# Indirect Visual Odometry with Optical Flow

Chenguang Huang, Andong Tan

München, 22. July 2019

# Overview

# 1 Introduction

## 1.1 Lukas Kanade method for optical flow

Input: two images, key points in one image

Output: the key points in the other image(the motion of the key points).

Input key point position in one image

Output key point position in the other image

# 1 Introduction

## 1.1 Lukas Kanade method for optical flow

Inspiration in SLAM:

replace feature descriptors matching with optical flow method!

Input key point position in one image

Output key point position in the other image

# 1 Introduction

## 1.1 Lukas Kanade method for optical flow

Basic mathematical principle

A method to estimate the motion of object in image. It is based on one assumption:

the brightness of a moving point remains constant over time

$$I(x(t), t) = \text{const.} \quad \forall t \in [0, T]$$

Derivative with respect to time

$$\frac{d}{dt} I(x(t), t) = \nabla I(x(t), t) \frac{dx(t)}{dt} + \frac{\partial I(x(t), t)}{\partial t} = 0 \quad \forall t \in [0, T]$$

# 1 Introduction

## 1.1 Lukas Kanade method for optical flow

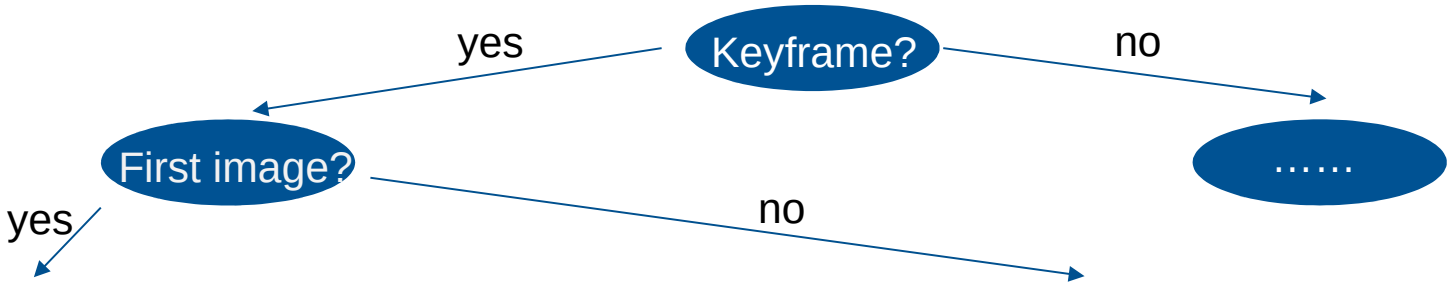
$$\frac{d}{dt} I(x(t), t) = \nabla I(x(t), t) \frac{dx(t)}{dt} + \frac{\partial I(x(t), t)}{\partial t} = 0 \quad \forall t \in [0, T]$$

Solve this function for v

$$\boxed{\nabla I^{\top} v + I_t = 0}$$

# 1 Introduction

Keypoints detection in every key frame:



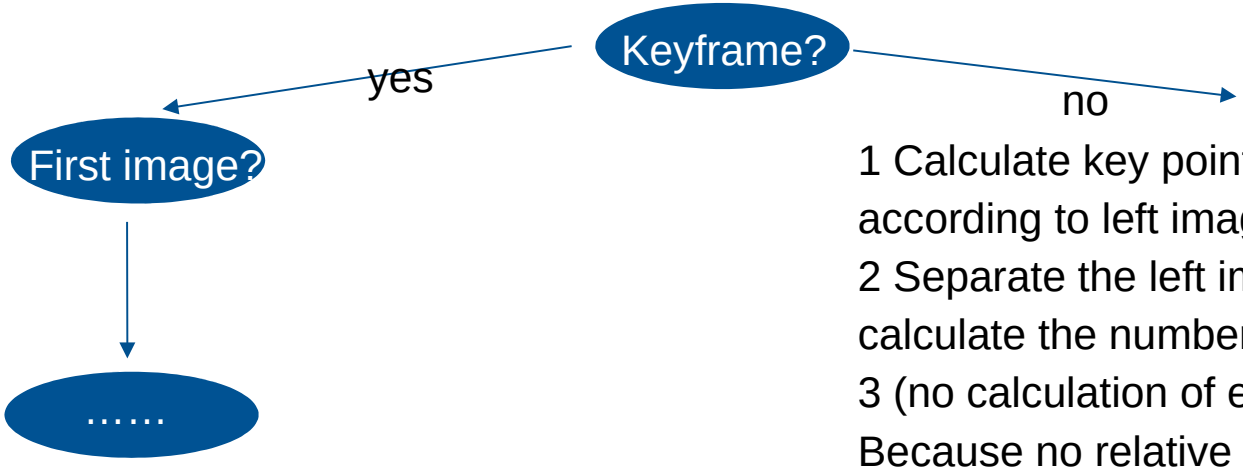

1 Key points detection in left image
2 Calculate key points in right image according to left image (OF)
3 Compute essential matrix, find inlier
4 Localize camera
5 Initialize landmarks

1 Calculate key points in left image according to left image in last time step (OF)
2 Separate the left image into cells and detect new points in empty cells
3 Calculate the keypoints in right image (OF) according to left image
4 Compute essential matrix and find inlier
5 Localize camera
6 Create new landmarks
7 Optimization
(8 frame-frame triangulation)

OF = Optical Flow

# 1 Introduction

Use optical flow and keypoints detection in key frame:




1 Calculate key points in left image according to left image in last time step (OF)
2 Separate the left image into cells and calculate the number of empty cells
3 (no calculation of essential matrix) Because no relative pose between frame

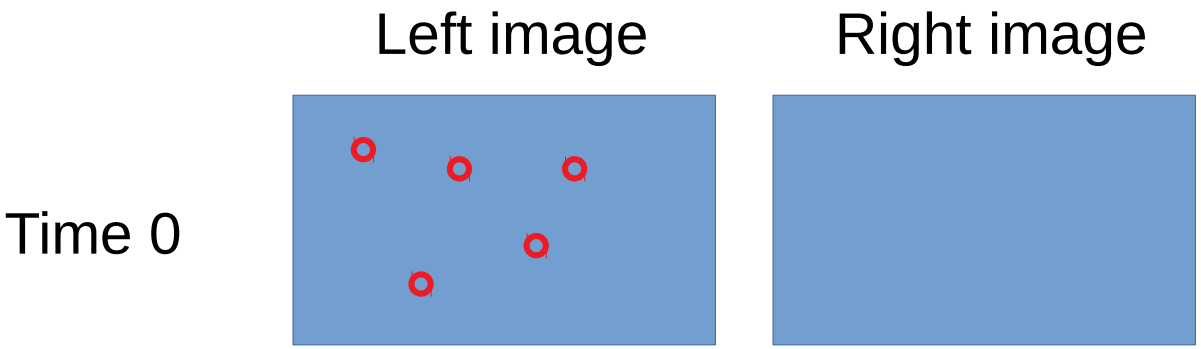
4 Localize camera
6 If not enough key points or inliers or too many empty cells && optimization is not running and not finished, set the next frame as key frame

OF = Optical Flow

# 1 Introduction

## 1.2 Project pipeline: first key frame

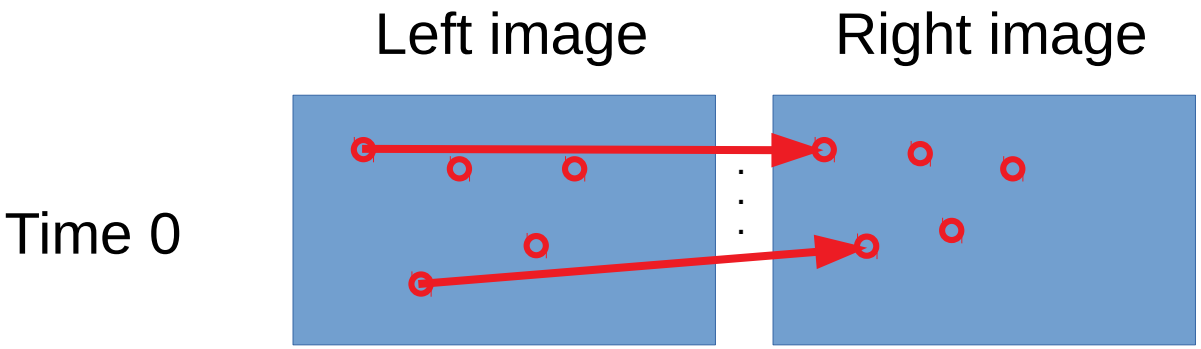
Step 1: detect key points in left image



# 1 Introduction

1.2 Project pipeline: first key frame

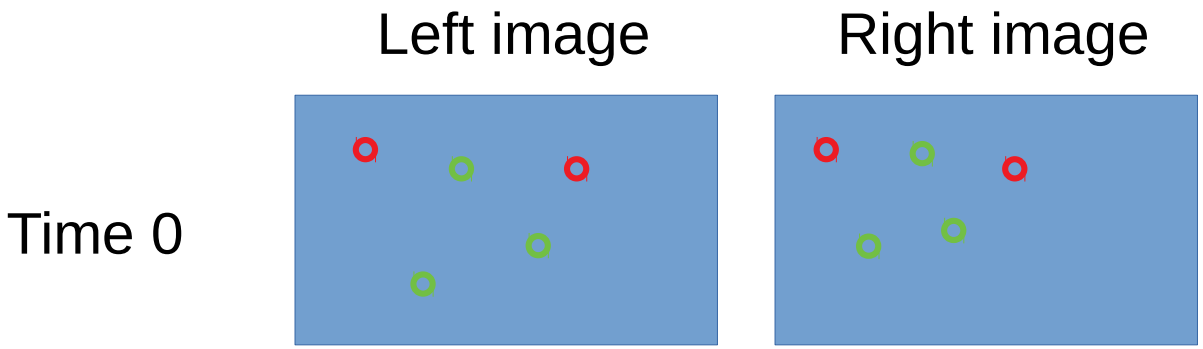
Step 2: use optical flow to get corresponding key points in right image



# 1 Introduction

## 1.2 Project pipeline: first key frame

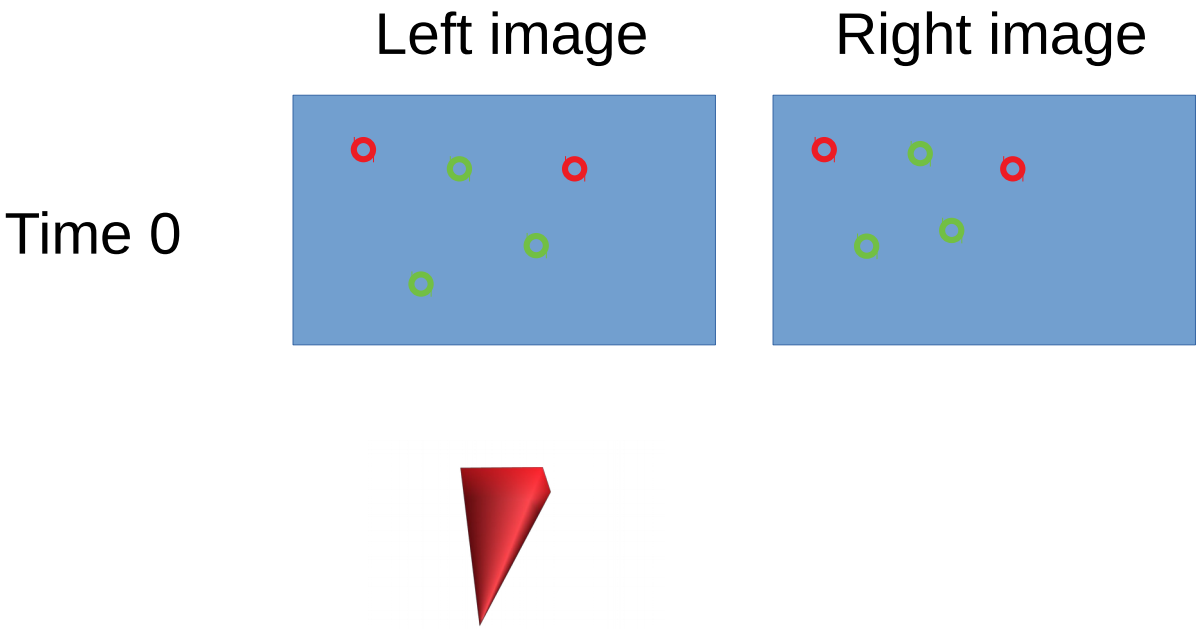
Step 3: check epipolar constraint and find inlier



# 1 Introduction

1.2 Project pipeline: first key frame

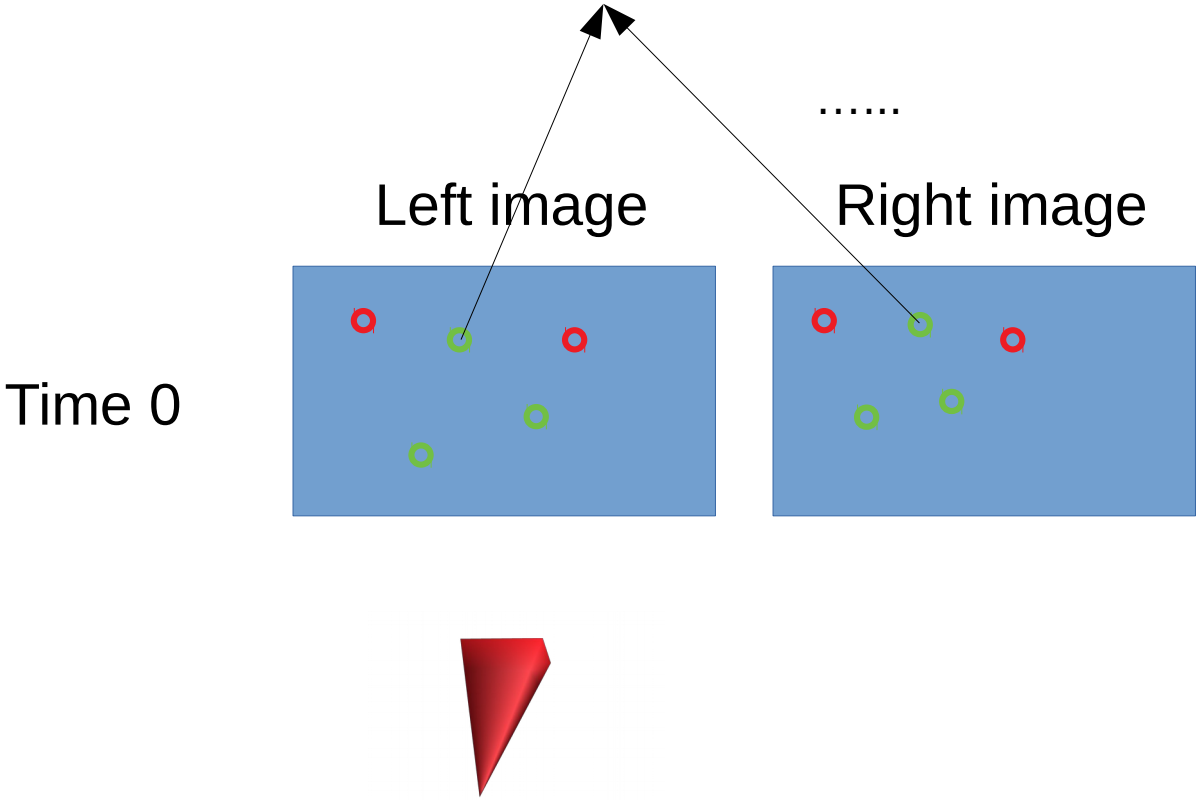
Step 4: localize camera(initialize the first left camera pose with identity matrix)



# 1 Introduction

## 1.2 Project pipeline: first key frame

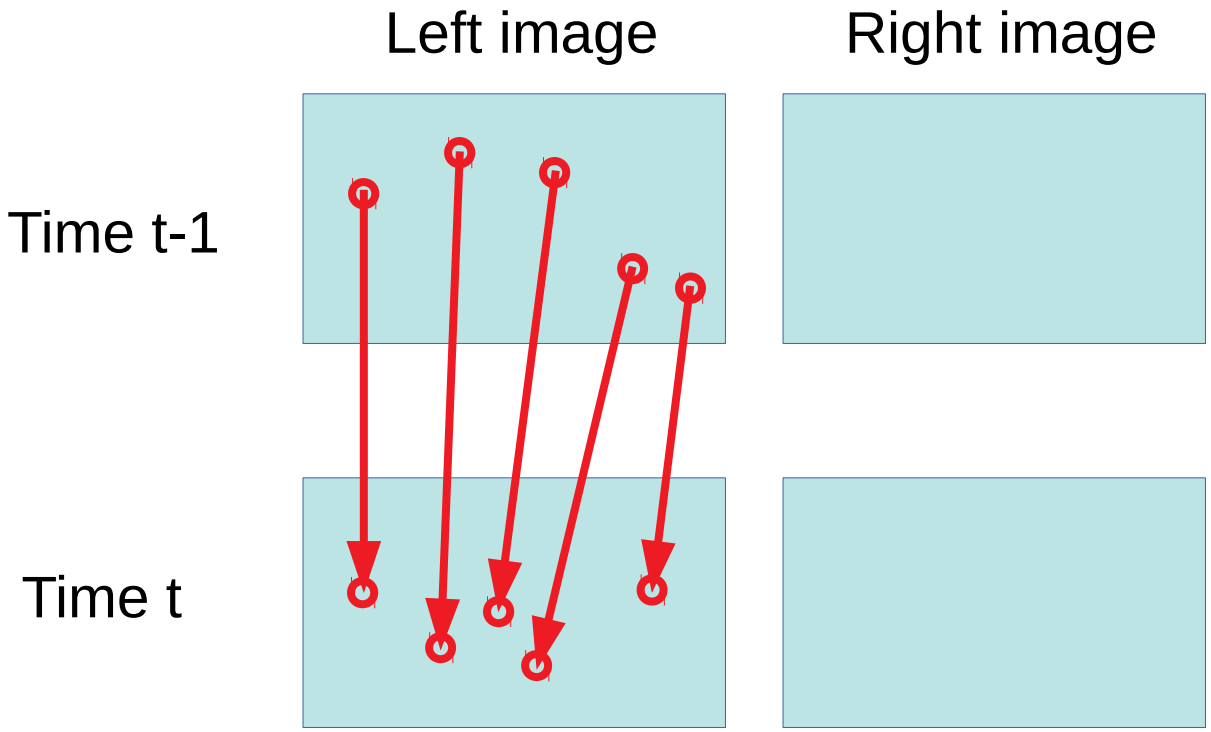
Step 5: triangulate landmarks



# 1 Introduction

1.2 Project pipeline: second and later key frame

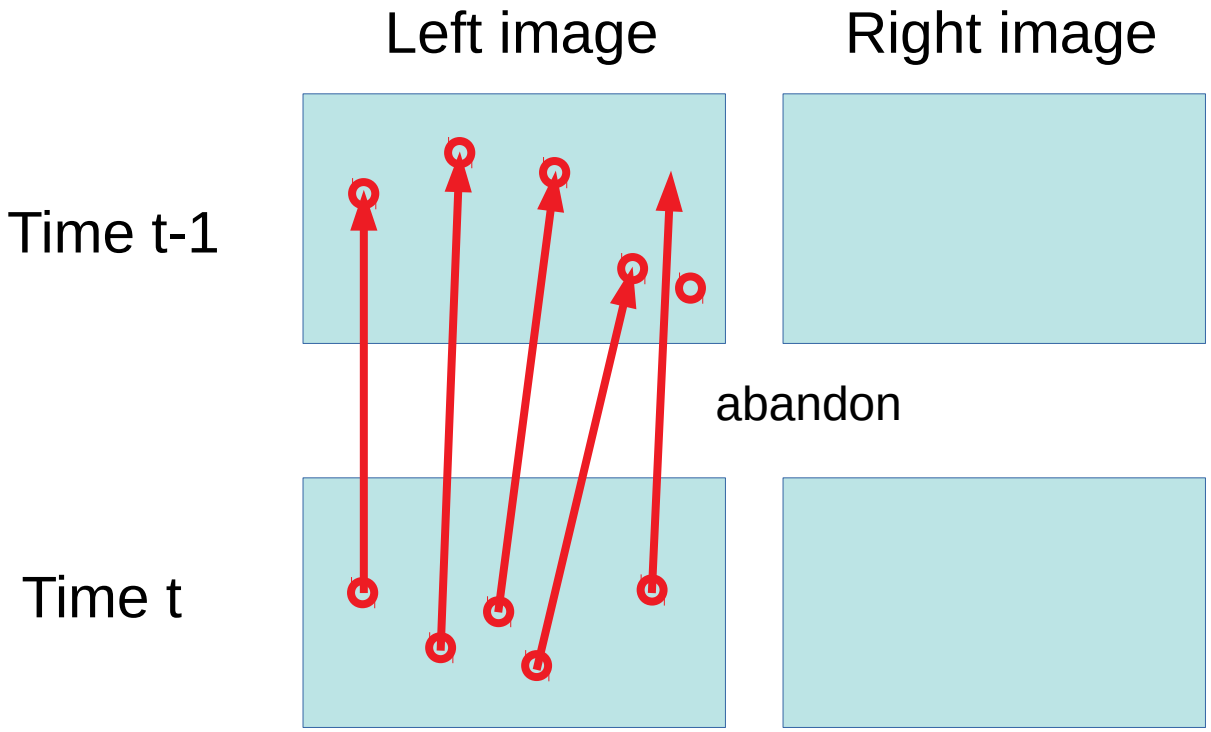
Step 1: use optical flow to get corresponding key points last time step of left frame



# 1 Introduction

1.2 Project pipeline: second and later key frame

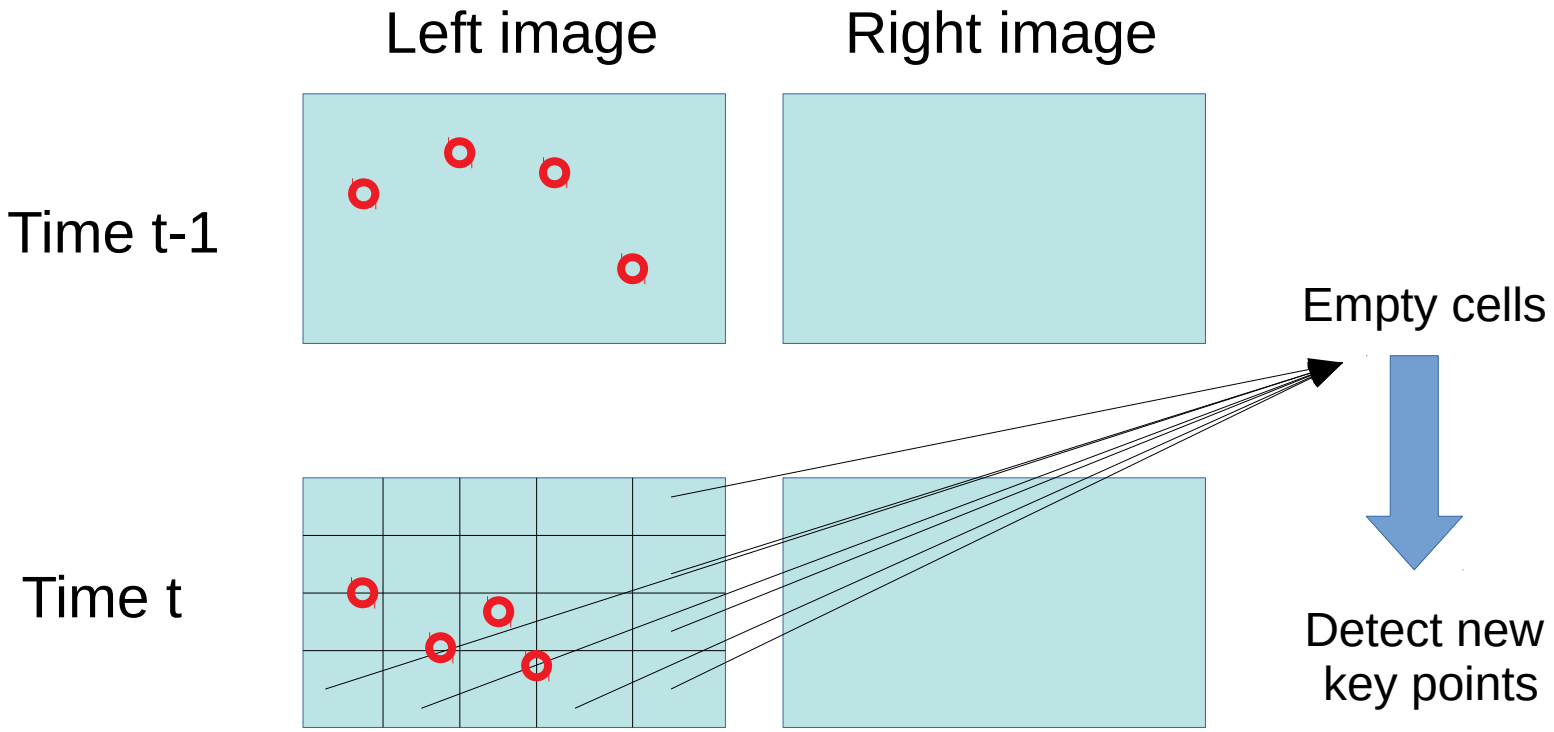
Step 1: use optical flow to get corresponding key points last time step of left frame(backward check)



# 1 Introduction

1.2 Project pipeline: second and later key frame

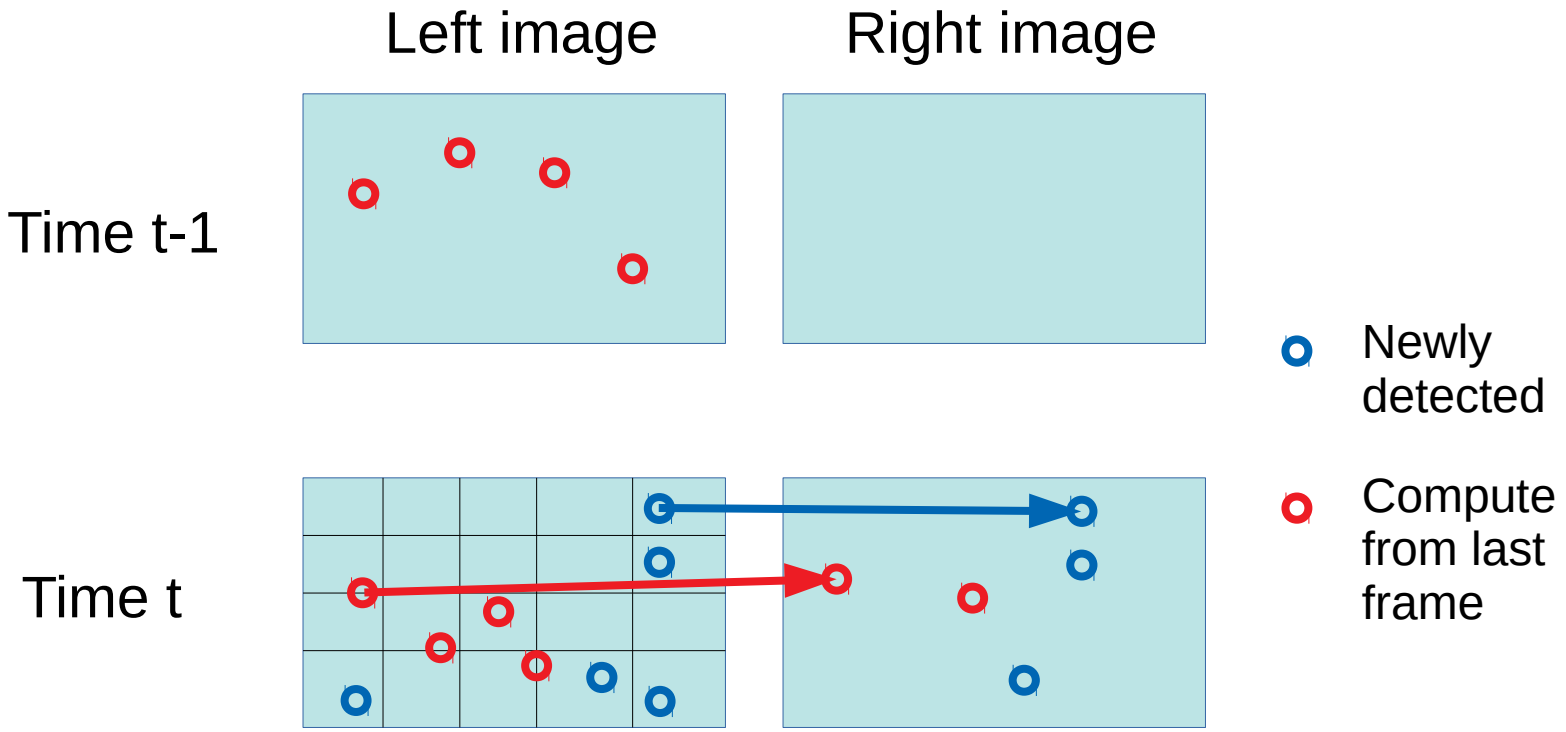
Step 2: make grid in left image and detect key points in empty cells



# 1 Introduction

1.2 Project pipeline: second and later key frame

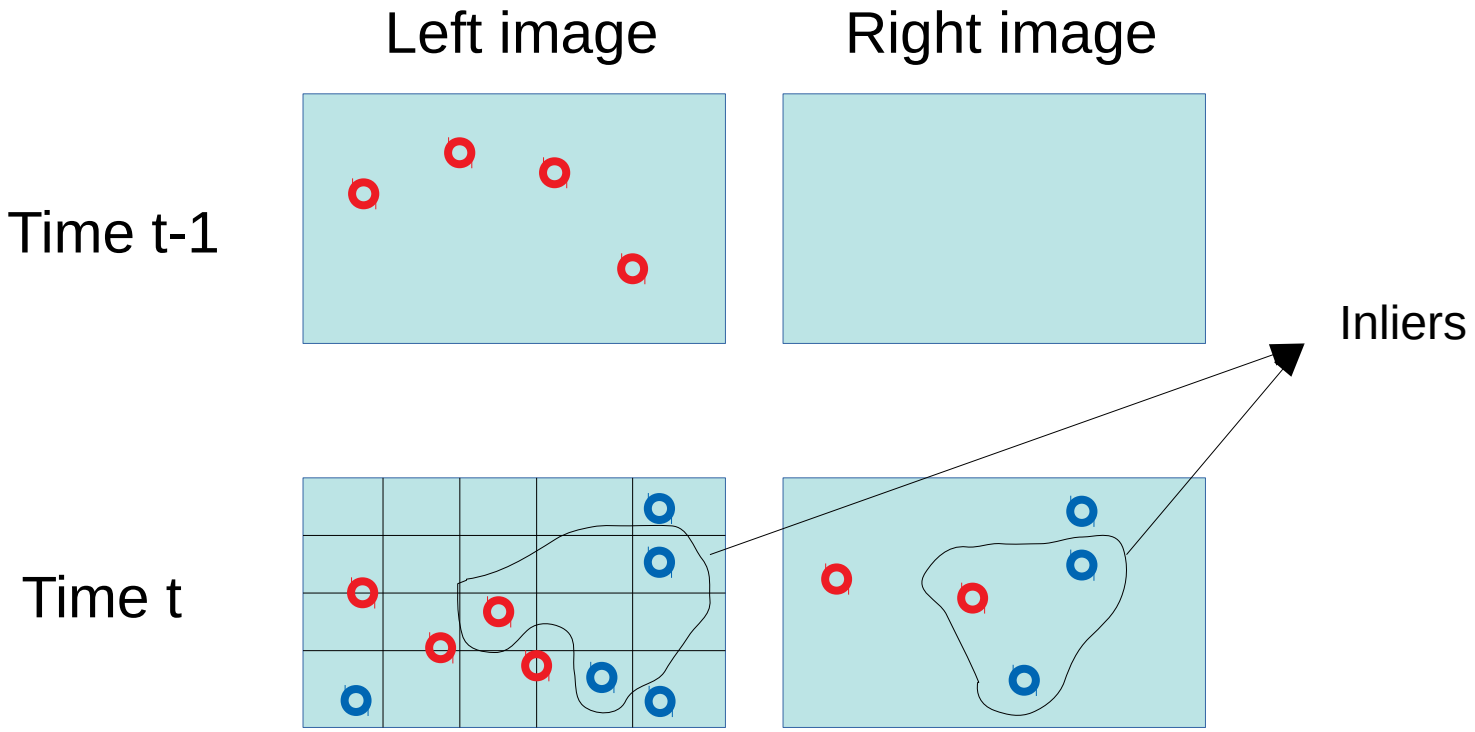
Step 3: use optical flow to find the corresponding key points in the right image



# 1 Introduction

## 1.2 Project pipeline: second and later key frame

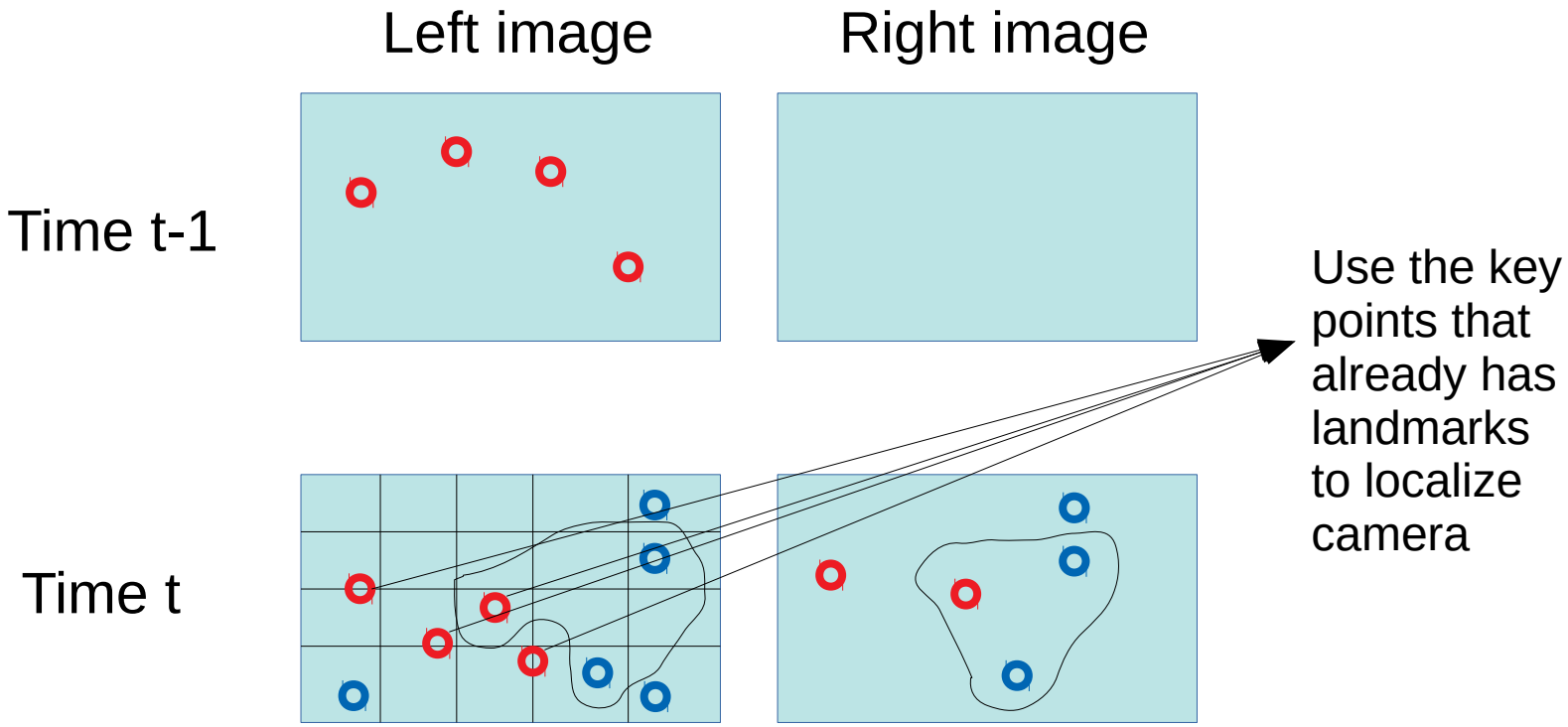
Step 4: use epipolar constraint to find inliers



# 1 Introduction

## 1.2 Project pipeline: second and later key frame

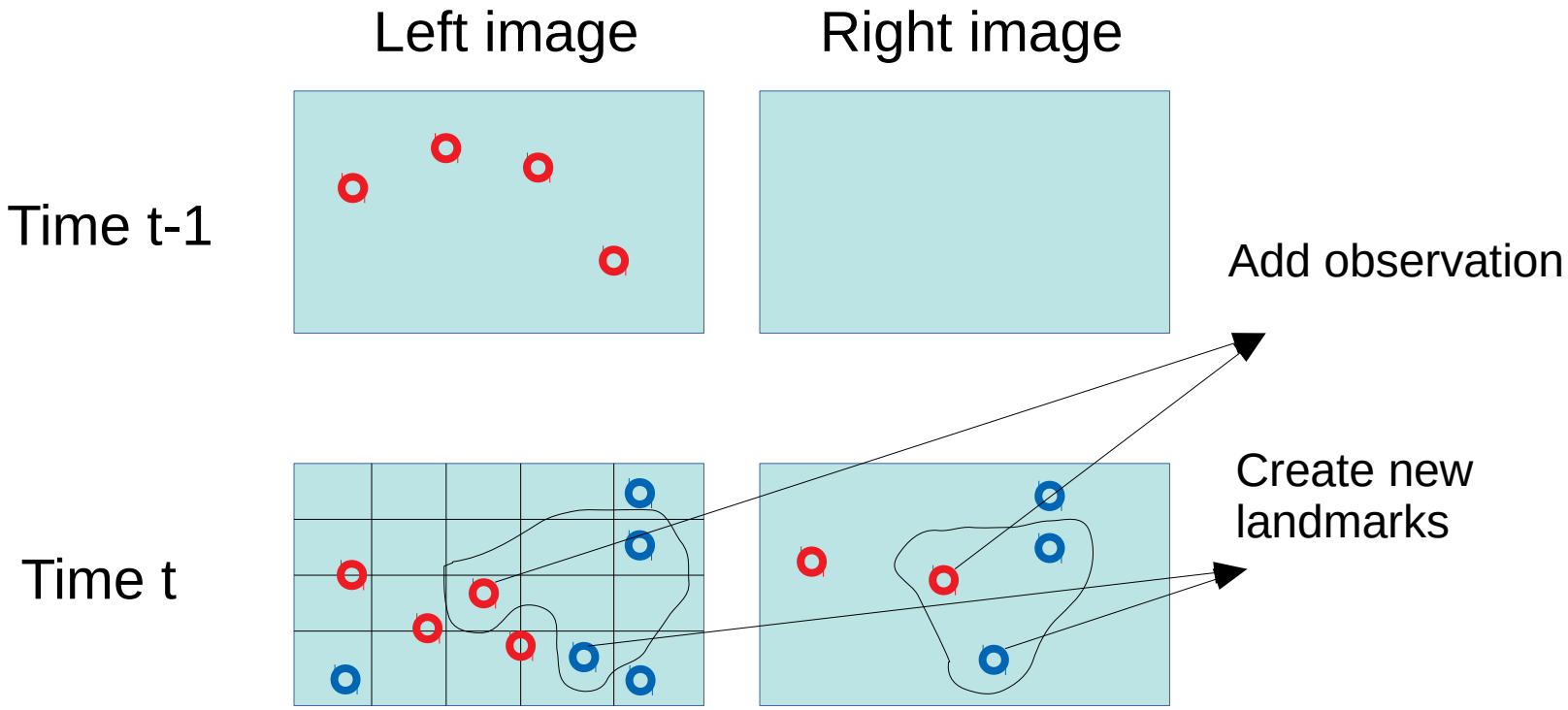
### Step 5: localize camera



# 1 Introduction

## 1.2 Project pipeline: second and later key frame

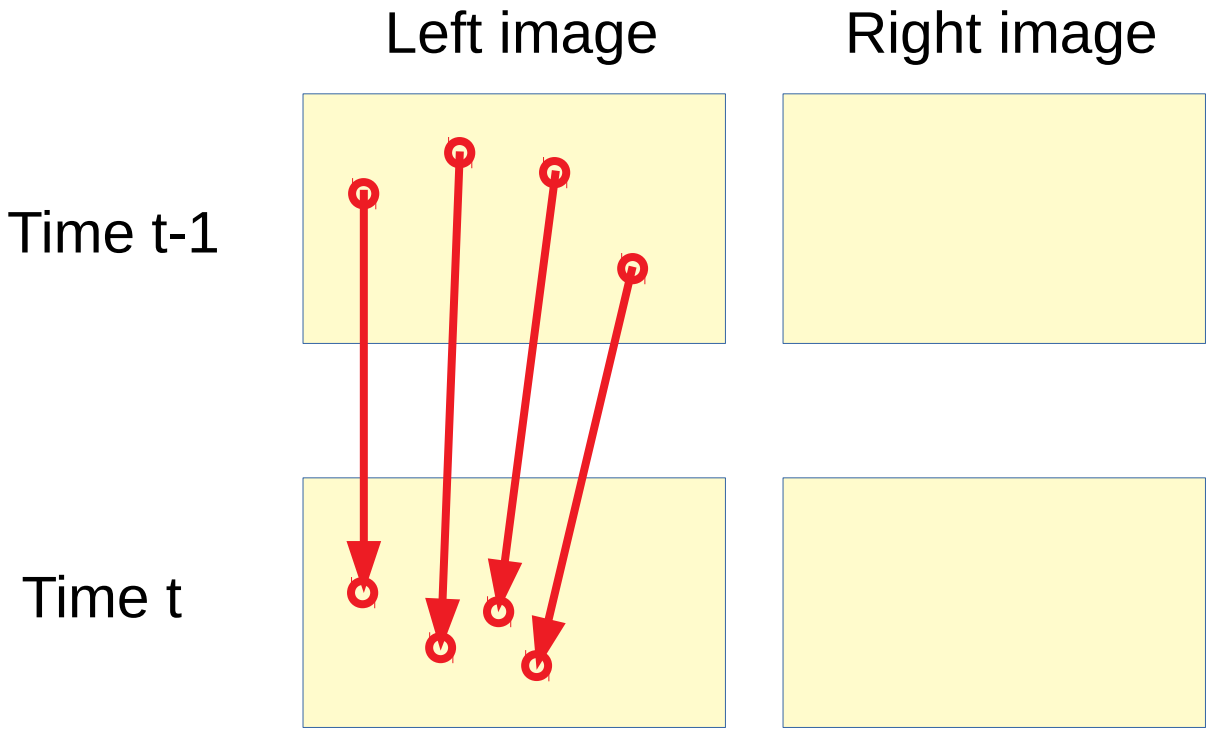
Step 6: add observation and create landmarks



# 1 Introduction

## 1.2 Project pipeline: non key frame

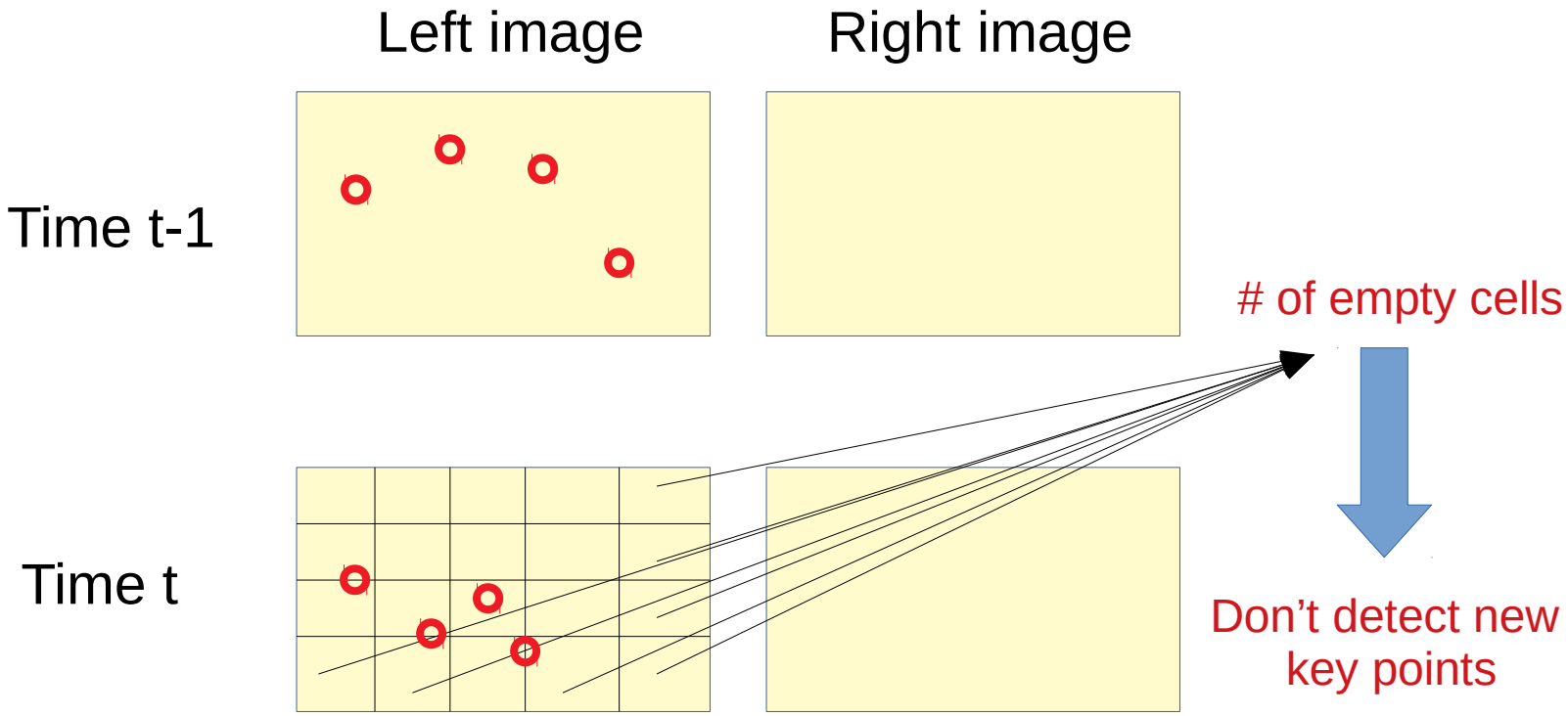
Step 1: use optical flow to get corresponding key points last time step of left frame



# 1 Introduction

1.2 Project pipeline: second and later key frame

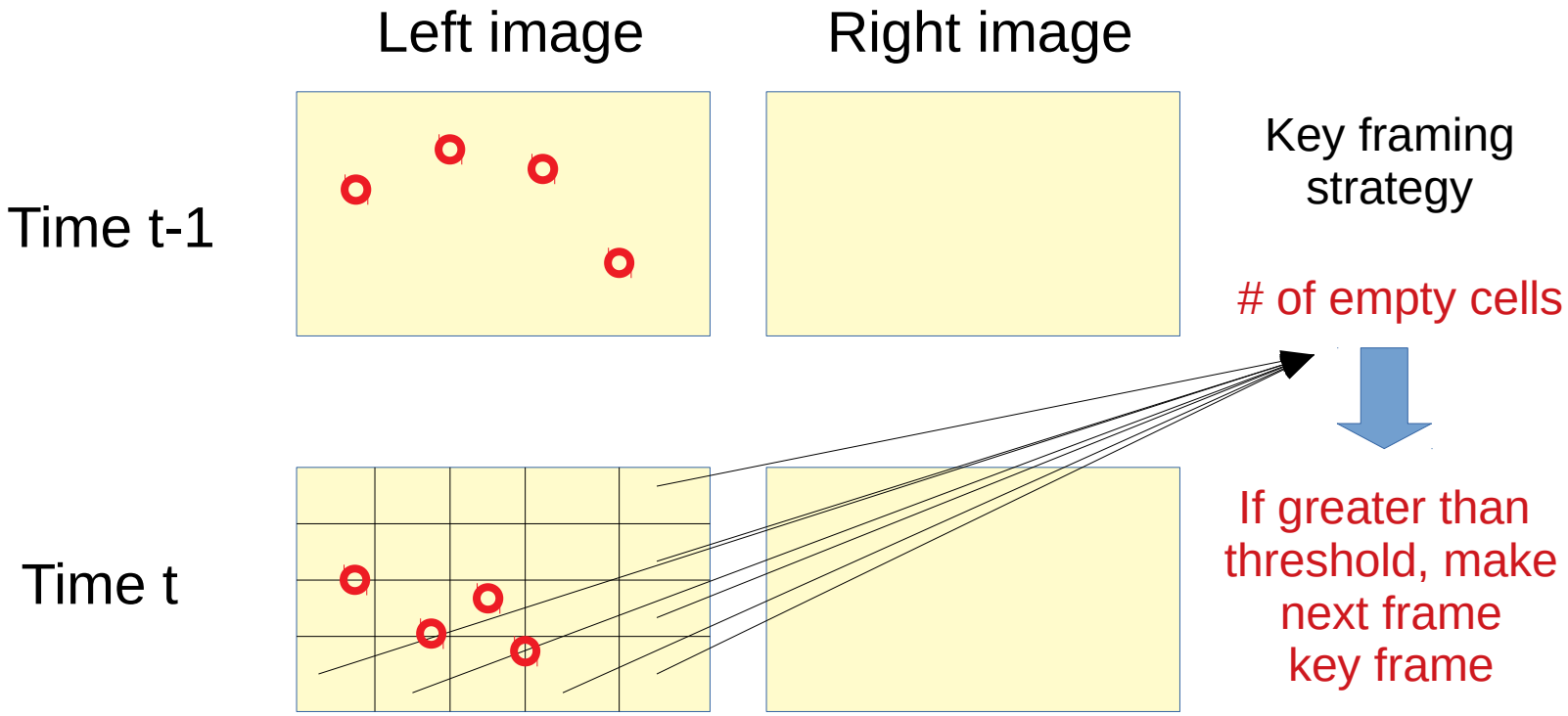
Step 2: make grid in left image and count the number of empty cells



# 1 Introduction

1.2 Project pipeline: second and later key frame

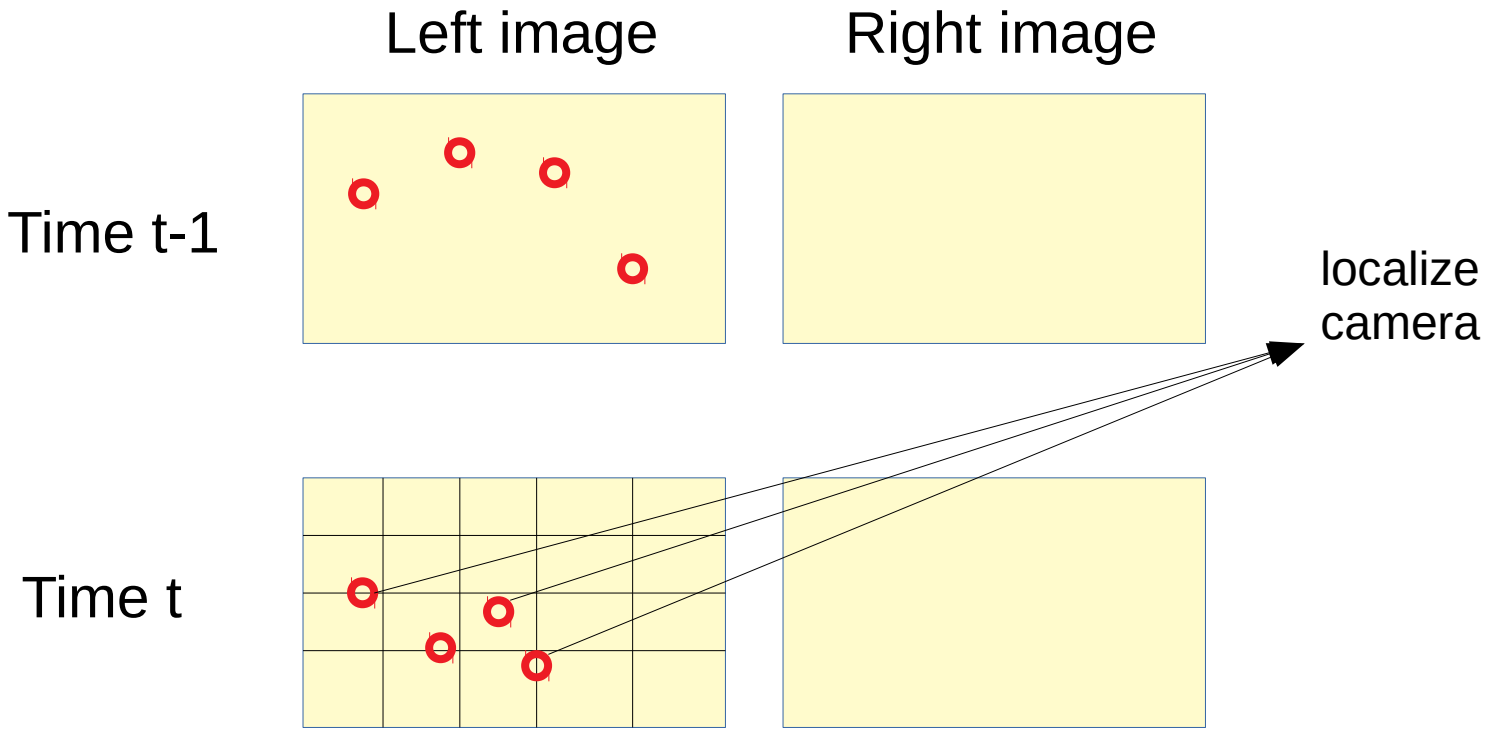
Step 2: make grid in left image and count the number of empty cells



# 1 Introduction

## 1.2 Project pipeline: second and later key frame

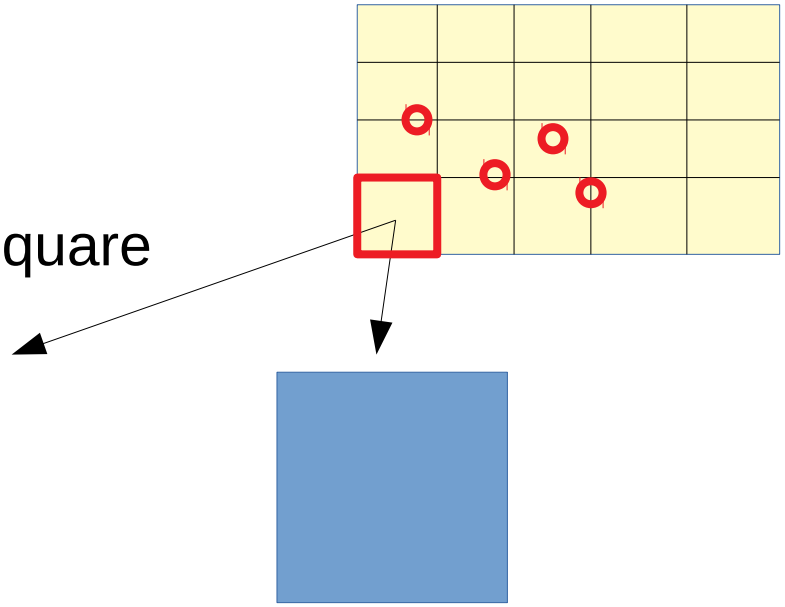
Step 5: localize camera



# 1 Introduction

## 1.3 Parameter and strategy

### 1.3.1 Grid shape: retangle vs square



| | | |
|---|---|---|
| pixel x pixel | 30 x 47 | 32 x 32 |
| rmse | 0.14 | 0.11 |

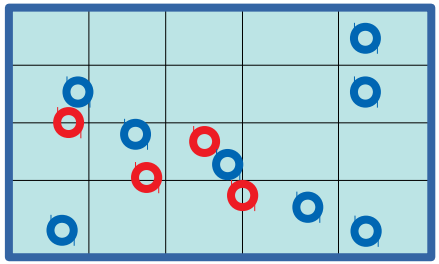
# 1 Introduction

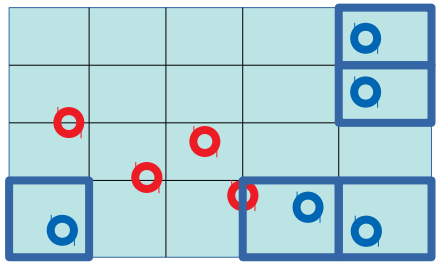
## 1.3 Parameter and strategy

### 1.3.2 Key points detection: in whole image vs in each cell

Detect in whole image

Then only keep key points in empty cells





Detect in empty cells respectively

rmse

0.1139

0.1116

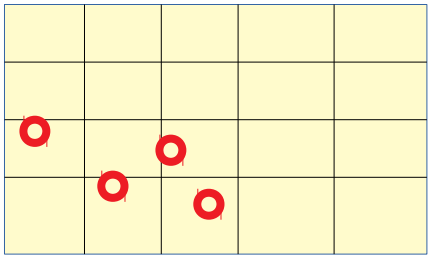
# 1 Introduction

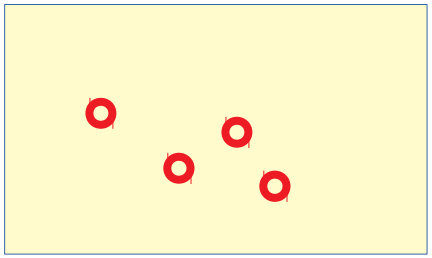
## 1.3 Parameter and strategy

### 1.3.3 Key framing: use grid vs use only inliers number

# of empty cells

If greater than threshold, make next frame key frame





# of localization inliers

If smaller than threshold, make next frame key frame

| | | |
|---|---|---|
| rmse | 0.11 | 0.43 |

# 2 Result

## 2.1 Precision comparison

Basic parameters

| image height (pixel) | image width (pixel) | grid size (pixel x pixel) | Minimum number of key points |
|---|---|---|---|
| 480 | 752 | 32 x 32 | 100 |

Strategy

1. Square grid
2. Add key points in each cell
3. Use empty cells number to decide next key frame

# 2 Result

## 2.1 Precision comparison

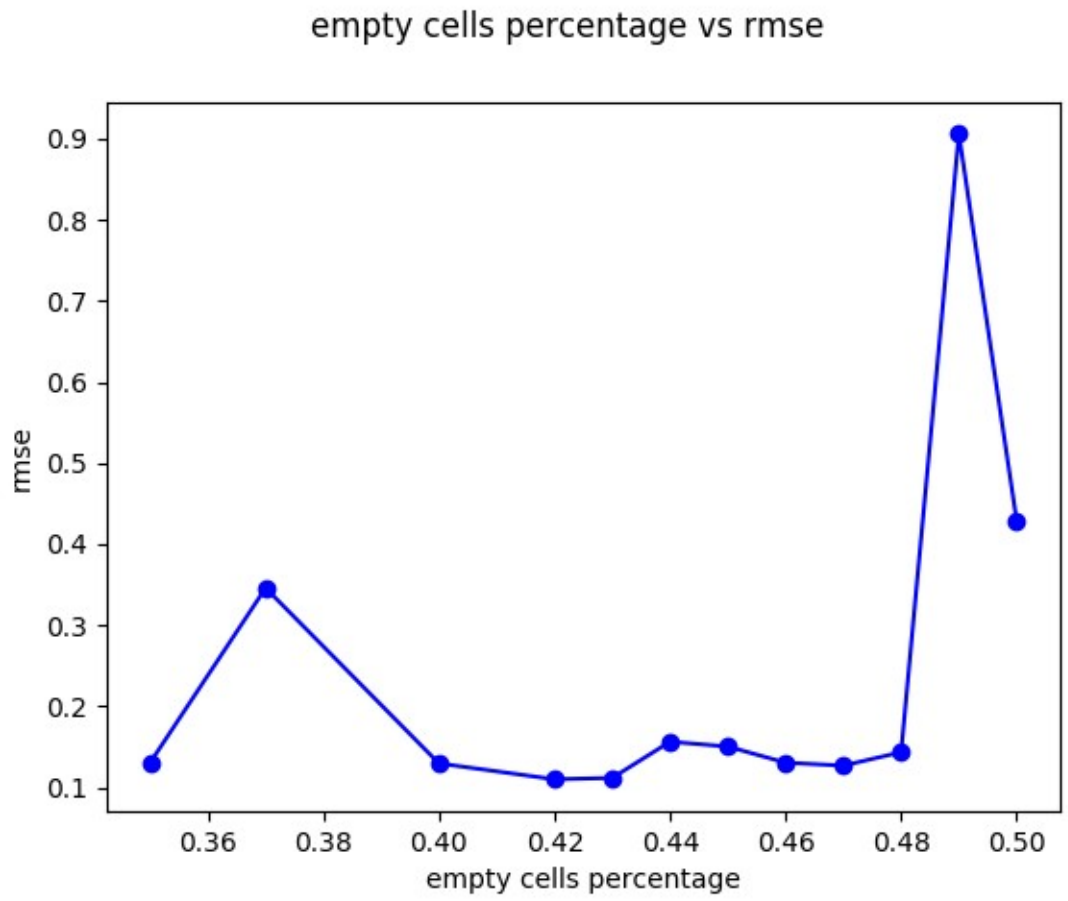
Variation parameters: maximum empty cells percentage*

| Maximum empty cells number | 0.35 | 0.4 | 0.42 | 0.44 | 0.46 | Exercise 5 |
|---|---|---|---|---|---|---|
| rmse | 0.1303 | 0.1295 | **0.1099** | 0.1563 | 0.1304 | **0.1012** |

*maximum empty cells percentage: when the empty cells percentage exceeds this value and optimization is finished, make next frame key frame

# 2 Result

## 2.1 Precision comparison



# 2 Result

## 2.2 Execution time comparison

### 2.2.1 Detection time

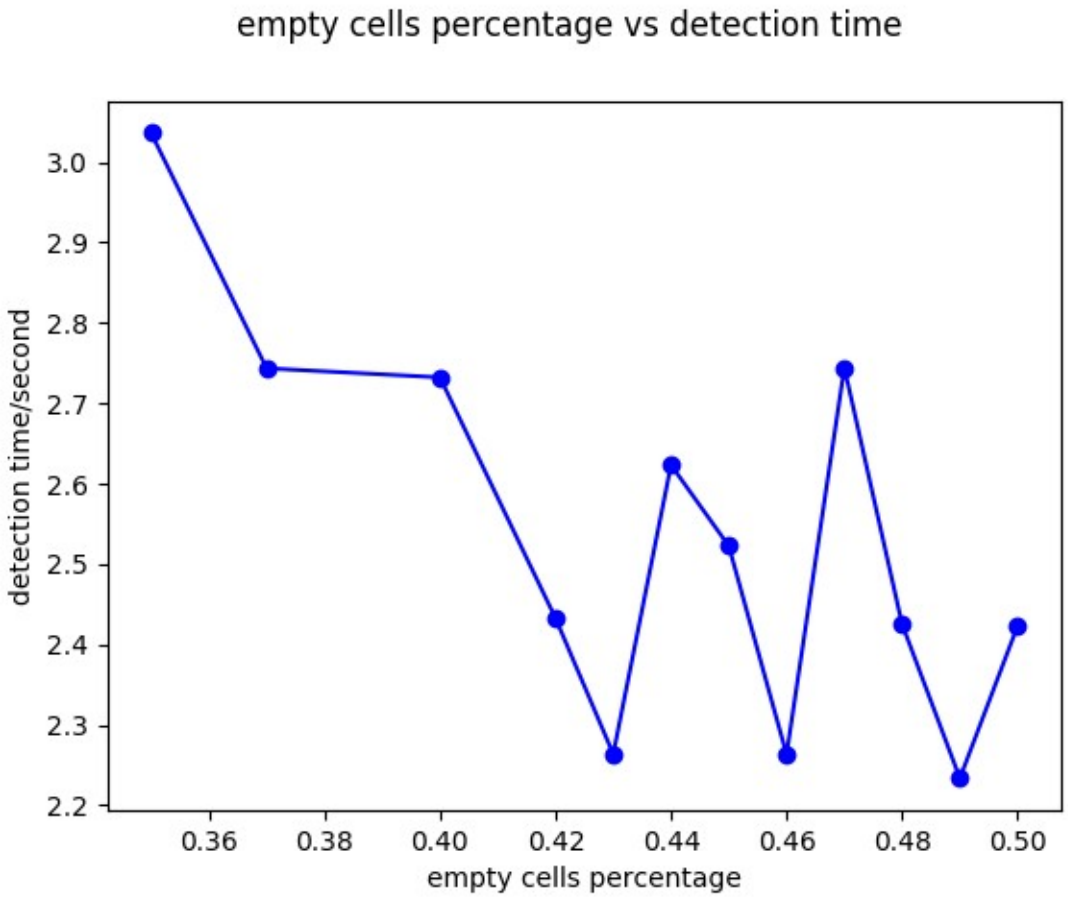
Variation parameters: maximum empty cells percentage*

| Maximum empty cells number | 0.35 | 0.4 | 0.42 | 0.44 | 0.46 | Exercise 5 |
|---|---|---|---|---|---|---|
| detection time (seconds) | 3.03643 | 2.73232 | 2.43286 | 2.62328 | 2.26326 | **44.0214** |

*maximum empty cells percentage: when the empty cells percentage exceeds this value and optimization is finished, make next frame key frame

# 2 Result

## 2.2 Execution time comparison



# 2 Result

## 2.2 Execution time comparison

### 2.2.2 Optimization time

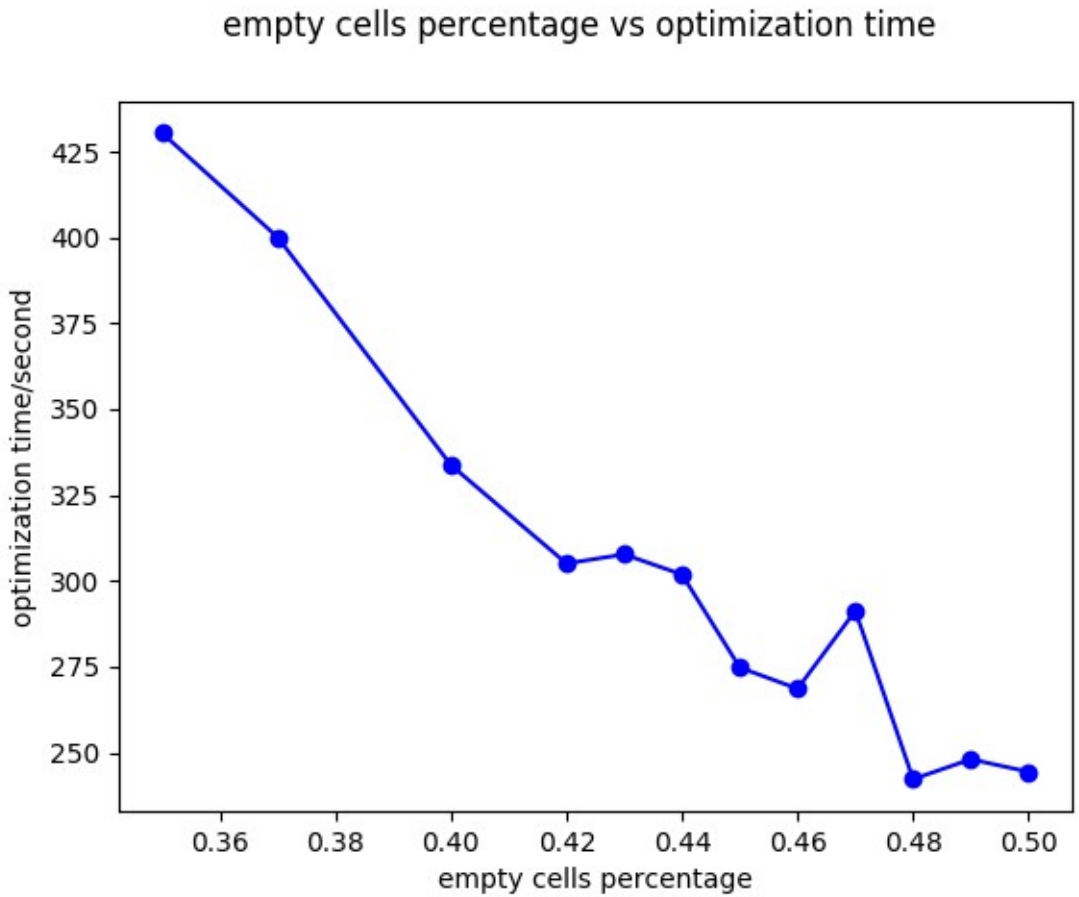
Variation parameters: maximum empty cells percentage*

| Maximum empty cells number | 0.35 | 0.4 | 0.42 | 0.44 | 0.46 | Exercise 5 |
|---|---|---|---|---|---|---|
| Opt time (seconds) | 430.423 | 333.723 | 305.043 | 301.82 | 268.572 | **65.8619** |
| # landmarks | 357718 | 306608 | 290657 | 272688 | 257608 | **221886** |
| # observation | 2806876 | 2120591 | 1939460 | 1715910 | 1545726 | **559563** |

*maximum empty cells percentage: when the empty cells percentage exceeds this value and optimization is finished, make next frame key frame

# 2 Result

## 2.2 Execution time comparison



# 2 Result

## 2.2 Execution time comparison

### 2.2.3 Frame to frame optical flow time

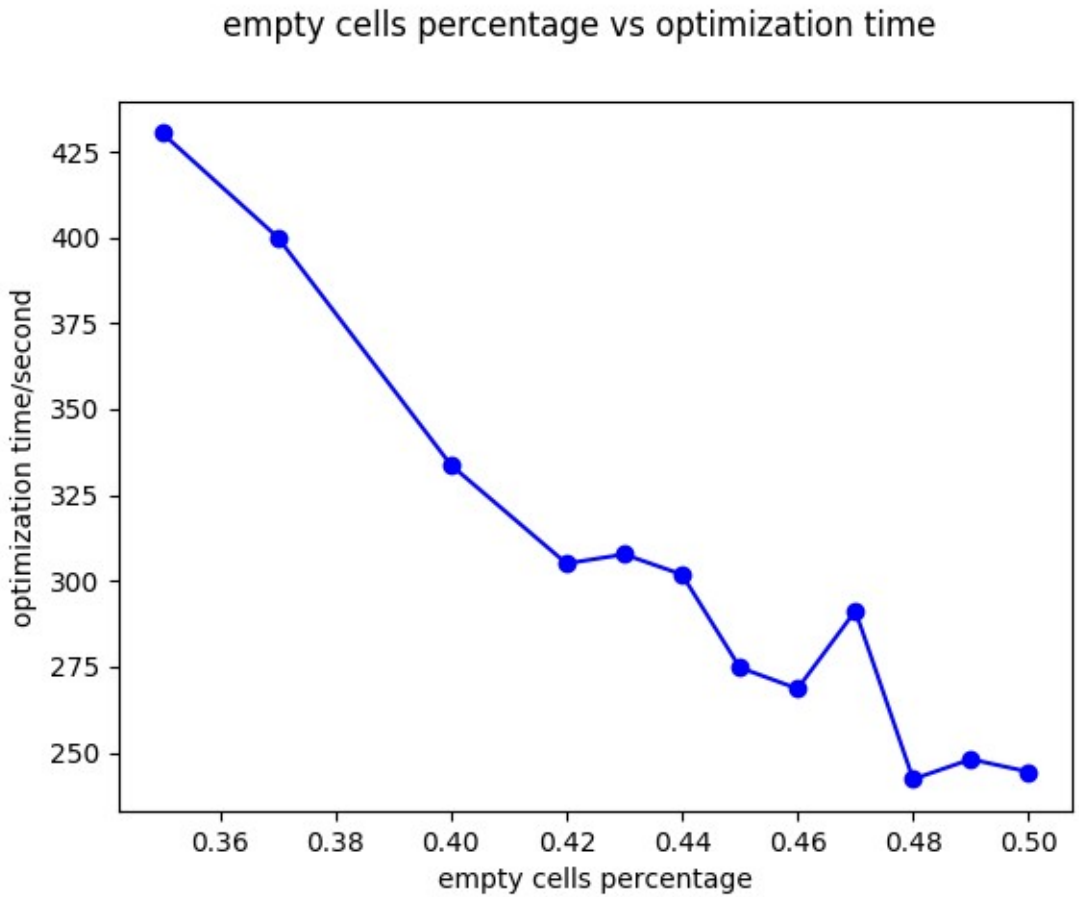
Variation parameters: maximum empty cells percentage*

| Maximum empty cells number | 0.35 | 0.4 | 0.42 | 0.44 | 0.46 | Exercise 5 feature matching time |
|---|---|---|---|---|---|---|
| f2f optical flow time (seconds) | 143.192 | 133.53 | 117.438 | 135.881 | 120.76 | **5.79** |

*maximum empty cells percentage: when the empty cells percentage exceeds this value and optimization is finished, make next frame key frame

# 2 Result

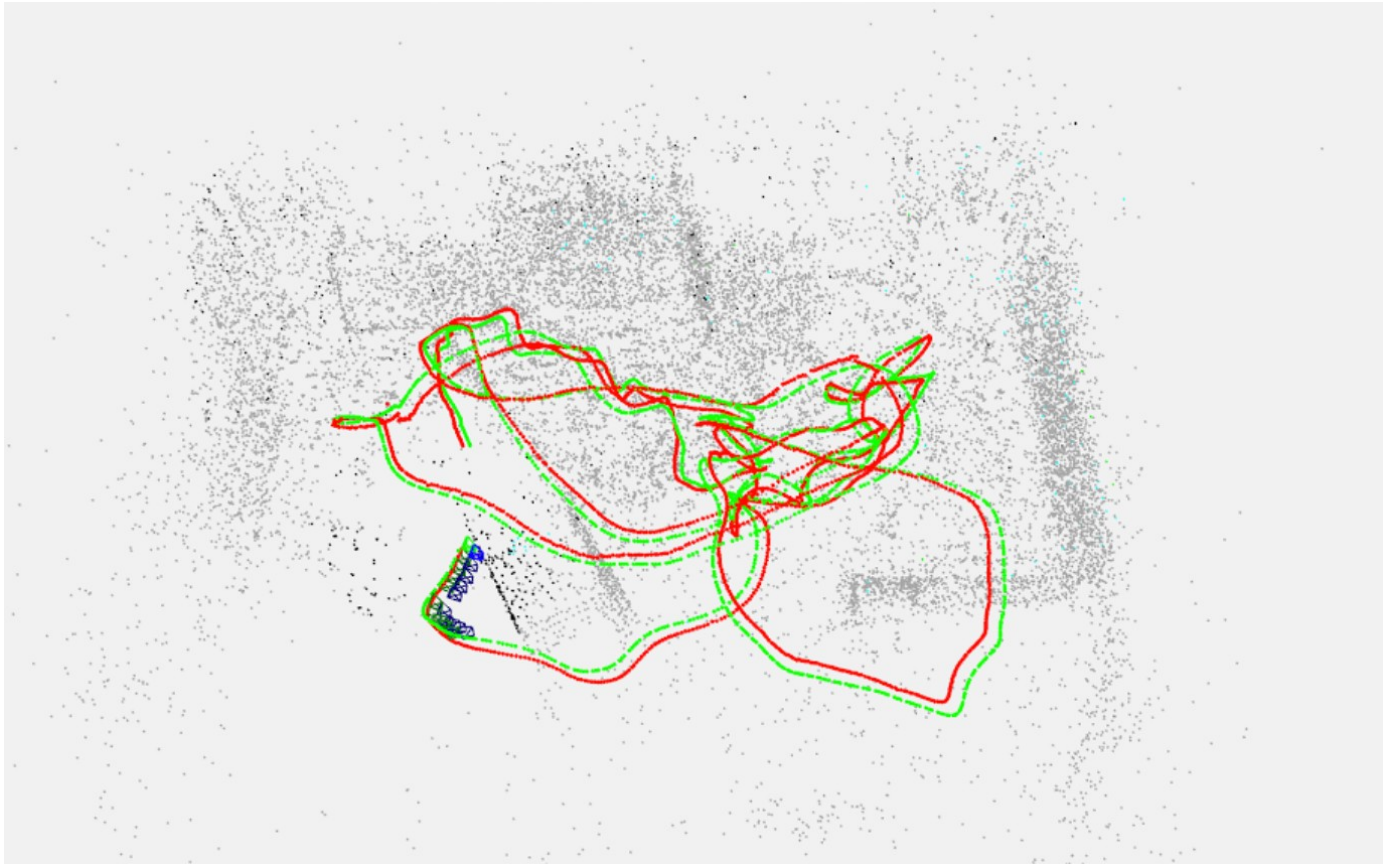
## 2.2 Execution time comparison



# 2 Result

## 2.3 Visualization comparison

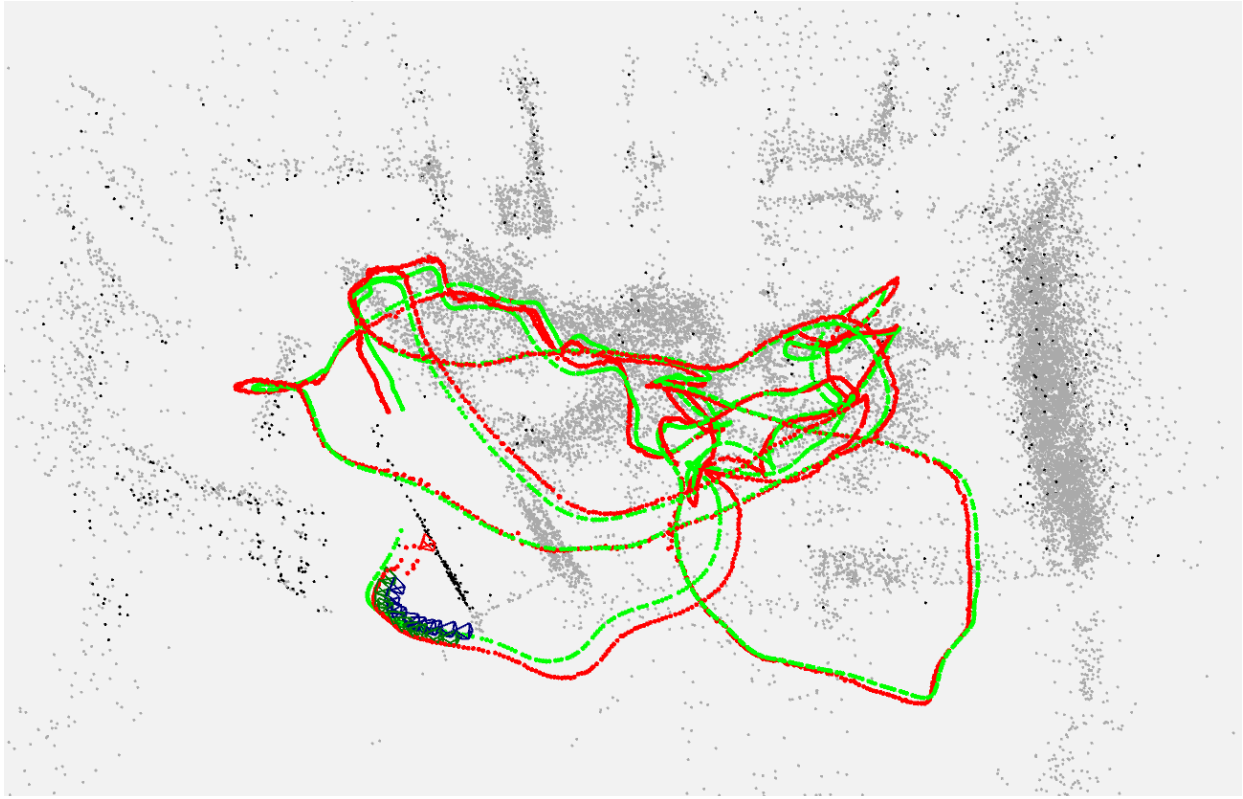
Maximum empty cells percentage: 0.42



# 2 Result

## 2.3 Visualization comparison

Exercise 5

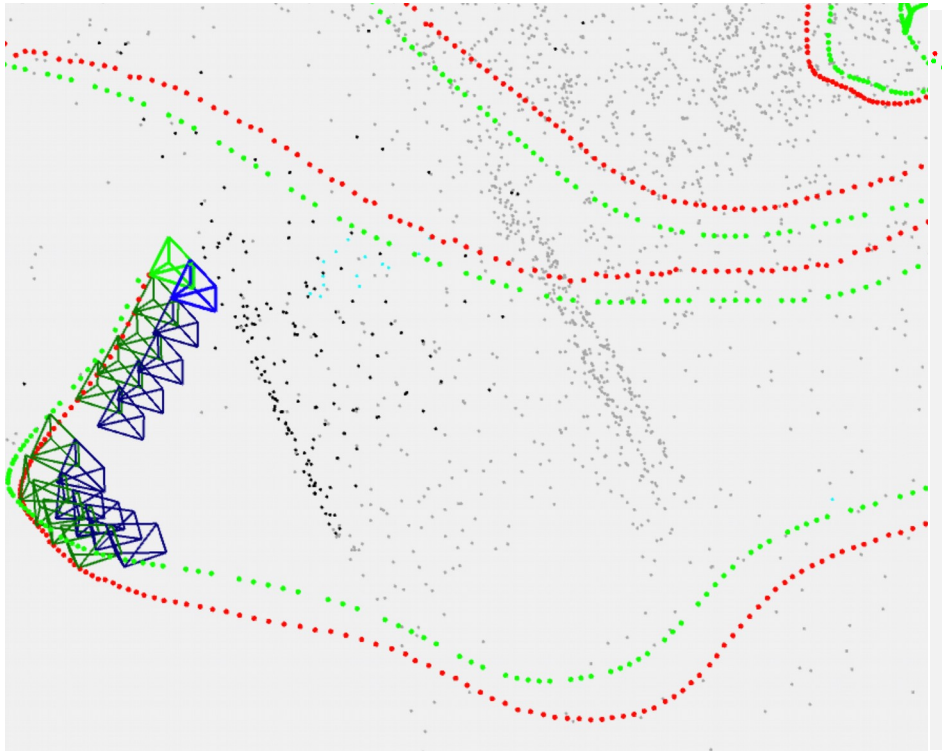


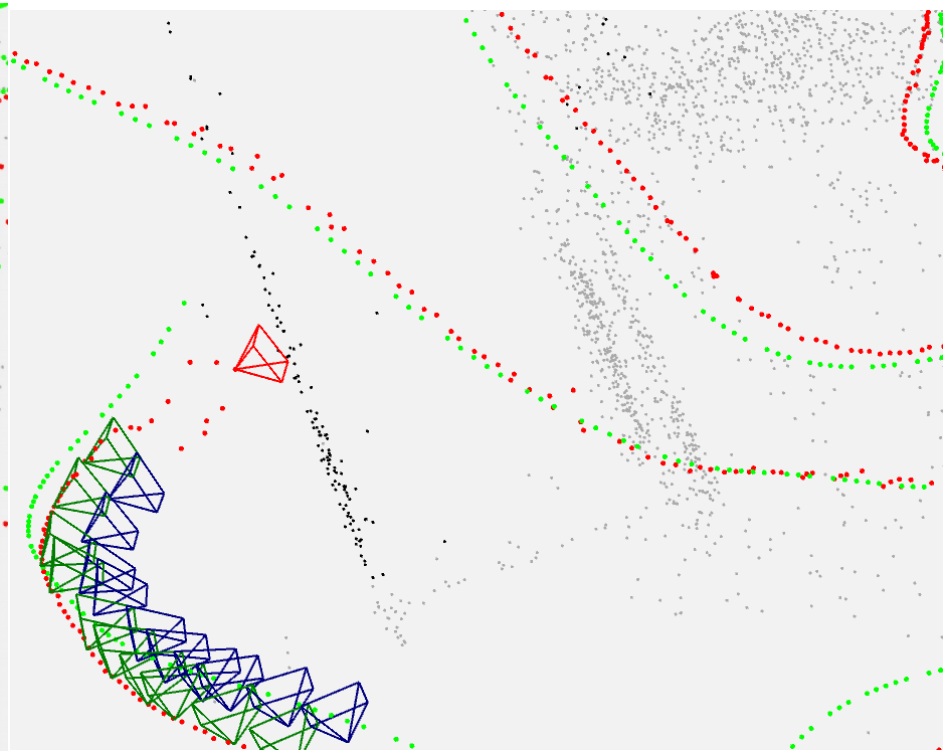
# 2 Result

## 2.3 Visualization comparison

Maximum empty cells number: 0.42

Exercise 5

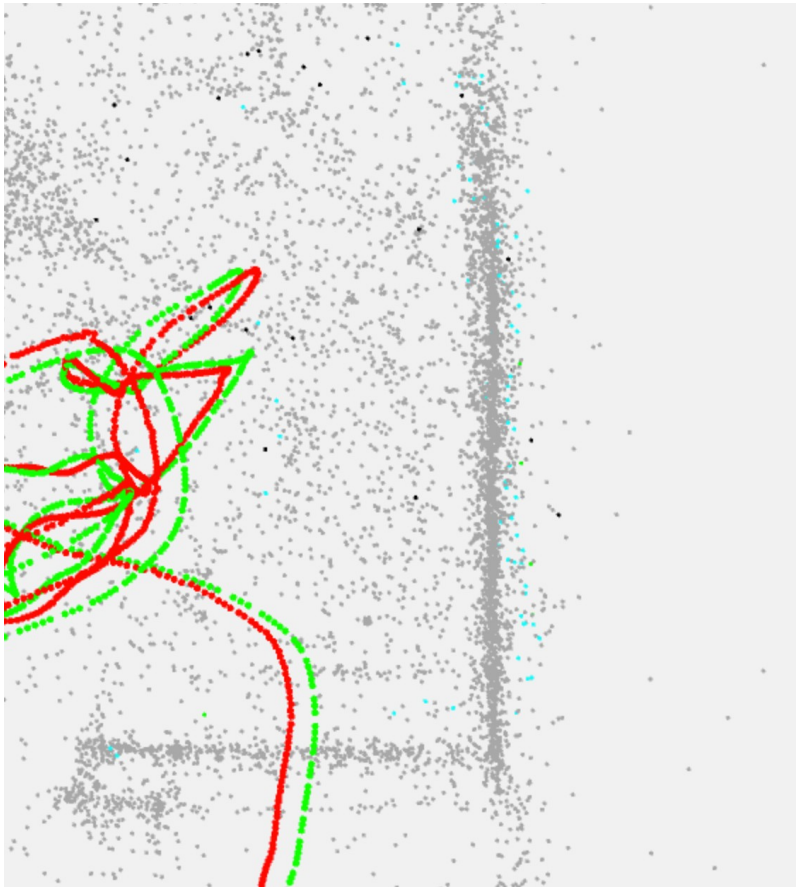




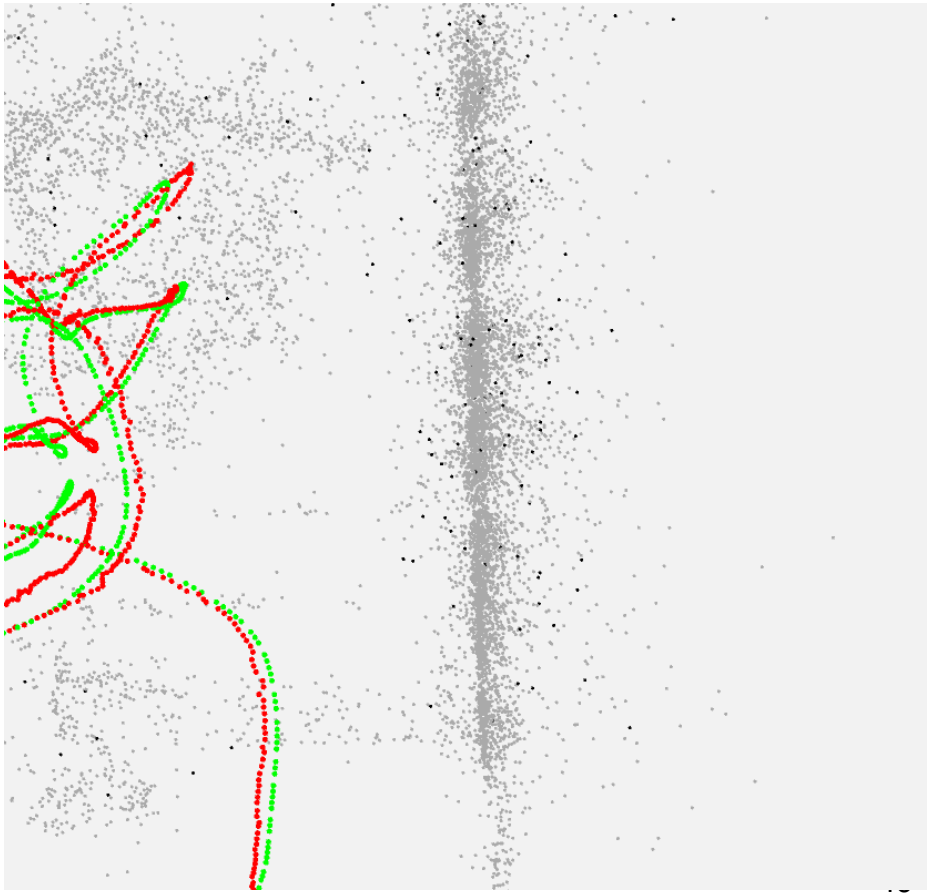
# 2 Result

## 2.3 Visualization comparison

Maximum empty cells number: 0.42



Exercise 5



# 3 Conclusion

What we did in this project:

1. We use Lucas and Kanade Optical Flow method to replace normal feature matching in slam.
2. We try different key framing strategies: use grid or inliers.
3. We try different setups for the pipeline, such as shape of grid, key points detection strategies.
4. We visualize the ground truth and estimated trajectories of camera and do analysis.

# 3 Conclusion

What we can do in the future:

1. implement optical flow manually, using intensity invariant optical flow to make the pipeline more stable.

2. try to implement frame to frame triangulation: using left images at different time step to triangulate new landmarks.

# Q&A

Any question?

Thank you for your attention!