Rectify the above images.

3. Compute the disparity map D^t from I_l^t, I_r^t and the map D^{t+1} from I_l^{t+1}, I_r^{t+1}.

4. Use FAST algorithm to detect features in I_l^t, I_l^{t+1} and match them.

5. Use the disparity maps D^t, D^{t+1} to calculate the 3D posistions of the features detected in the previous steps. Two point Clouds $\mathcal{W}^t$, $\mathcal{W}^{t+1}$ will be obtained

6. Select a subset of points from the above point cloud such that all the matches are mutually compatible.

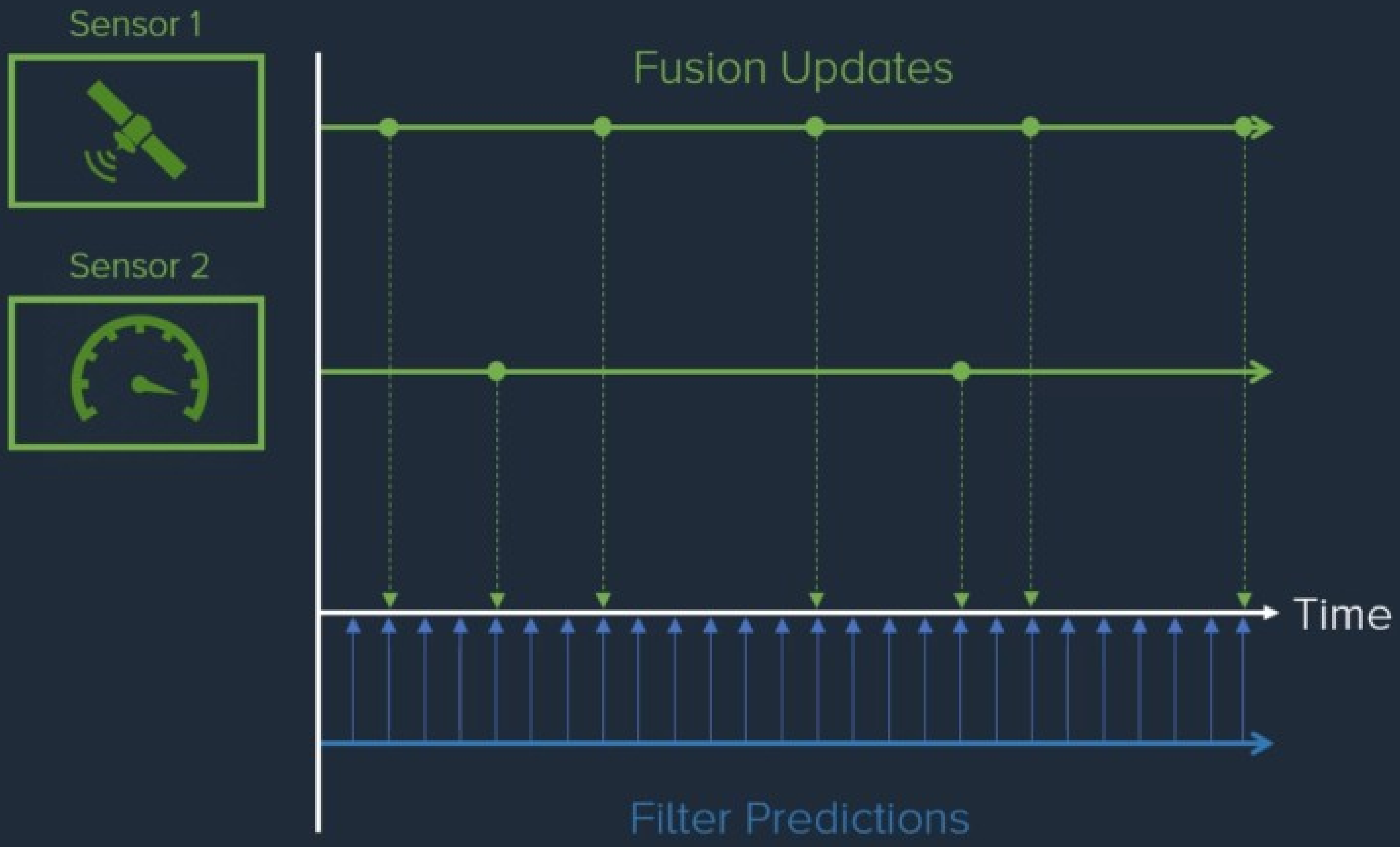
7. Estimate R, t from the inliers that were detected in the previous step.

Fusion of Visual Data and Inertial Measurements

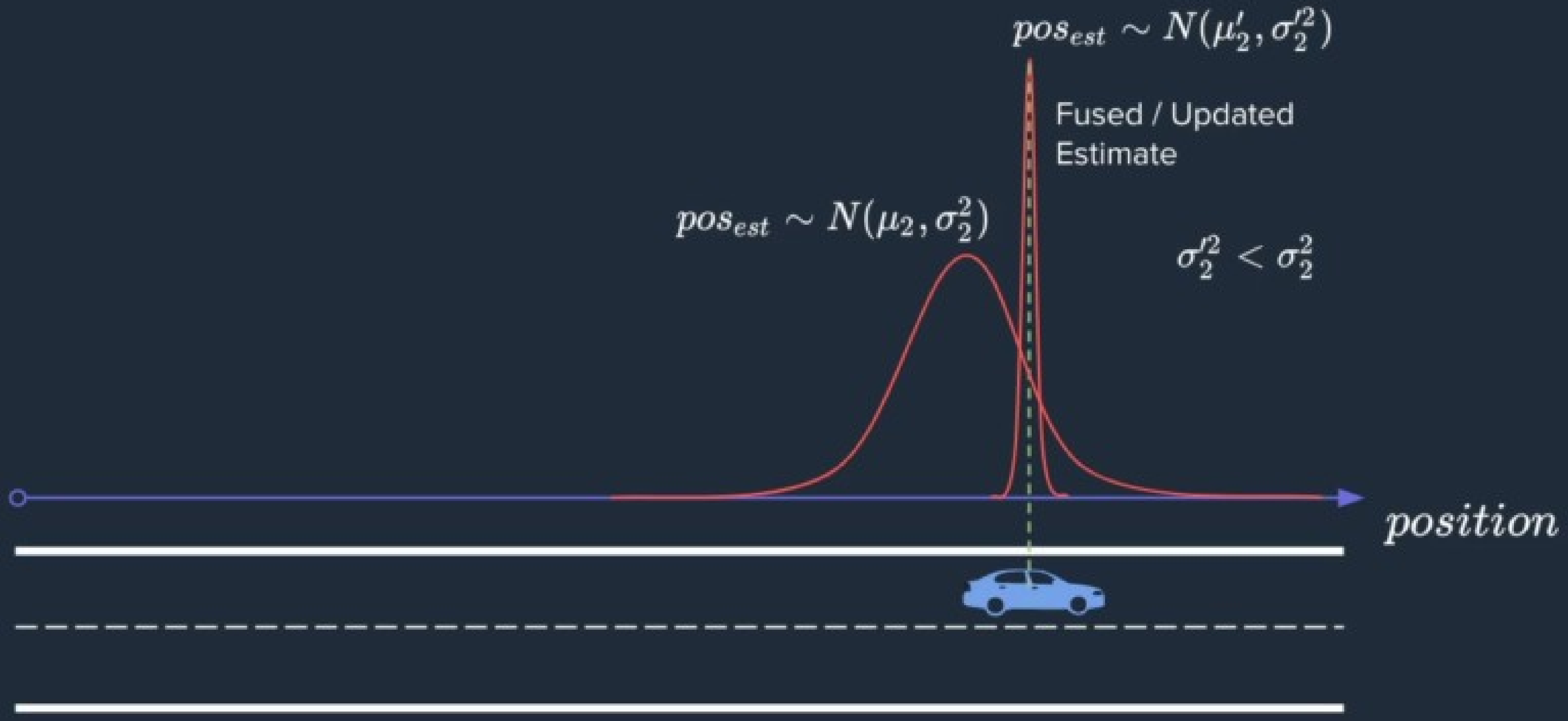
- The shortage of visual data is that error accumulates with distance while the shortage in inertial measurement is that error accumulates with time.
- Visual and inertial measurements fused can be classified into two ways, which are the loosely coupled approach and tightly coupled approach.
- The loosely coupled approach is the approach that separately estimates the image motions and inertial measurements, and then fuses these two estimates to obtain the final estimate.
- The tightly coupled approach is the approach that fuses the visual and inertial data directly at the measurement level to jointly estimate all IMU and camera states
- Generally, the tightly coupled approach is better in term of accuracy and robustness in motion estimation.

Fusion of Visual Data and Inertial Measurements

- There were also two different approaches for VIO approaches according to how visual and inertial measurements are fused, namely the filtering-based approach and optimization-based approach.
- EKF usually comprises an estimation step and an updating step. In the estimation step, the inertial sensors provide the acceleration and the measurements of three axes rotational velocity. These measurements act as the data-driven dynamic model for a 3D rigid motion and make the motion estimation. In the updating step, the visual measurement models (cameras) update the estimation results, by providing the ranging and angular measurements among the mobile platform and features.
- The optimization-based approach mainly depends on the feature extraction and image alignment optimization, where both are image processing techniques


Sensor 1
Sensor 2
Fusion Updates
Time
Filter Predictions

Estimation Problem



$pos_{est} \sim N(\mu_2', \sigma_2'^2)$
Fused / Updated Estimate
$pos_{est} \sim N(\mu_2, \sigma_2^2)$
$\sigma_2'^2 < \sigma_2^2$
$position$

Challenges

- Camera Calliberation.
- Initial State Estimation.
- Dataset Creation

Augmented Reality

- The combination of visual information and inertial measurements has been greatly used as motion tracking technique in an Augmented Reality environment which can be used for obtaining an accurate measurement.

- Both the ARKit platform and ARCore platform use this technique to enable accurate motion tracking in real time.