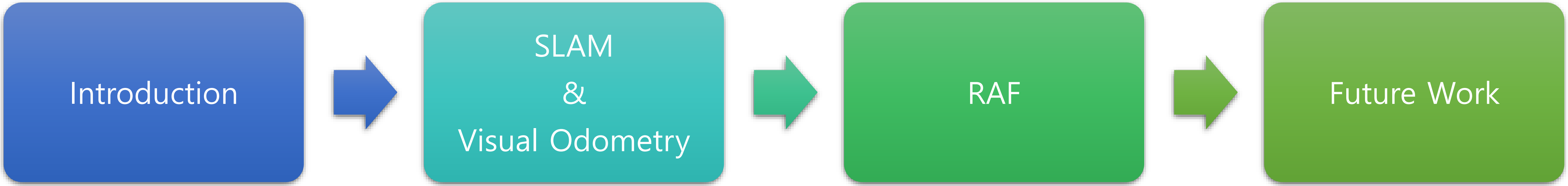
Robust Aged Feature

Jeon Hyun Ho

Contents



Introduction

SLAM
&
Visual Odometry

RAF

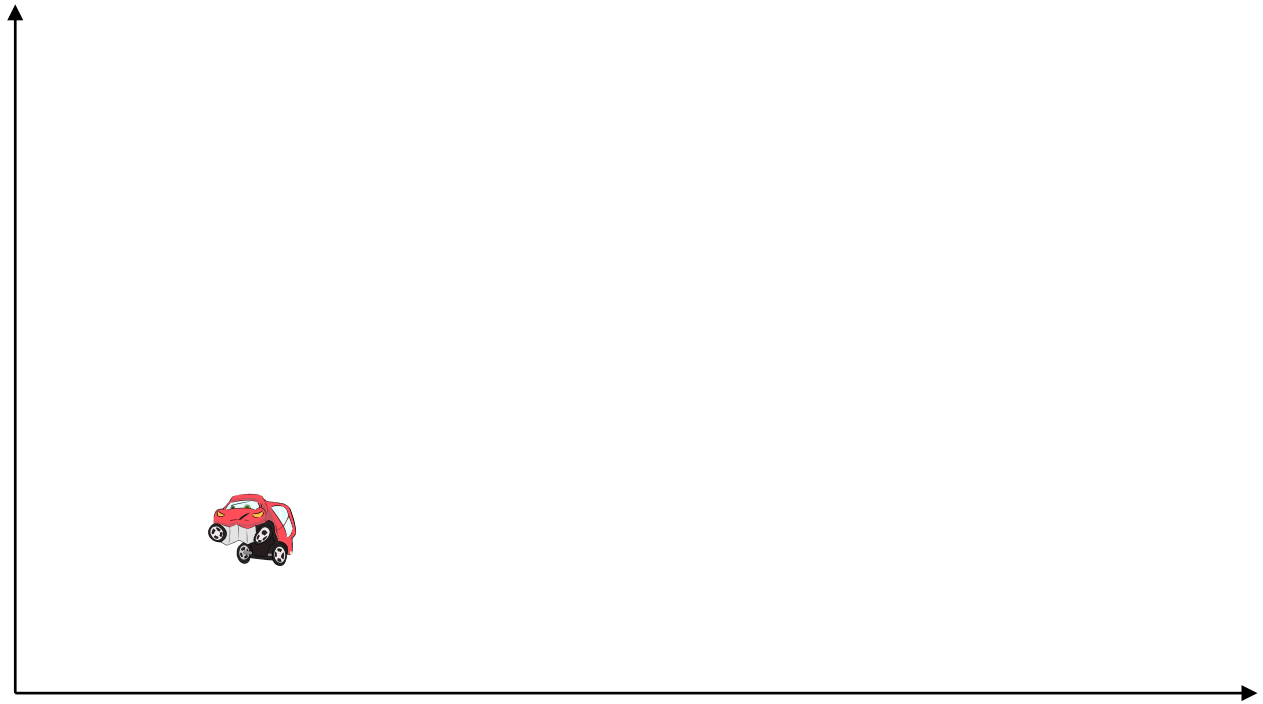
Future Work

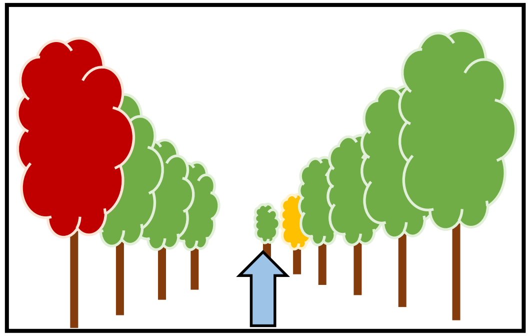
Introduction

- Motivation

Introduction

- Concept

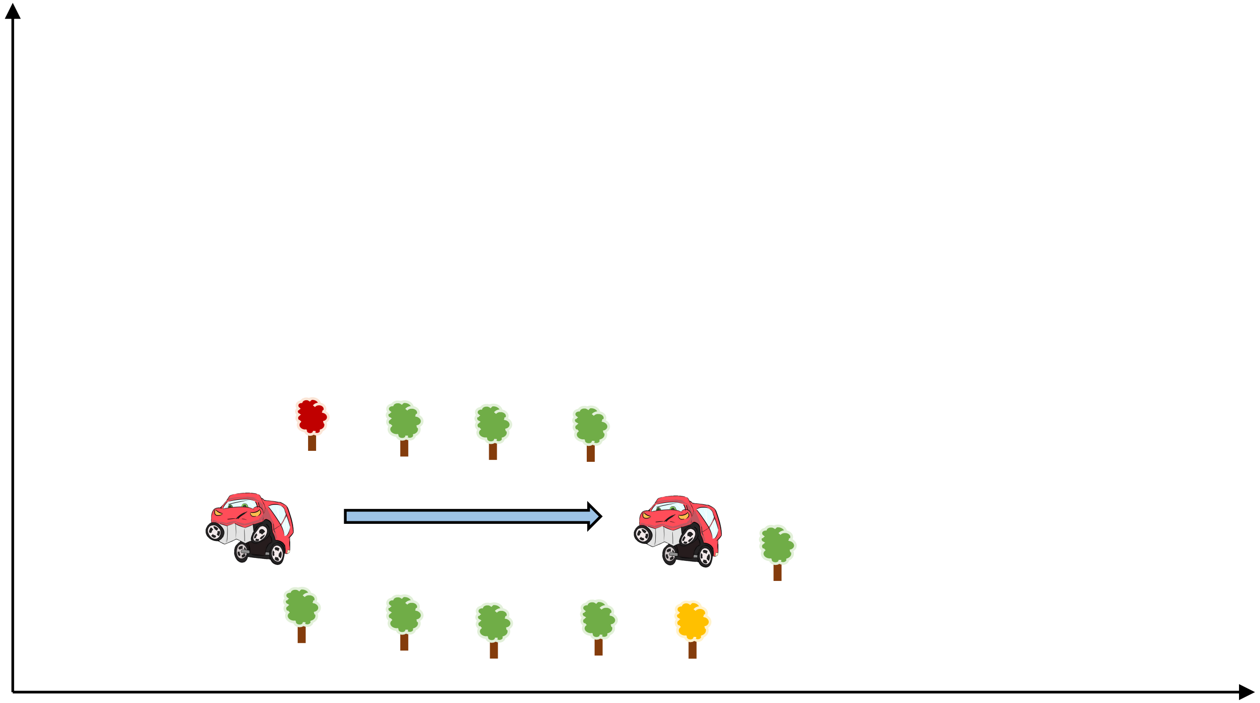


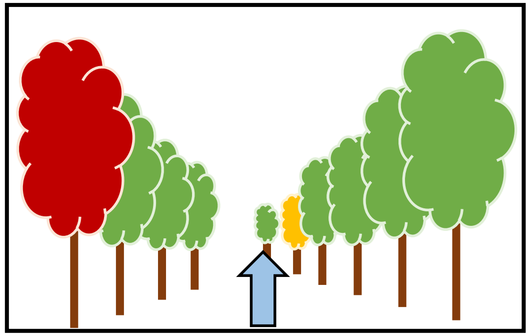


View

Introduction

- Concept

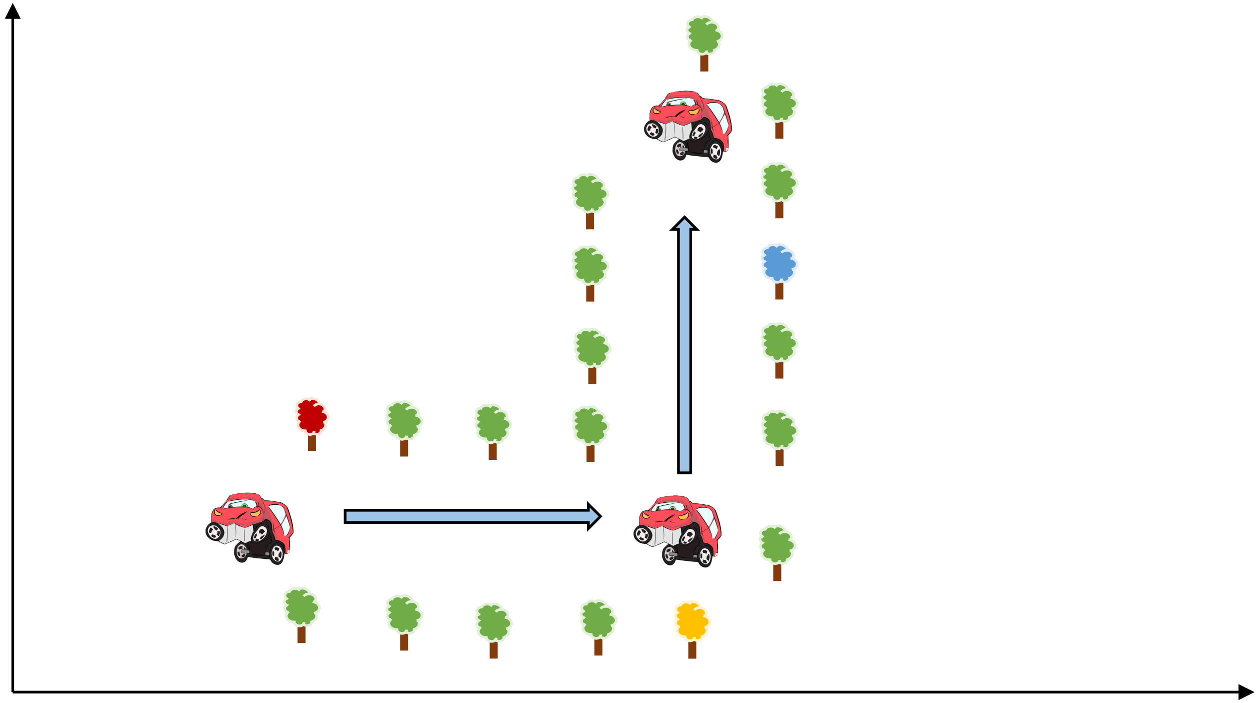


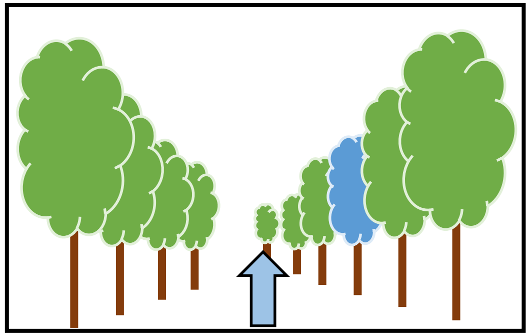


View

Introduction

- Concept

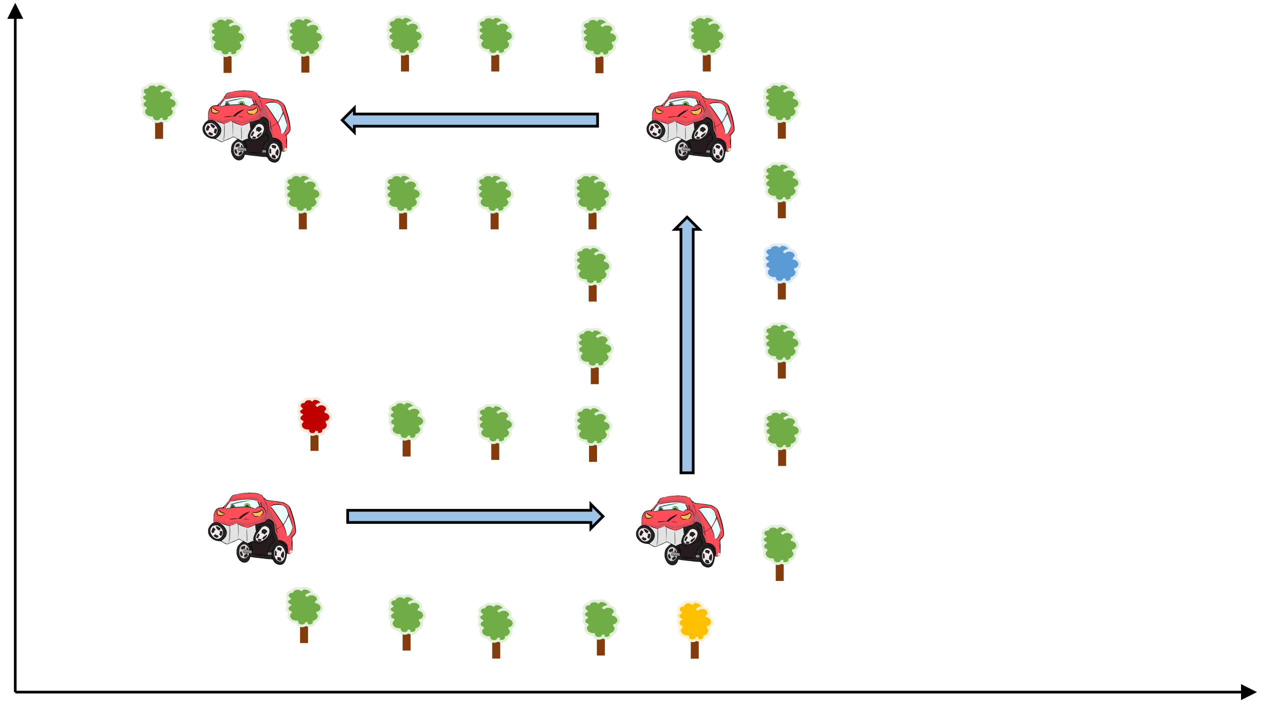


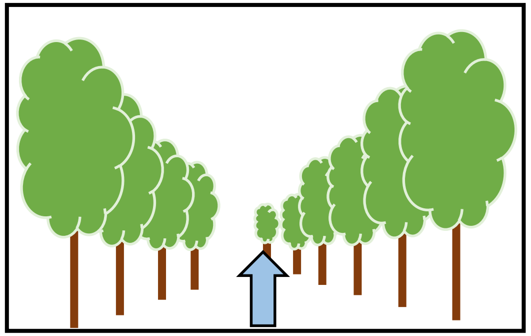


View

Introduction

- Concept

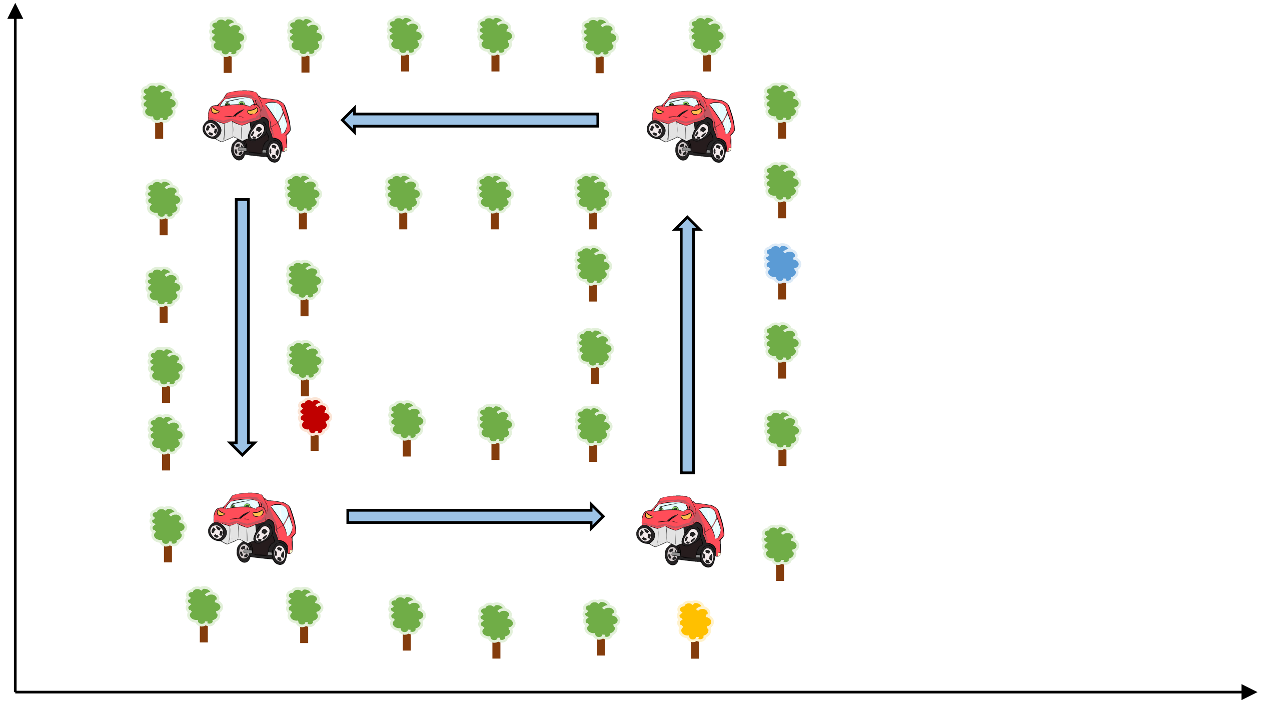


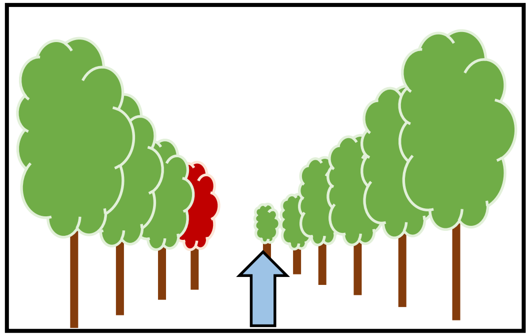


View

Introduction

- Concept





View

SLAM

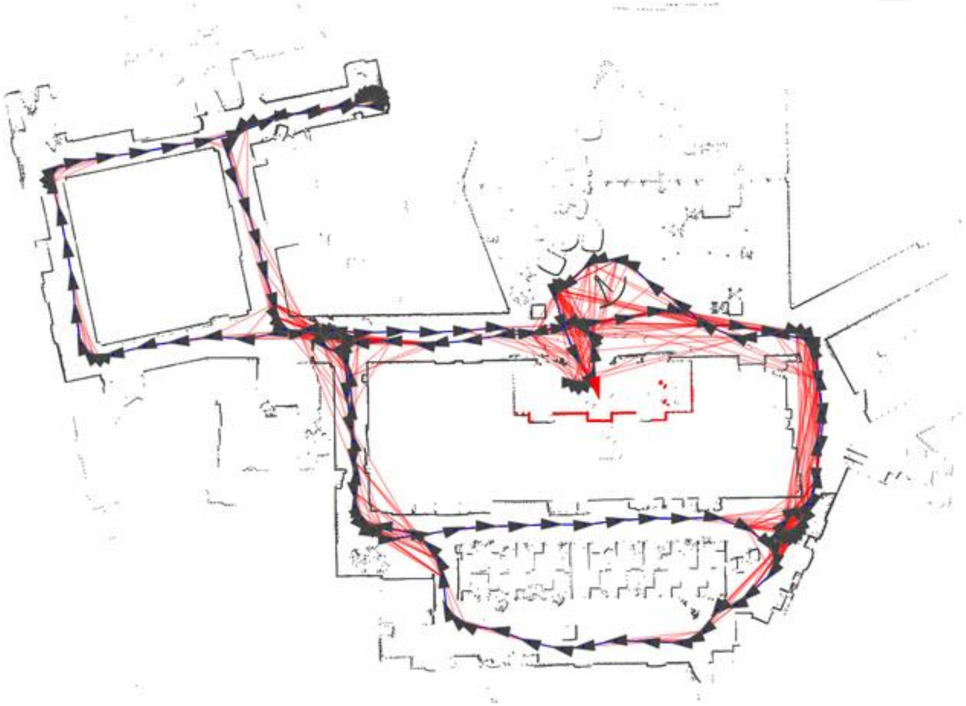
- What is SLAM?

  (Simultaneous Localization And Mapping)

  A robot is exploring an unknown environment.

  - Given
    - Robot motions
    - Observations of nearby features

  - Estimate
    - Map
    - Pose (Position + Orientation)
    - (Path Planning)

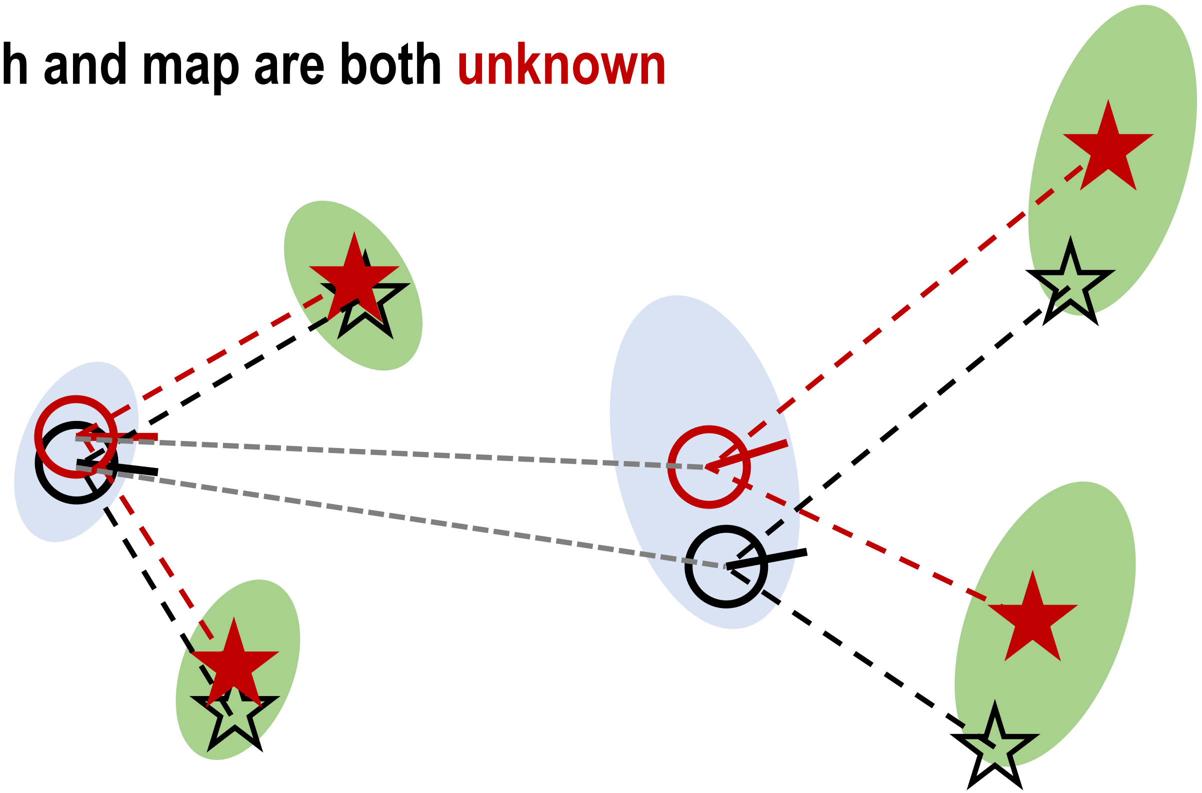


Graph SLAM

SLAM

- **Why is SLAM a hard problem?**

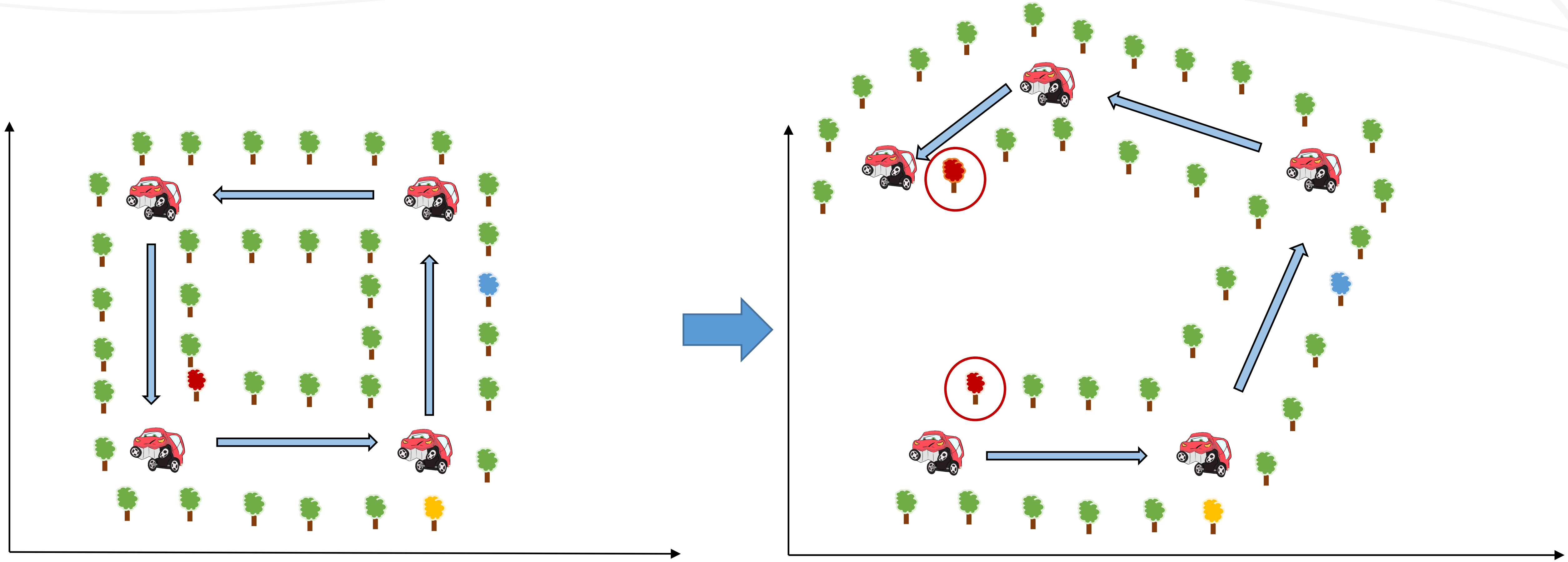
Robot path and map are both unknown



	Robot	Feature
True	⊖	☆
Estimate	⊖	★

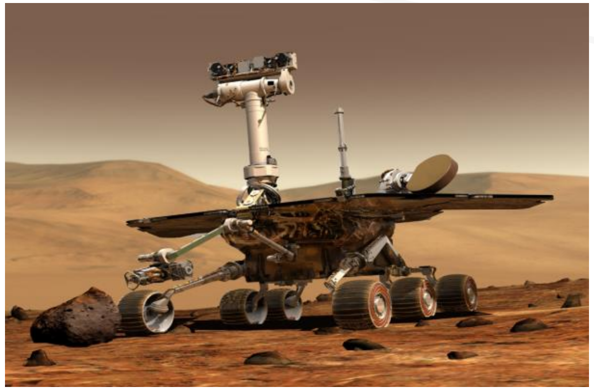
SLAM

- Why is SLAM a hard problem?



SLAM

- Applications
  - Indoor
  - Space
  - Self-driving car
  - Underground
  - Undersea



Visual Odometry

- What is Visual Odometry?

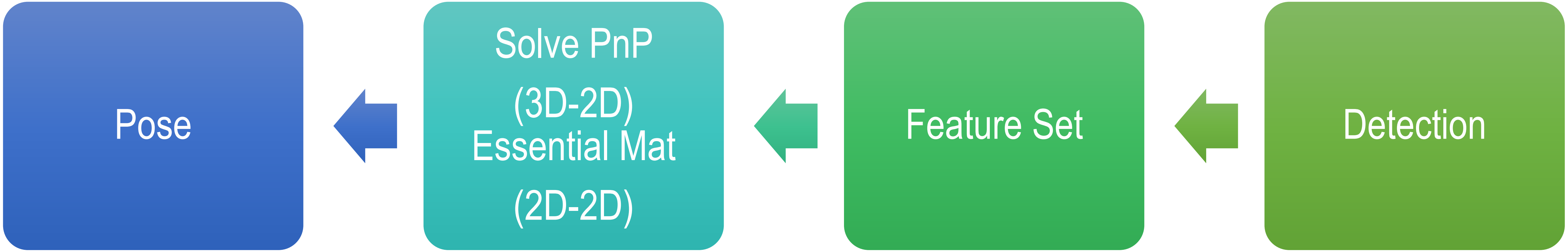
A robot is exploring an unknown environment.

- Given
  - ➢ Robot motions
  - ➢ Observations of nearby features (Image, LIDAR)

- Estimate
  - ➢ Map
  - ➢ Pose (Position + Orientation) → ∑ : Odometry
  - ➢ Path Planning

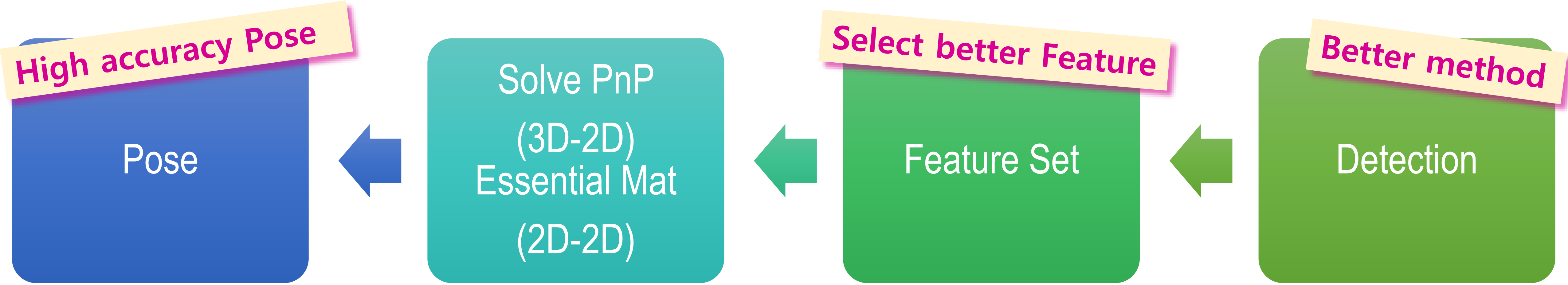
Visual Odometry

- Estimate Pose



Visual Odometry

- **Estimate Better Pose**



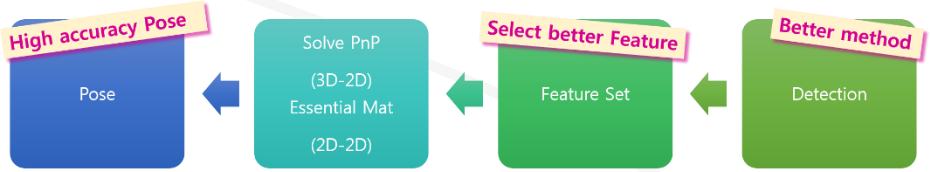
Visual Odometry

- Estimate Better Pose

High accuracy Pose
Solve PnP
(3D-2D)
Essential Mat
(2D-2D)
Select better Feature
Better method

Robust Aged Feature

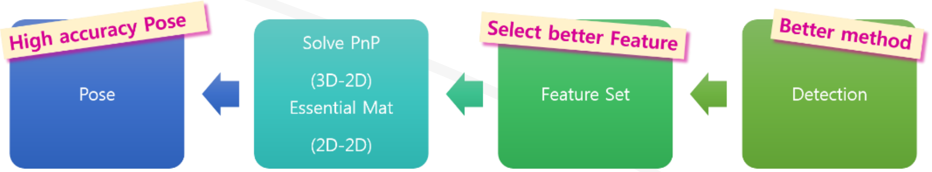
RAF



- RAF Framework Change
  - Detection

Color image

RAF

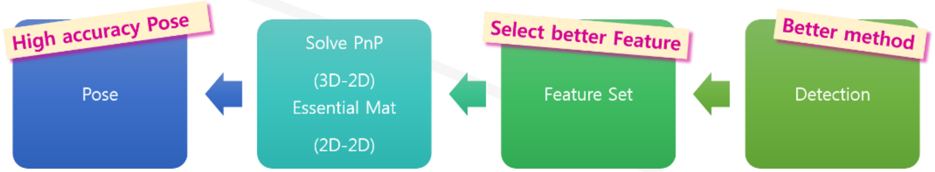


- **RAF Framework Change**

  - **Detection**

Gray scale image

RAF



High accuracy Pose

Pose

Solve PnP

(3D-2D)

Essential Mat

(2D-2D)

Select better Feature

Feature Set

Better method

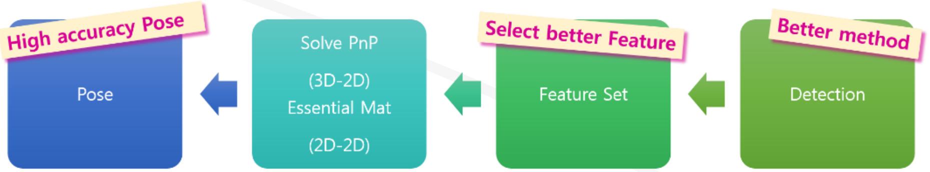
Detection

- **RAF Framework Change**

  - **Detection**

Gradient image

RAF



High accuracy Pose

Pose

Solve PnP

(3D-2D)

Essential Mat

(2D-2D)

Select better Feature

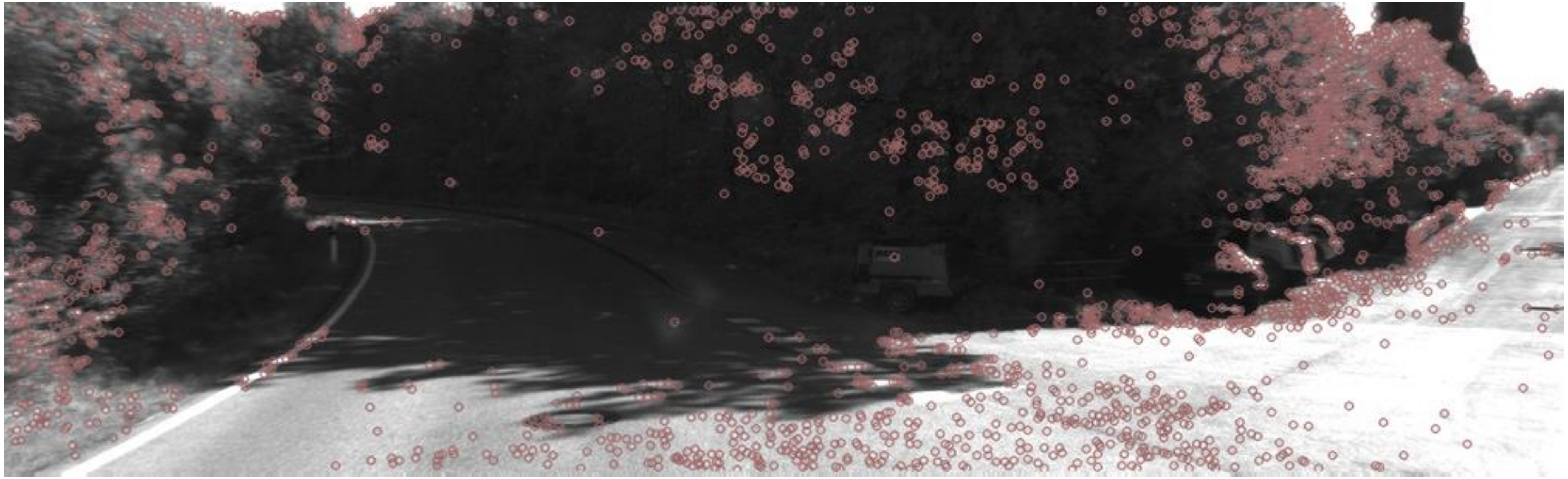
Feature Set

Better method

Detection

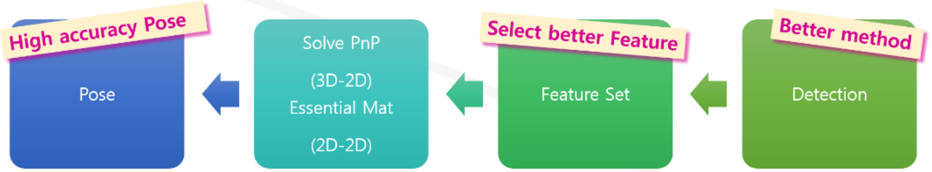
- **RAF Framework Change**

  - **Detection**



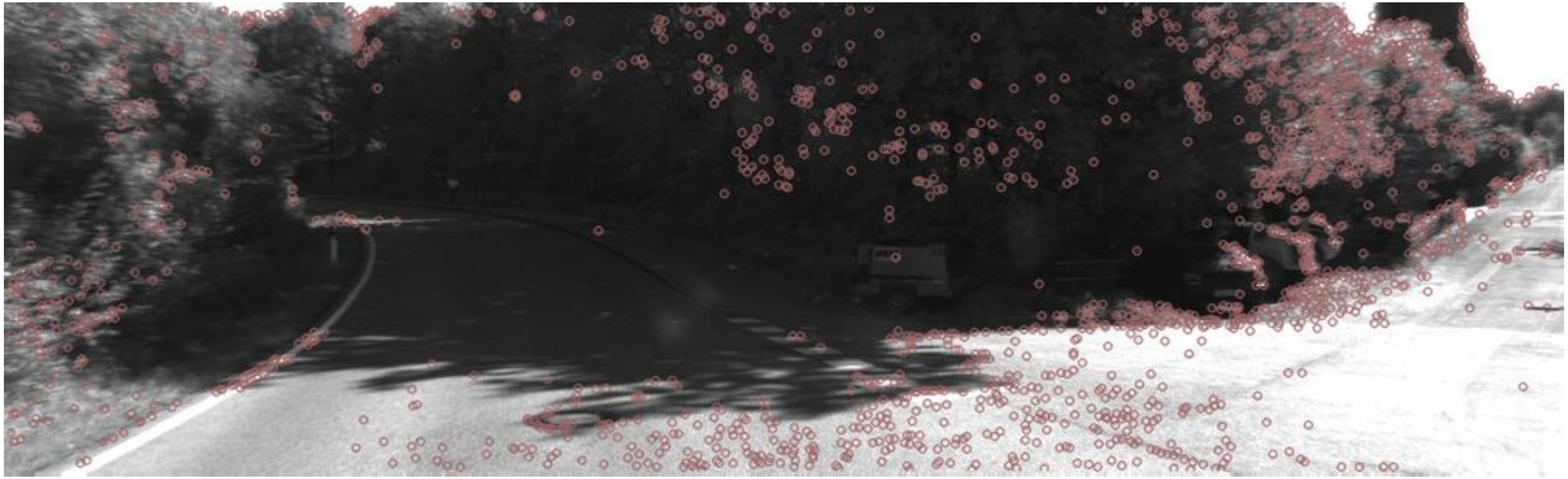
Detection : FAST (Gray)

RAF



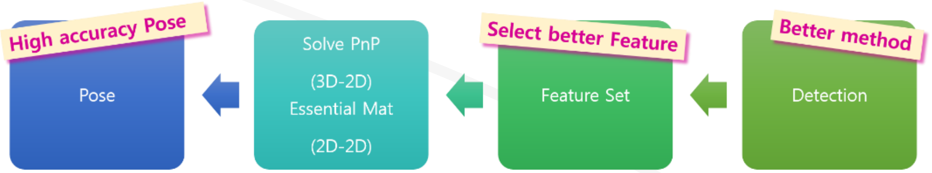
- **RAF Framework Change**

  - **Detection**



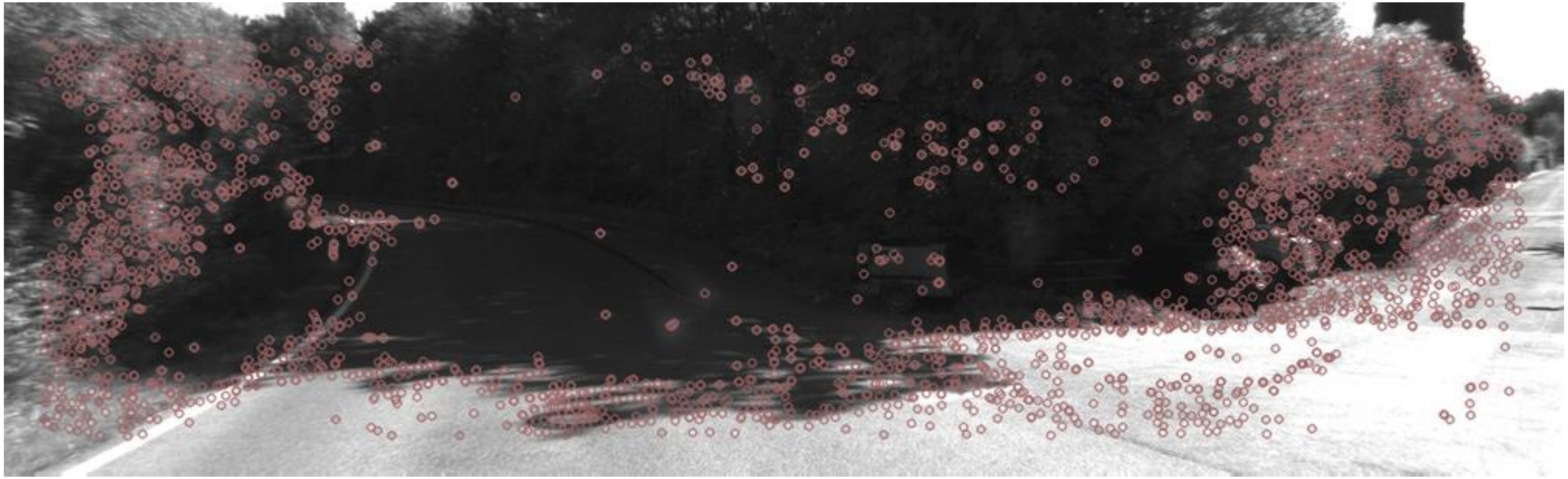
Detection : FAST (Gradient)

RAF

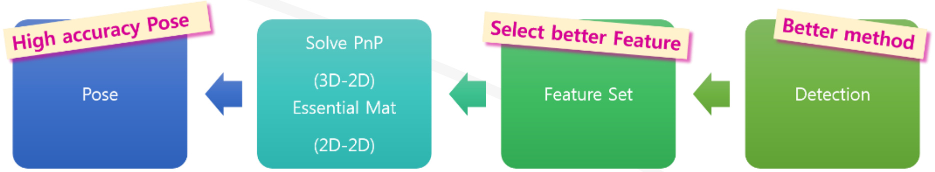


- **RAF Framework Change**

  - **Detection**



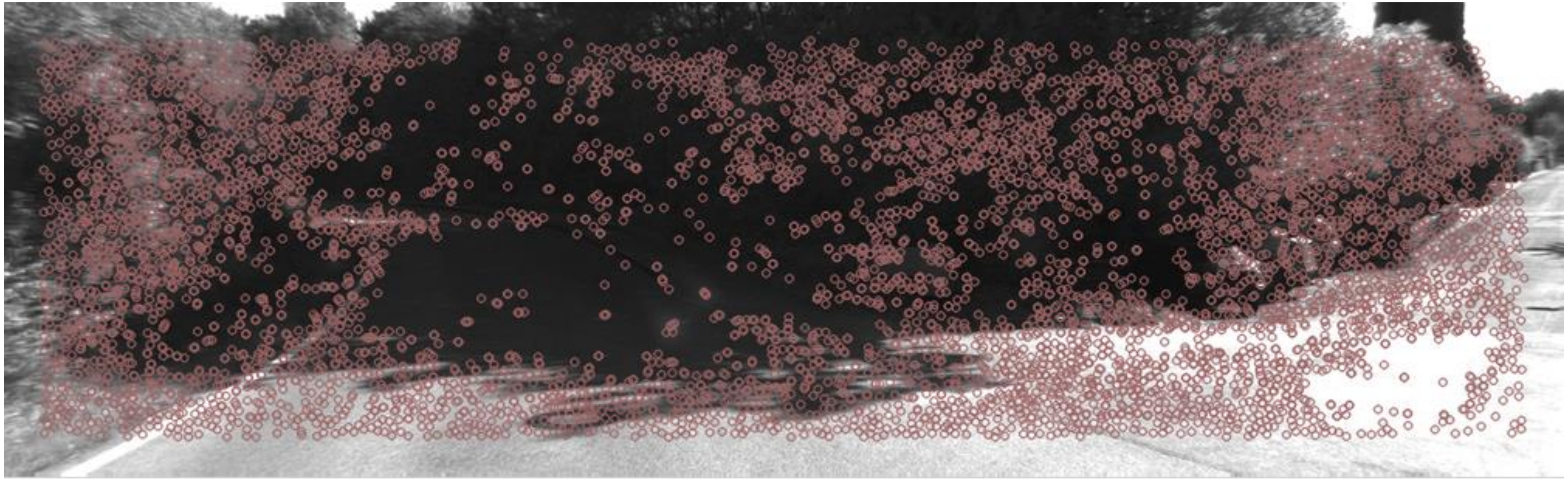
Detection : AKAZE - 1



RAF

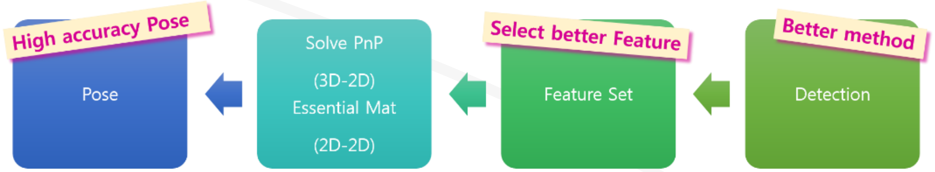
- **RAF Framework Change**

  - **Detection**



Detection : AKAZE - 2

RAF



High accuracy Pose

Pose

Solve PnP

(3D-2D)

Essential Mat

(2D-2D)

Select better Feature

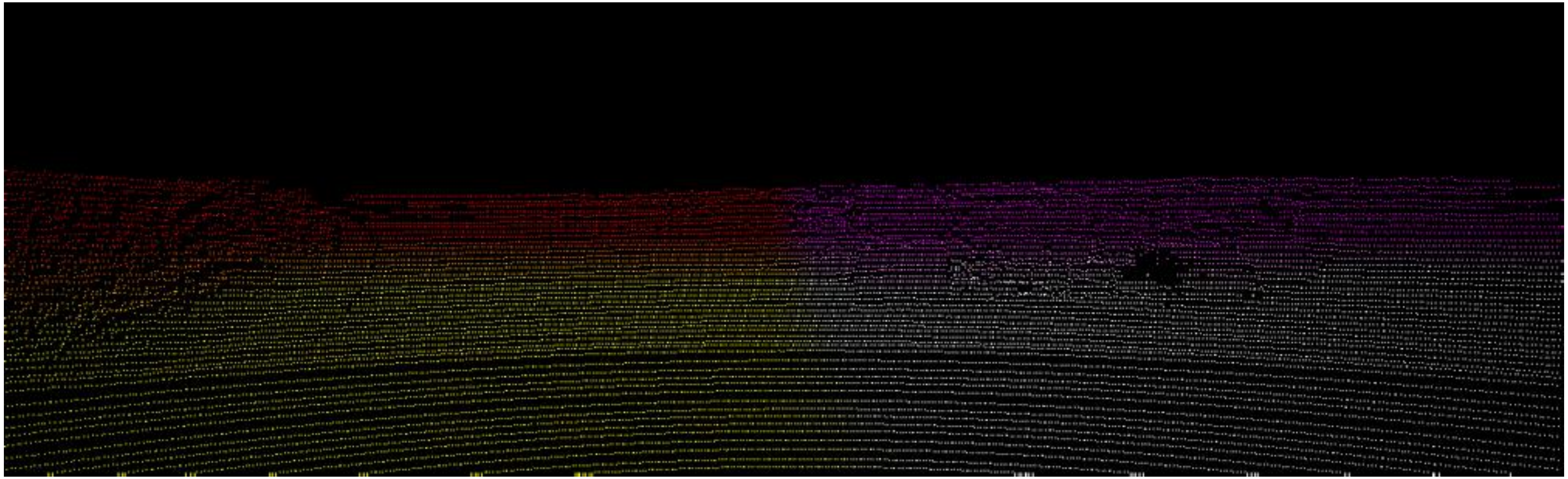
Feature Set

Better method

Detection

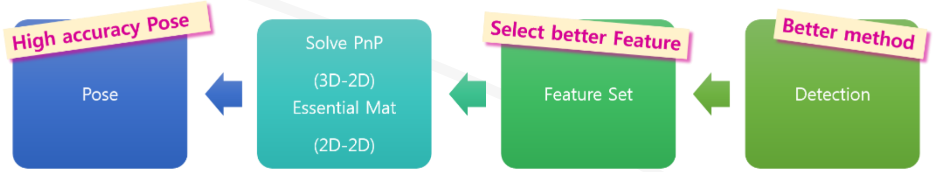
- **RAF Framework Change**

  - **Interpolation**



Interpolation : X

RAF


High accuracy Pose

Pose

Solve PnP

(3D-2D)

Essential Mat

(2D-2D)

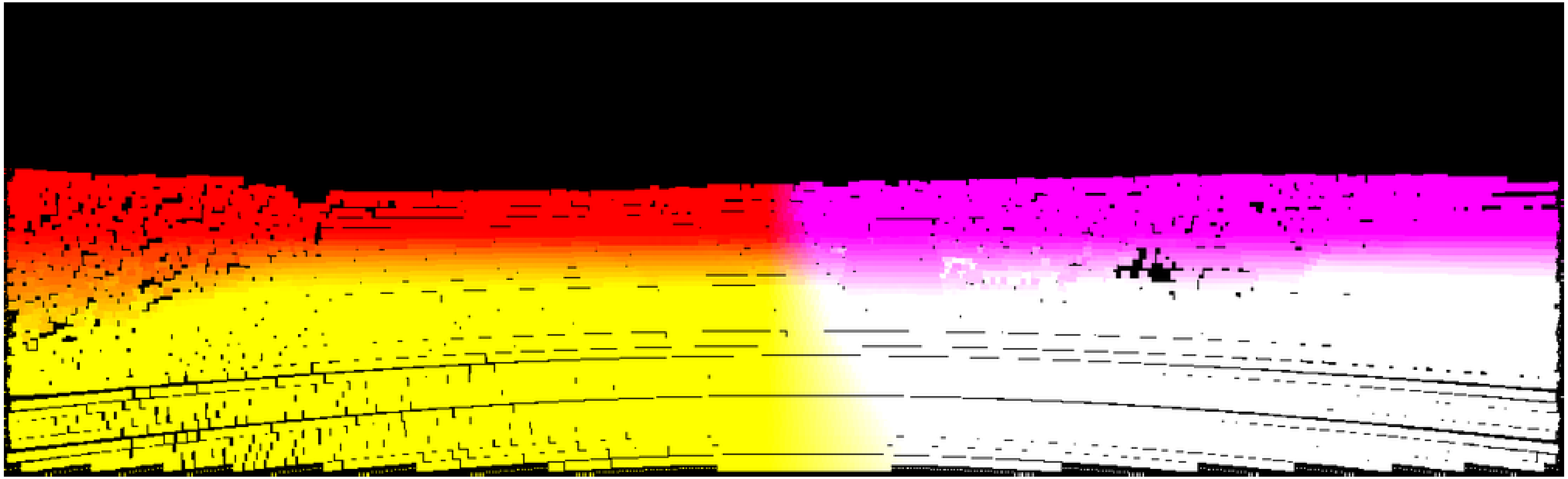
Select better Feature

Feature Set

Better method

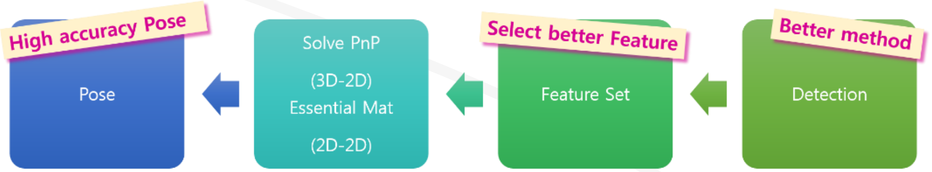
Detection

- **RAF Framework Change**
  - **Interpolation**

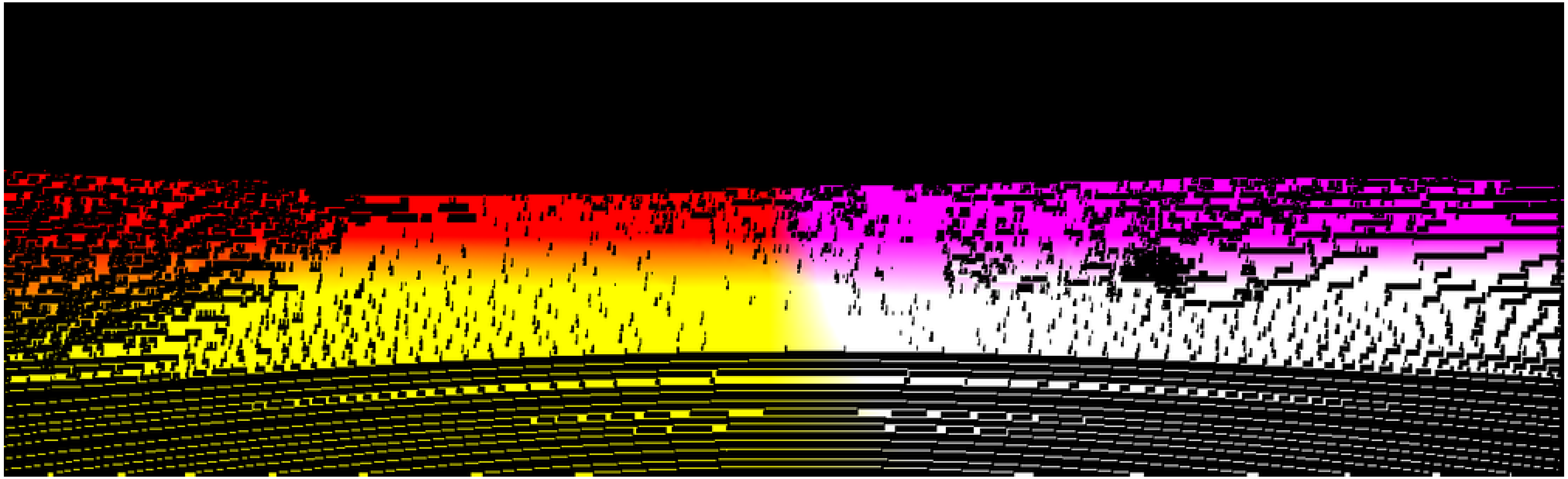


Interpolation : N x N mask

RAF

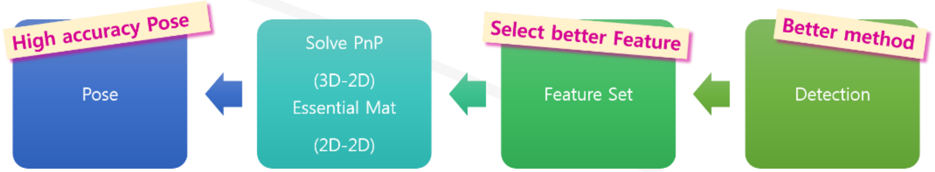


- RAF Framework Change
  - Interpolation



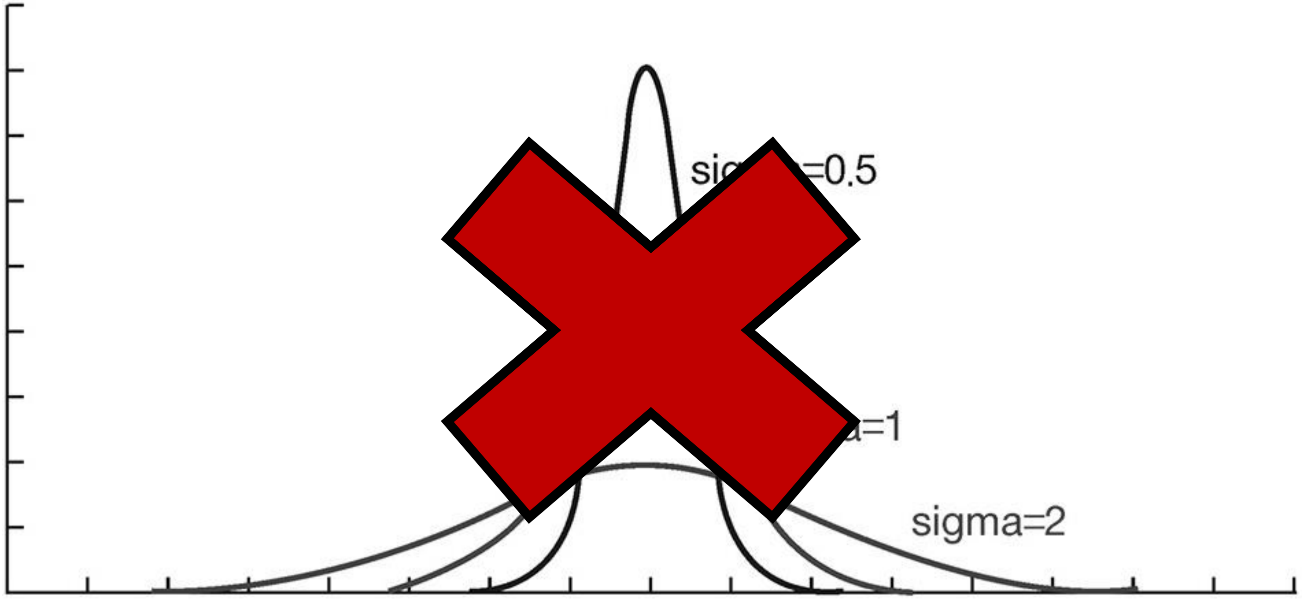
Interpolation : Bilinear

RAF

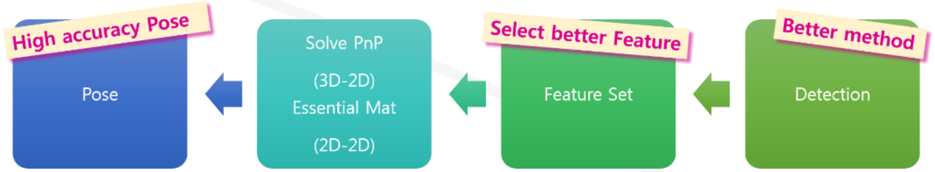


- **RAF Framework Change**

  - **Filtering**



Outlier Filtering



High accuracy Pose

Pose

Solve PnP

(3D-2D)

Essential Mat

(2D-2D)

Select better Feature

Feature Set

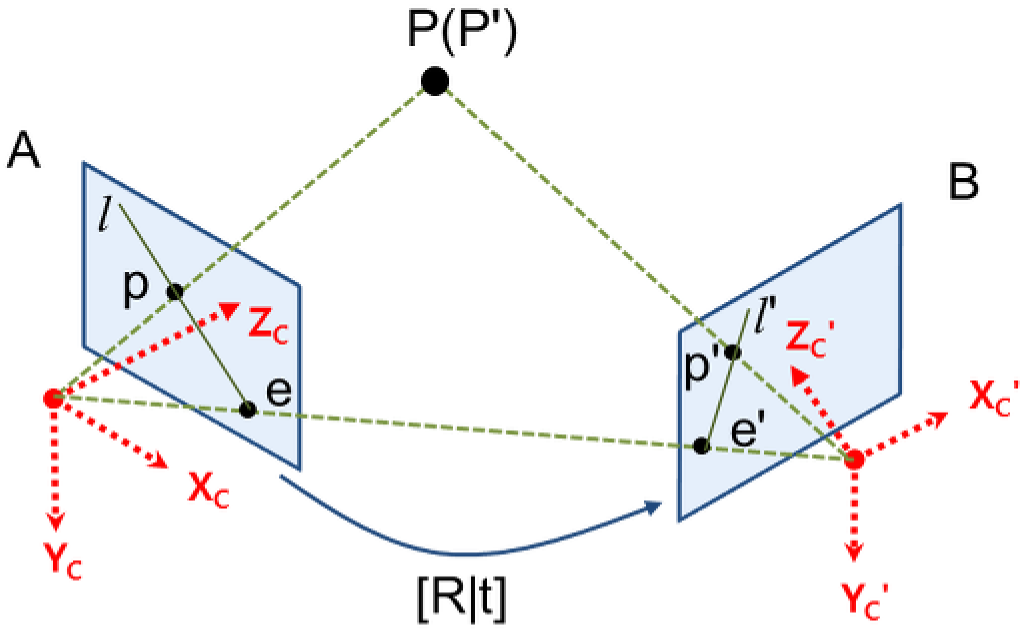
Better method

Detection

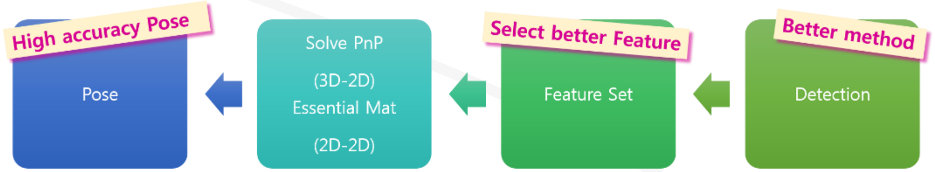
RAF

- **RAF Framework Change**

  - **Filtering**



Outlier Filtering



High accuracy Pose

Pose

Solve PnP

(3D-2D)
Essential Mat

(2D-2D)

Select better Feature

Feature Set

Better method

Detection

RAF

- **RAF Framework Change**

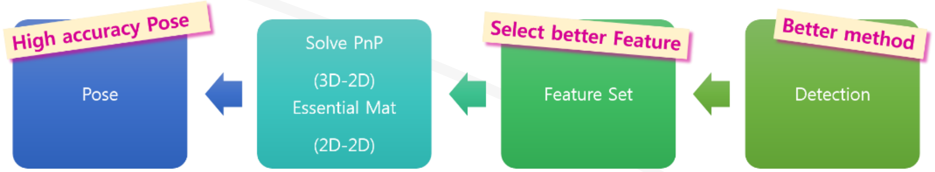
  - **Filtering**

avrgDistance

?

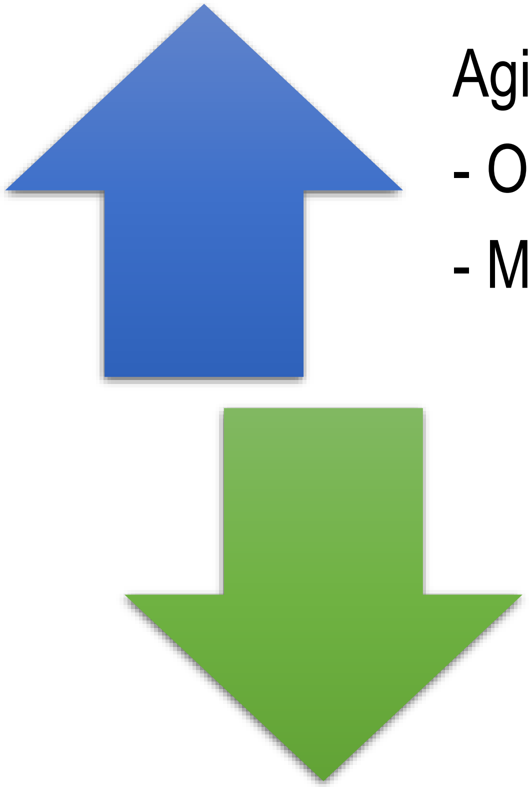
Stop motion Filtering

RAF



- **RAF Framework Change**

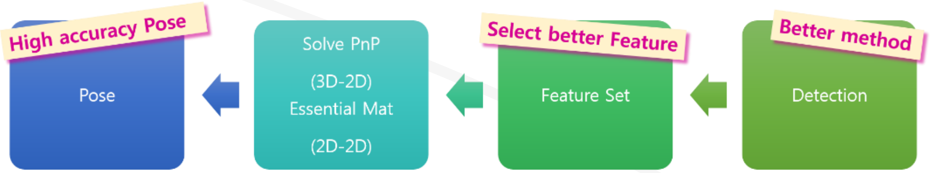
  - **Aging**
    - ➢ **First Detection**
      - **Initial Age**
    - ➢ **Delete**
      - **Age < Threshold**



Aging
- Optical Flow
- Match

De-Aging
- Fail (OF + Match)

RAF



- RAF Framework Change
  - Aging
    - ➢ First Detection
      - Initial Age
    - ➢ Delete
      - Age < Threshold

Aging, (3D)

- Detection (Aging)

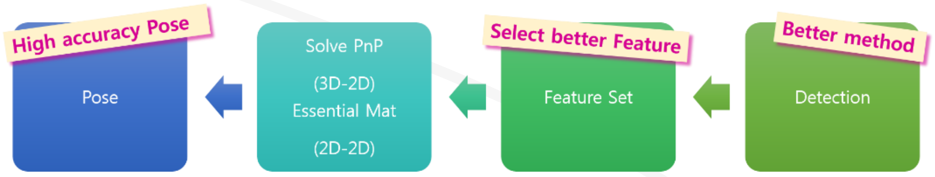
- Optical Flow, (Double)

- Match, (Double)

De-Aging

- Fail (OF + Match)

RAF



High accuracy Pose

Pose

Solve PnP

(3D-2D)

Essential Mat

(2D-2D)

Select better Feature

Feature Set

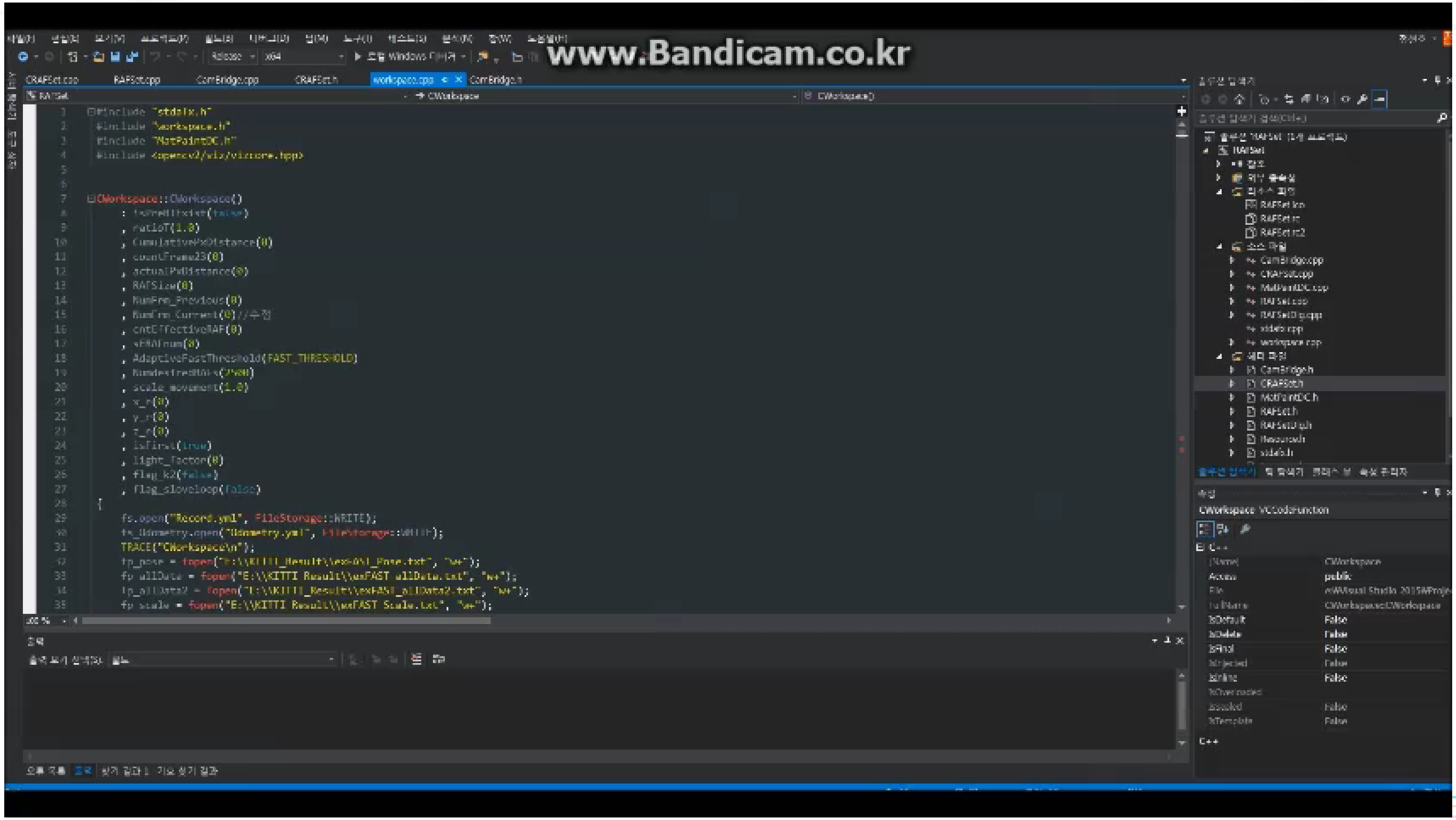
Better method

Detection

- RAF Framework Change
  - Localization & Mapping
    - ➢ Frist (0, 0, 0) ~ Current
      - Auto Loop closing
      - Error + Error + Error + …
    - ➢ Previous ~ Current
      - Small error

Implementation

- **result**



```cpp
#include "stdafx.h"
#include "workspace.h"
#include "MatPointDC.h"
#include <opencv2/viz/vizcore.hpp>


CWorkspace::CWorkspace()
        : isPreHitExist(false)
        , ratioT(1.0)
        , CumulativePxDistance(0)
        , countFrame23(0)
        , actualPxDistance(0)
        , RAFSize(0)
        , NumFrm_Previous(0)
        , NumFrm_Current(0)//수정
        , cntEffectiveRAF(0)
        , sERAFnum(0)
        , AdaptiveFastThreshold(FAST_THRESHOLD)
        , NumdesiredRois(2500)
        , scale_movement(1.0)
        , x_r(0)
        , y_r(0)
        , z_r(0)
        , isFirst(true)
        , light_factor(0)
        , flag_k2(false)
        , flag_sloveloop(false)
{
        fs.open("Record.yml", FileStorage::WRITE);
        fs_Odometry.open("Odometry.yml", FileStorage::WRITE);
        TRACE("CWorkspace\n");
        fp_pose = fopen("E:\\KITTI_Result\\exFAST_Pose.txt", "w+");
        fp_allData = fopen("E:\\KITTI_Result\\exFAST_allData.txt", "w+");
        fp_allData2 = fopen("E:\\KITTI_Result\\exFAST_allData2.txt", "w+");
        fp_scale = fopen("E:\\KITTI_Result\\exFAST_Scale.txt", "w+");
```

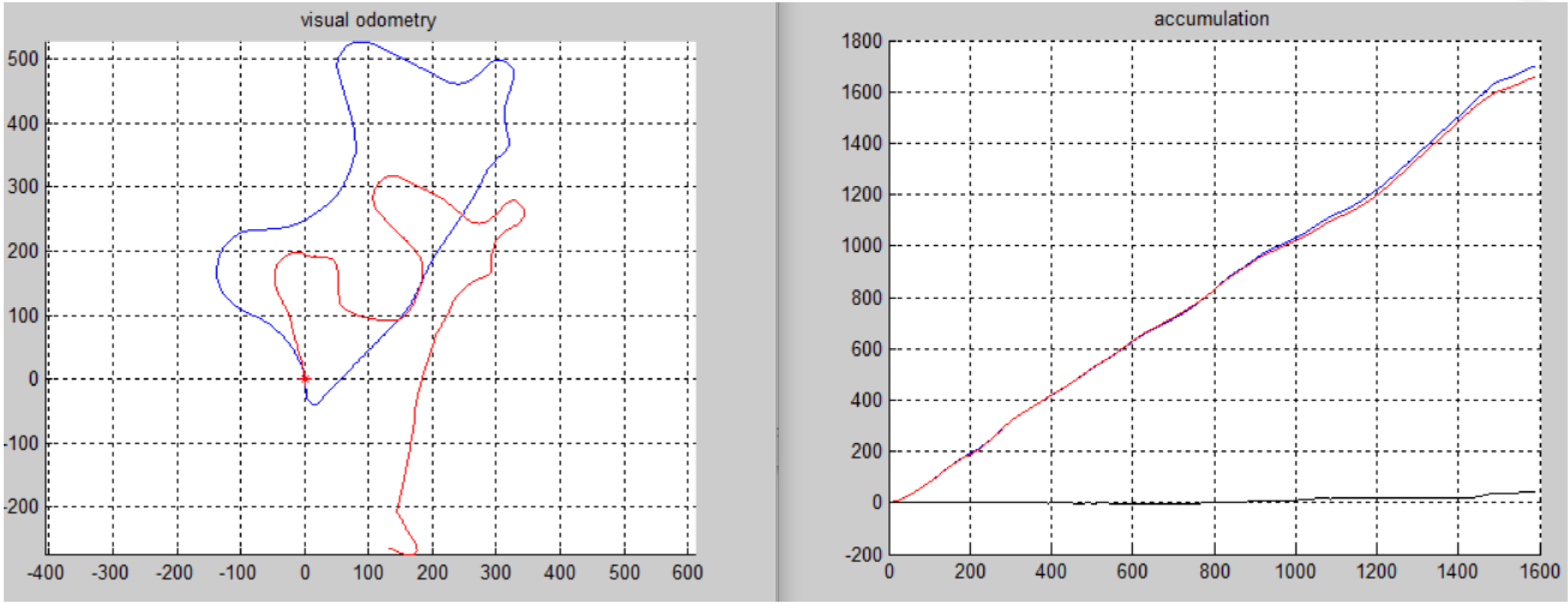
CWorkspace VCCodeFunction

```
(Name)        CWorkspace
Access        public
FullName      CWorkspace::CWorkspace
IsDefault     False
IsDelete      False
IsFinal       False
IsInjected    False
IsInline      False
IsOverloaded
IsSealed      False
IsTemplate    False
```

Implementation

True	🟦
Estimate	🟥 🟩

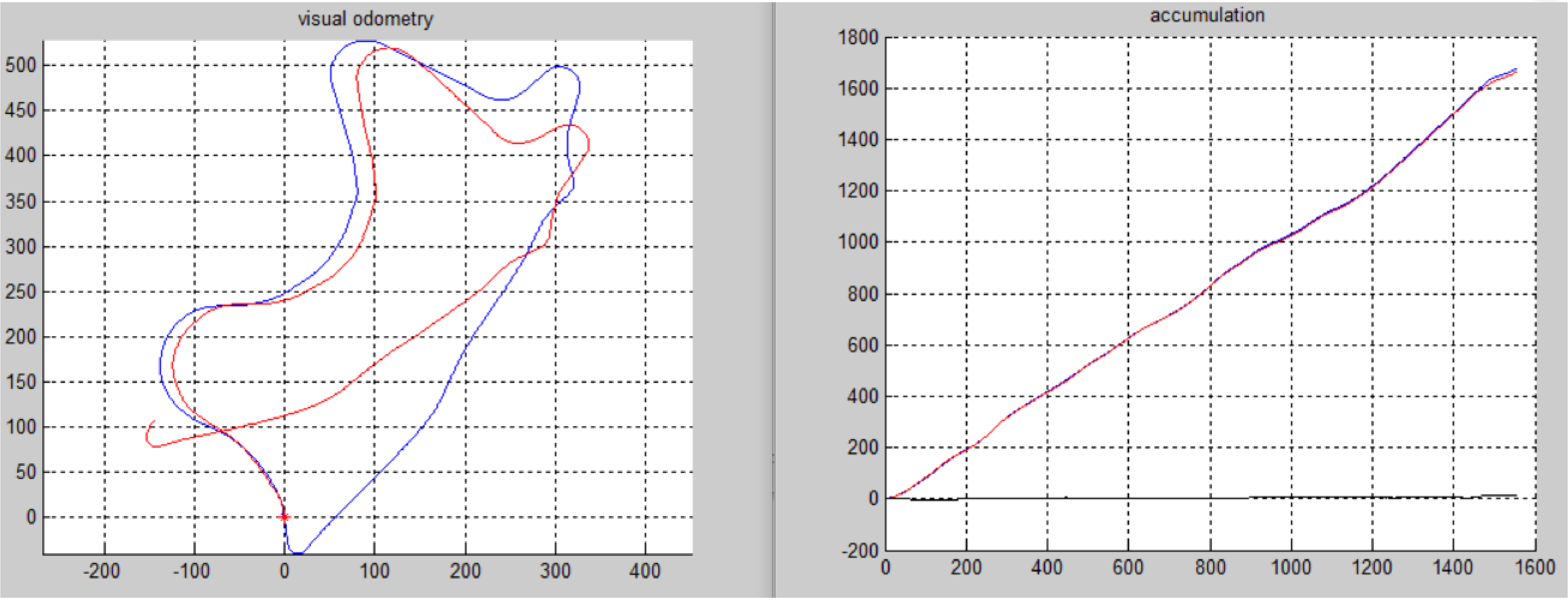
- **result**



Implementation

True	🟦
Estimate	🟥 🟩

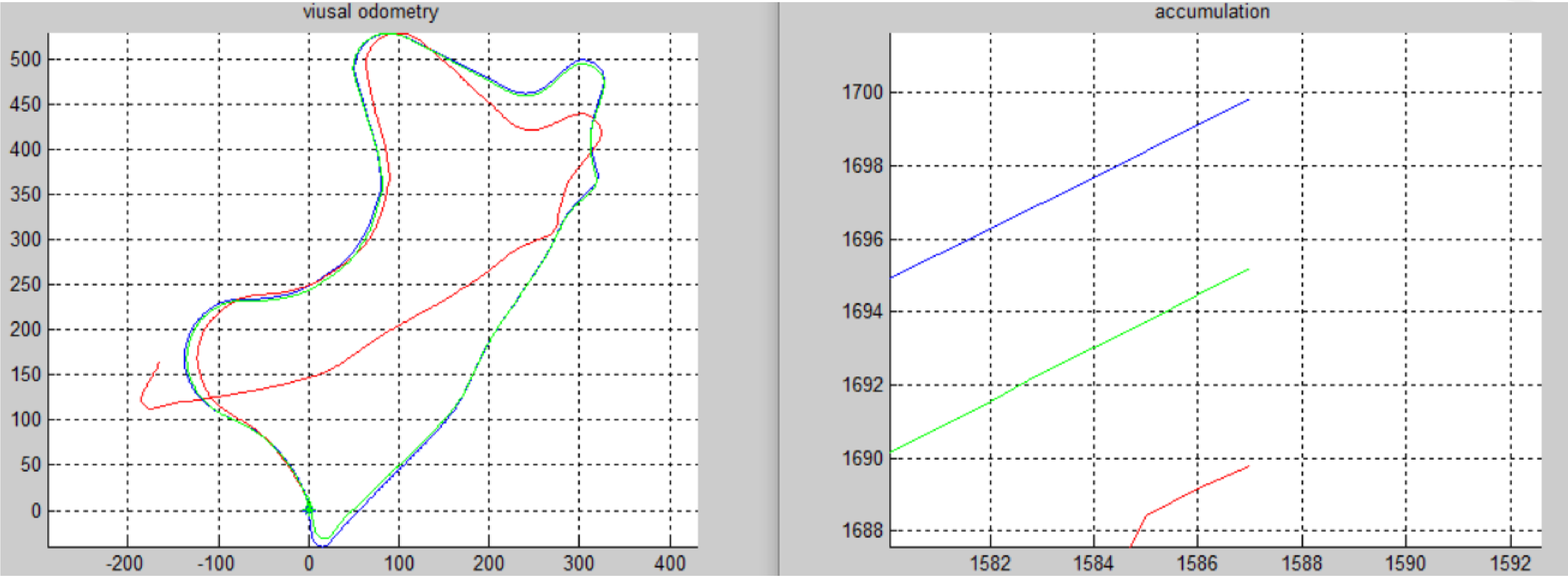
- **result**



Implementation

True	🟦
Estimate	🟥 🟩

- **result**



RAF

- Future Work
  - Detection
    - ➢ Adaptive Feature Detection
  - Interpolation
    - ➢ Plane Interpolation
  - Filtering
    - ➢ If the mean value of the current frame's OF >> mean value of the previous N frame's OF, Do not work -> ex) car
    - ➢ If the size and mean age are reduced suddenly, Do not work
    - ➢ If the high Age RAF frame number << current frame number, Delete RAF

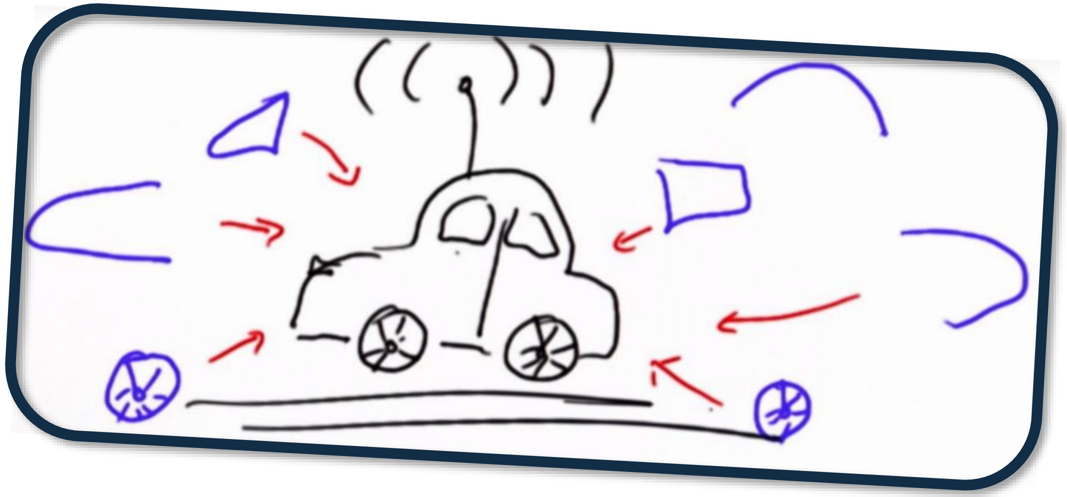
RAF

- Future Work
  - Aging
    - ➢ Outlier De-Aging
  - Pose Optimize
    - ➢ Graph SLAM
    - ➢ Loop Closing
    - ➢ Make Weight Function

Q&A

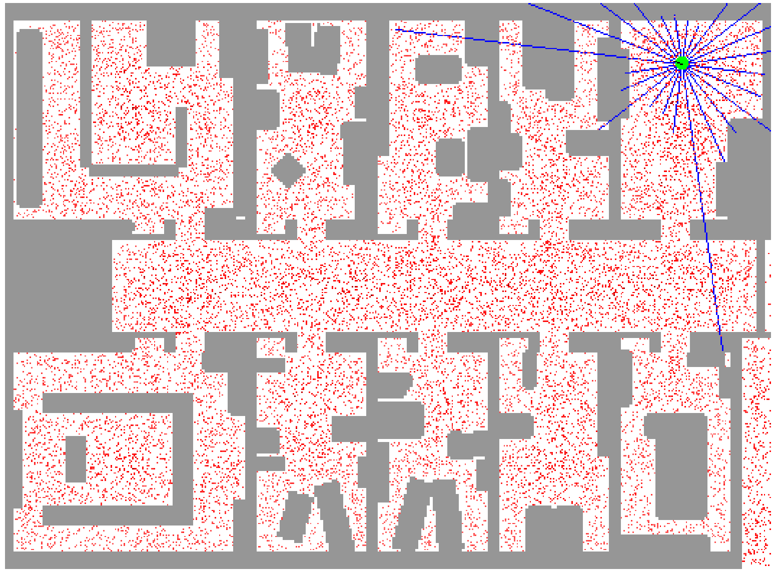
SLAM : Self-driving

- Overview
  - Localization
  - Mapping
  - Planning
  - Control



SLAM : Self-driving

- Localization
  - Given
    - Map
    - Sensor data
    - Robot kinematics
  - Goal
    - Find robot position
  - ex) Histogram Filter, Kalman Filter, Particle Filter



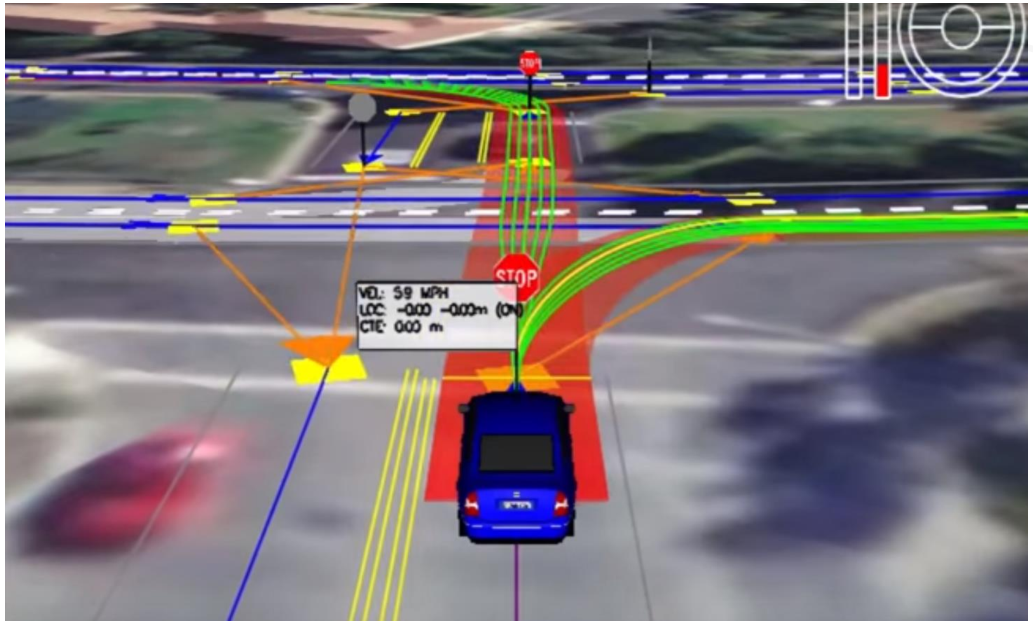
Particle Filter Localization

SLAM : Self-driving

- Tracking

SLAM : Self-driving

- Planning
  - Given
    - ➢ Map
    - ➢ Starting position
    - ➢ Goal position
    - ➢ Cost
  - Goal
    - ➢ Find minimum cost path
  - ex) A*(Path), Dynamic Programing(Policy)



Path Planning

SLAM : Self-driving

- Control
  - Path is known
  - Determine the actual motion commands
  - ex) PID

SLAM

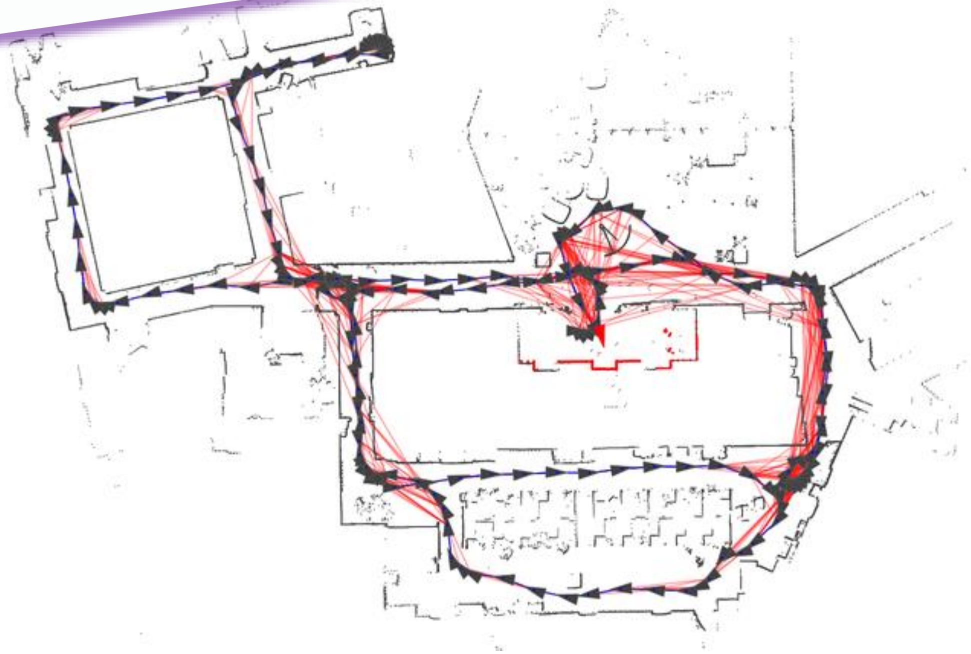
- What is SLAM?

A robot is exploring a

Self-Driving!

- Given
  - Robot motions
  - Observations of nearby features

- Estimate
  - Map
  - Pose (Position + Orientation)
  - (Path Planning)



Graph SLAM