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

1.2 Project Pipeline

1.3 Parameter and Strategy

2. Evaluation

2.1 Precision Comparison

2.2 Execution Time Comparison

2.3 Visualization Comparison

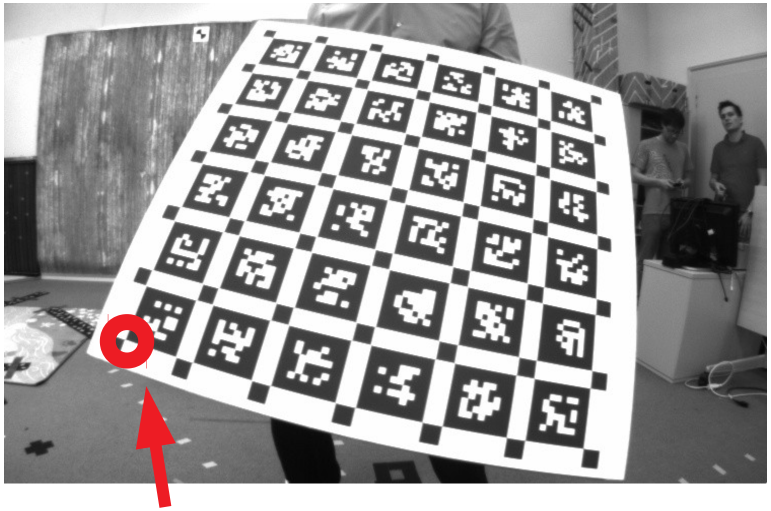
3. Conclusion

1 Introduction

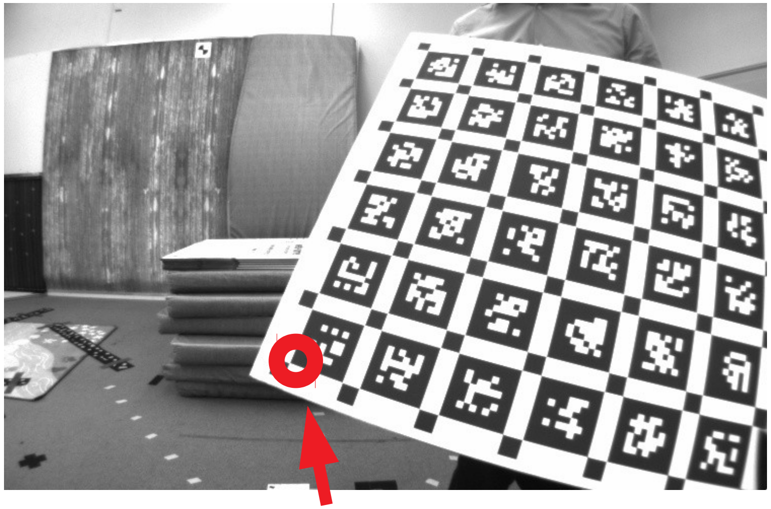
1.1 Lukas Kanade method for optical flow

Input: two images, key points in one image

Output: the key points in the other image



Input key point position in one image

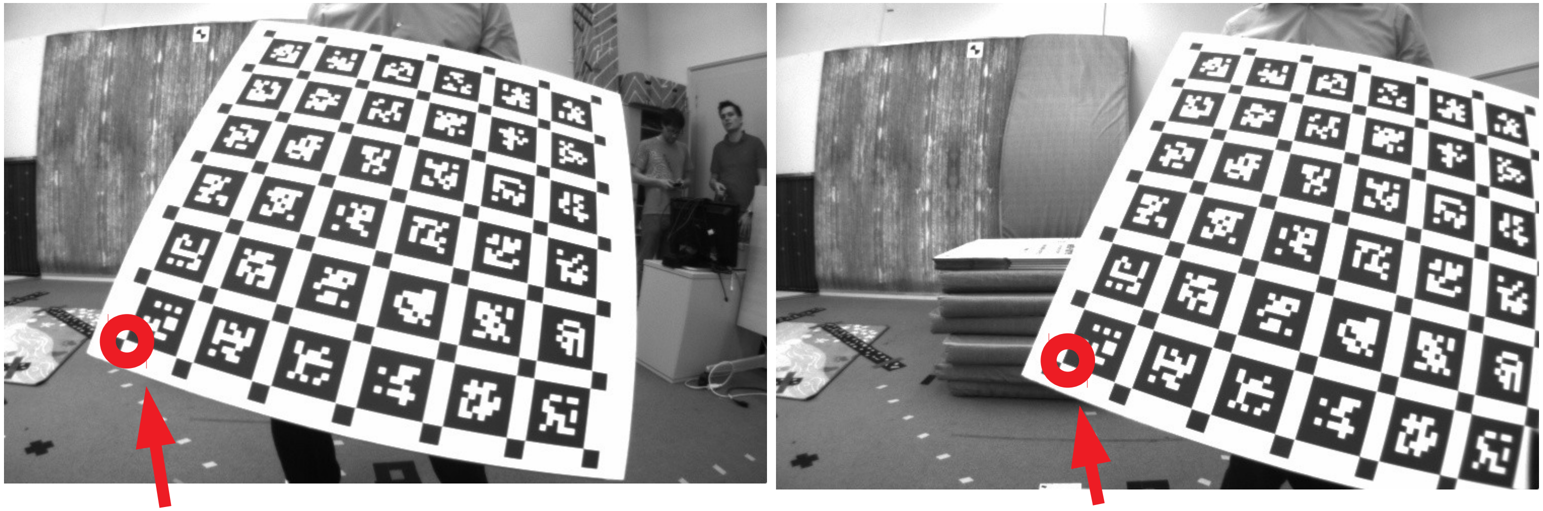


Output key point position in the other image

1 Introduction

1.1 Lukas Kanade method for optical flow

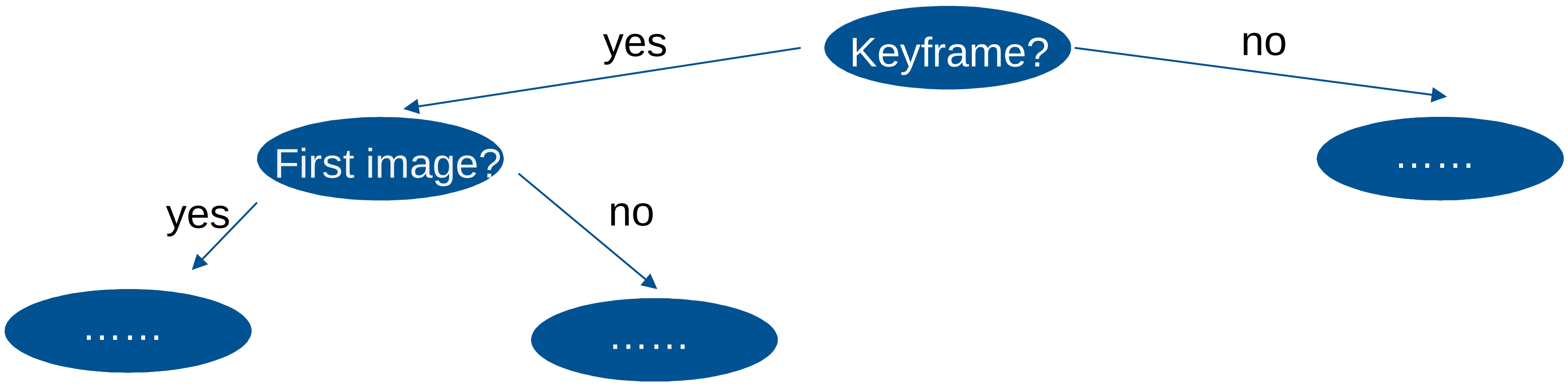
Inspiration in SLAM:
replace feature descriptors matching with optical flow method!



Input key point position in one image

Output key point position in the other image

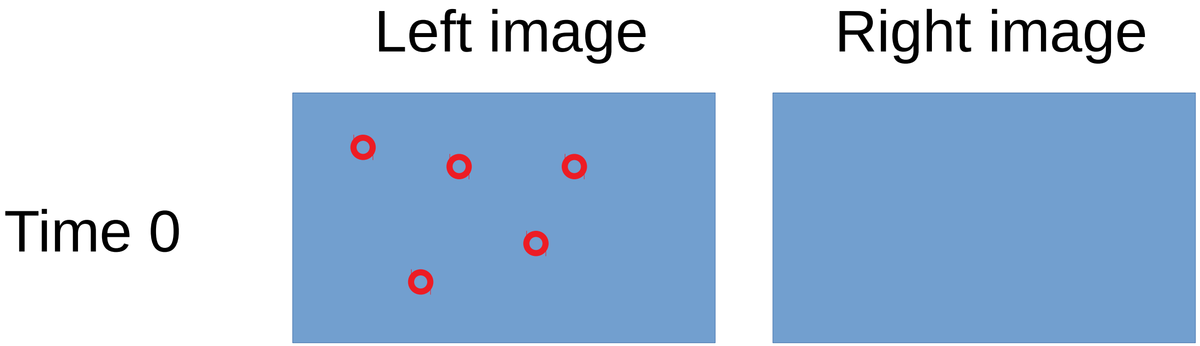
1 Introduction



1 Introduction

1.2 Project pipeline: first key frame

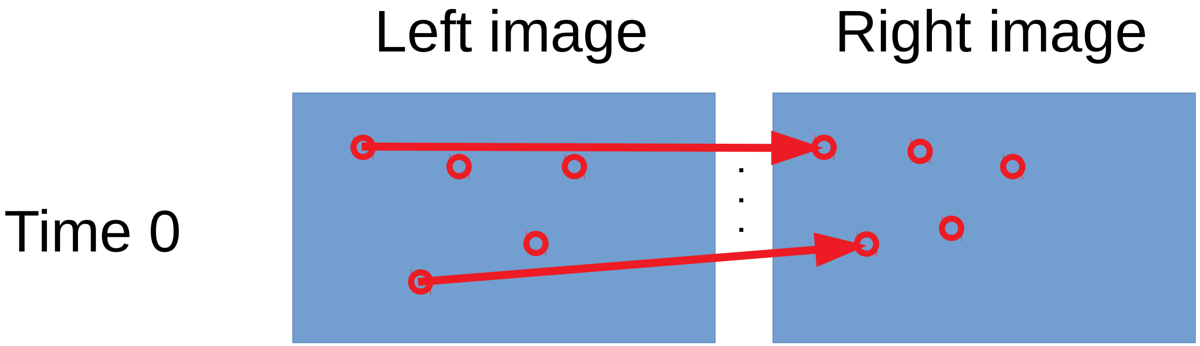
Step 1: detect key points in left image



1 Introduction

1.2 Project pipeline: first key frame

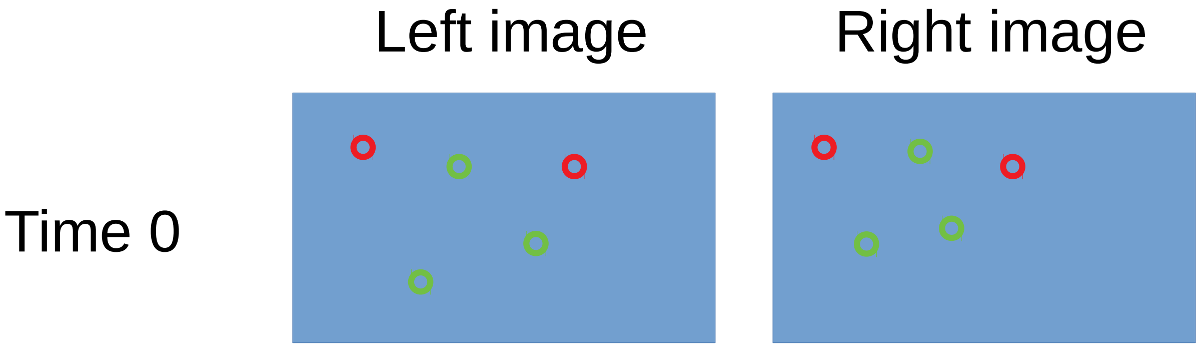
Step 2: optical flow to right image



1 Introduction

1.2 Project pipeline: first key frame

Step 3: check epipolar constraint and find inlier



Left image

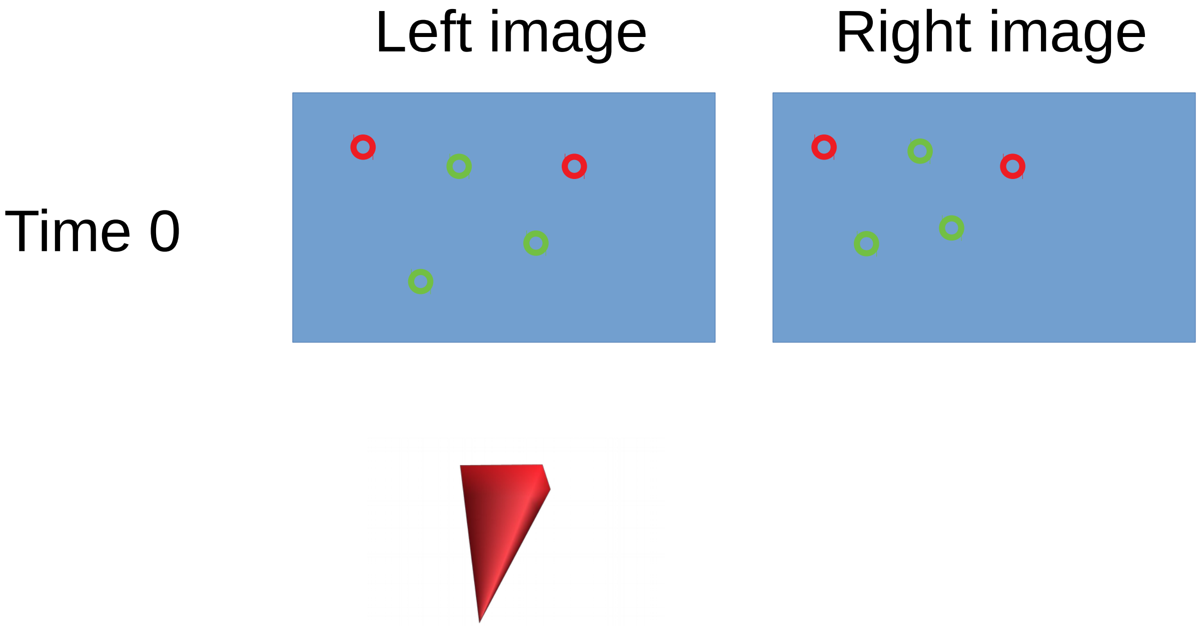
Right image

Time 0

1 Introduction

1.2 Project pipeline: first key frame

Step 4: localize camera



Left image

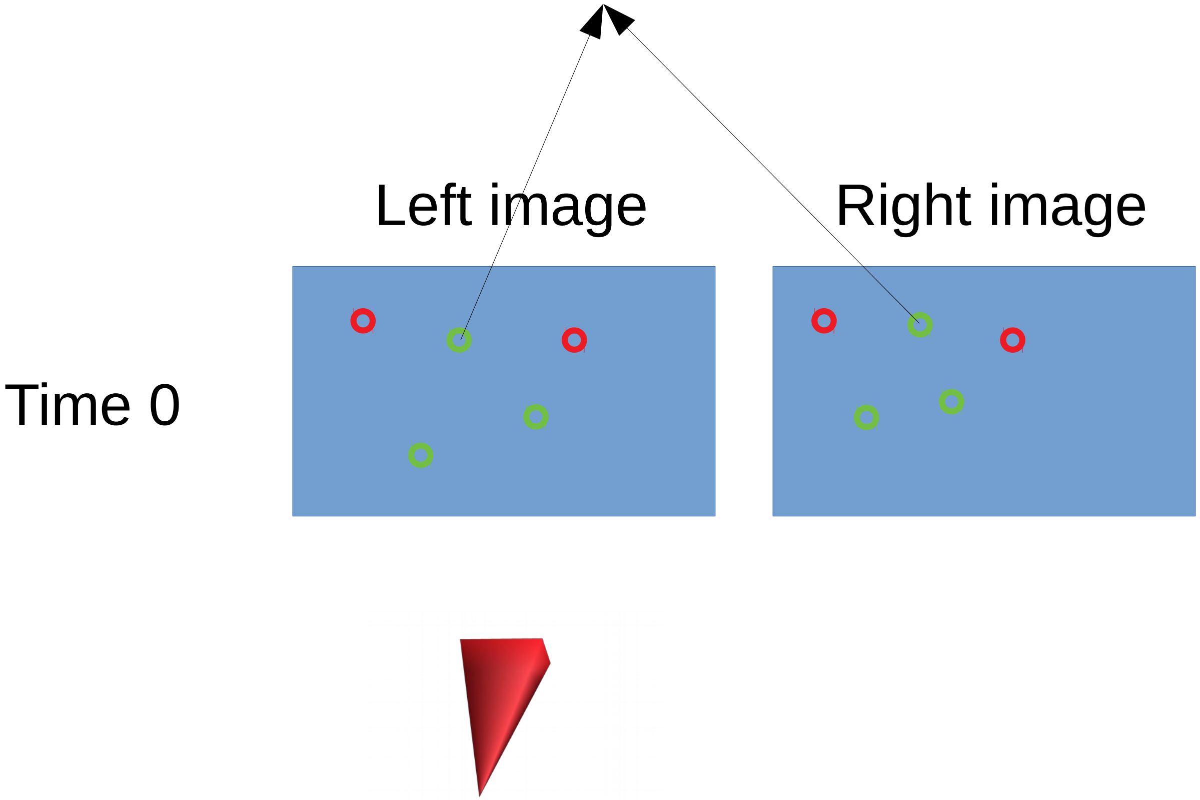
Right image

Time 0

1 Introduction

1.2 Project pipeline: first key frame

Step 5: triangulate landmarks



Left image

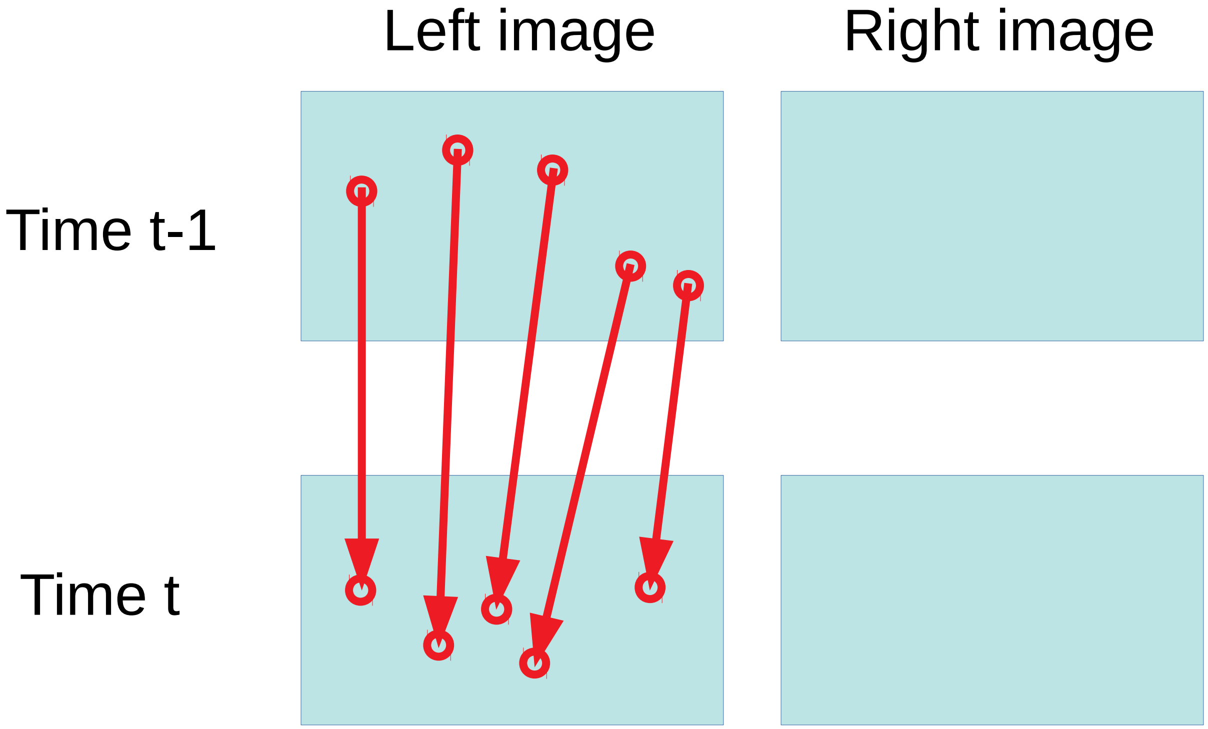
Right image

Time 0

1 Introduction

1.2 Project pipeline: second and later key frame

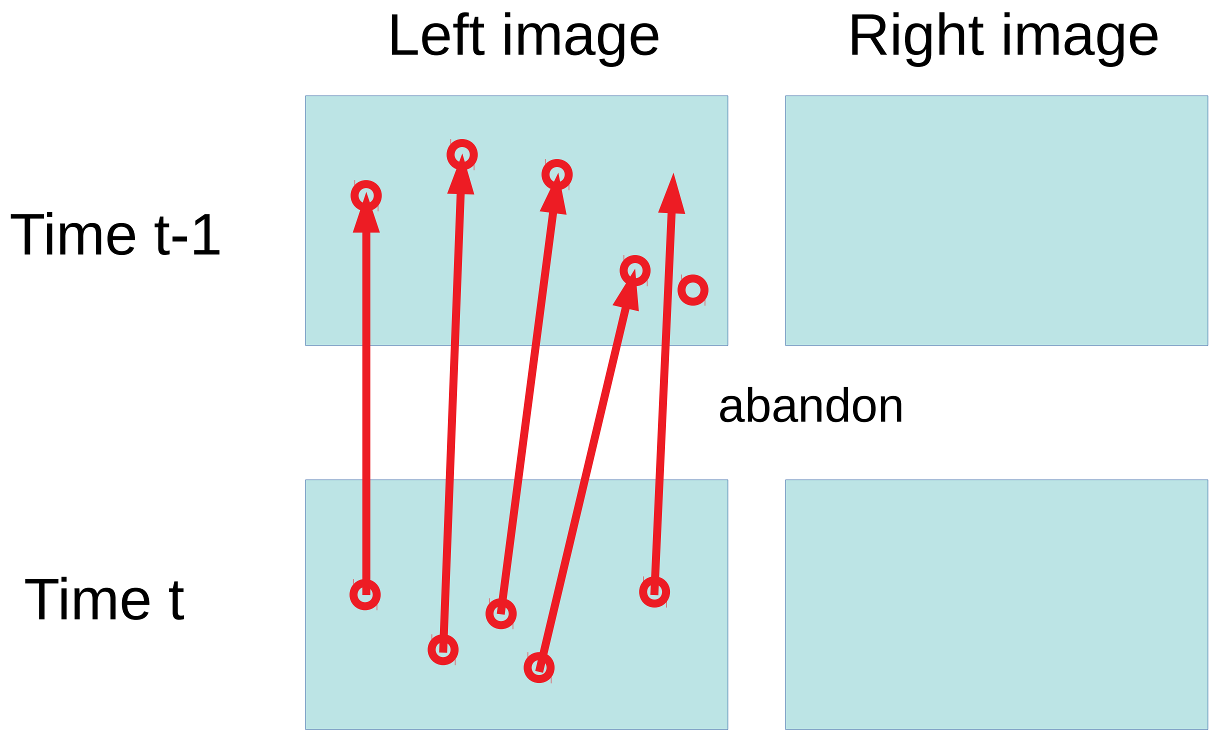
Step 1: optical flow to next left frame



1 Introduction

1.2 Project pipeline: second and later key frame

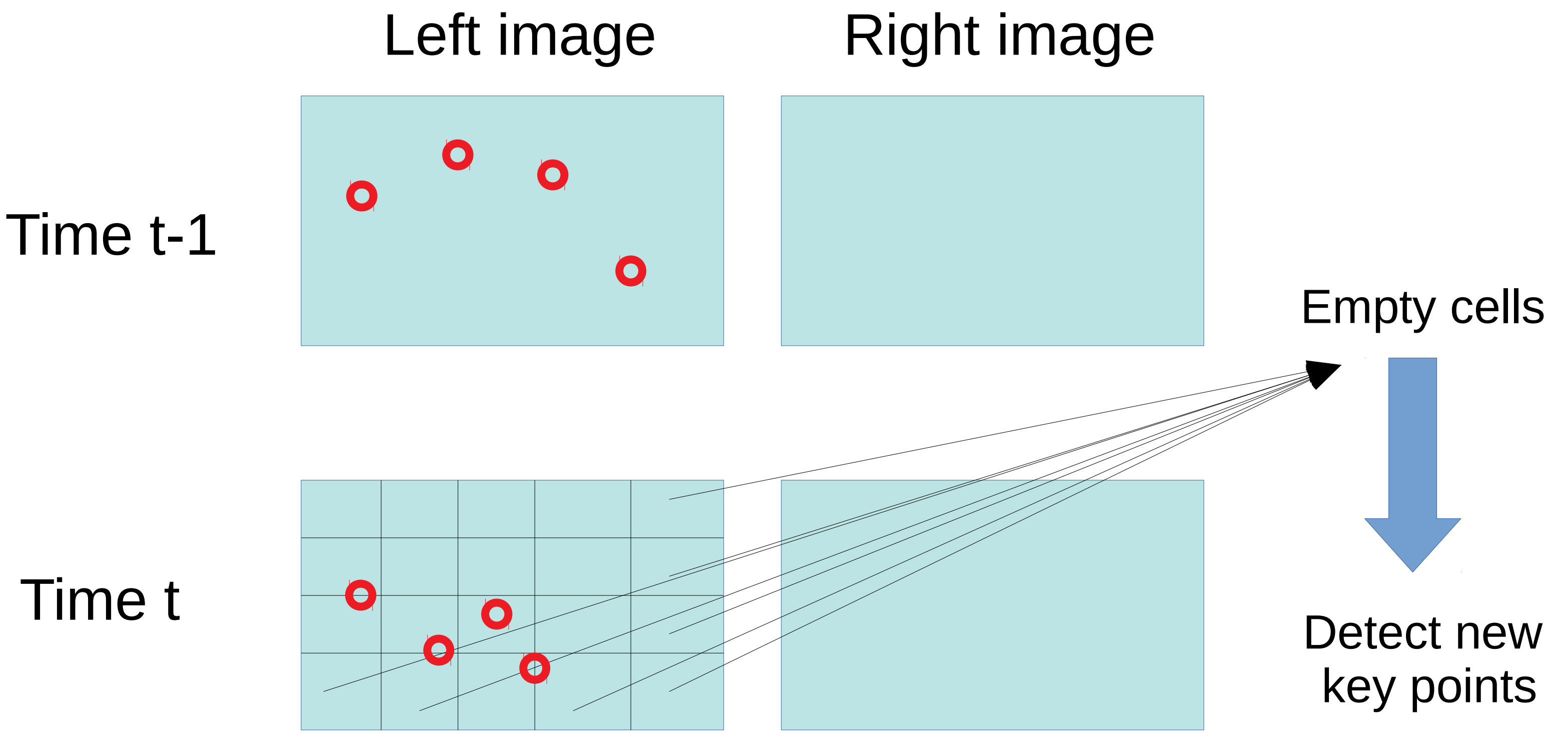
Step 1: optical flow: backward check



1 Introduction

1.2 Project pipeline: second and later key frame

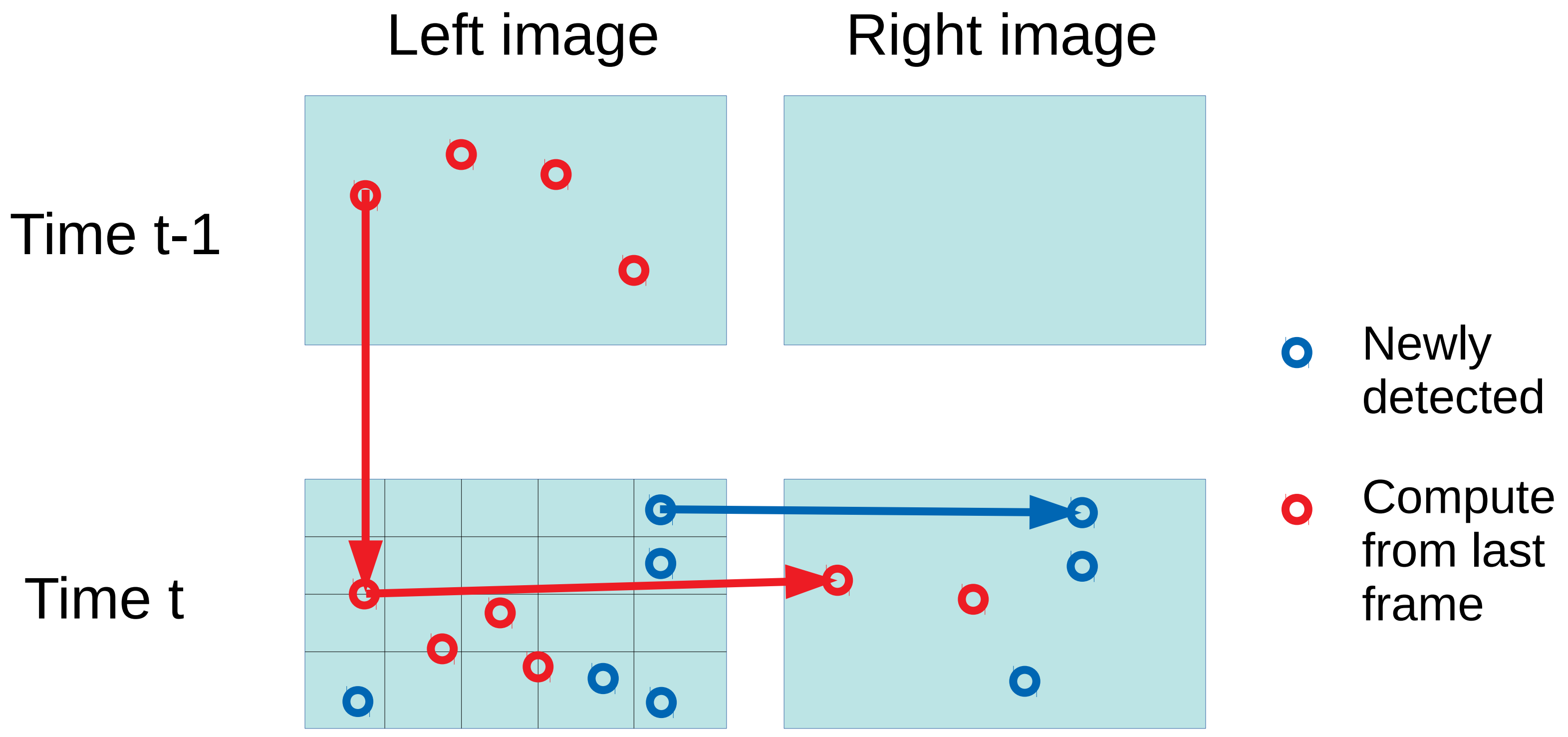
Step 2: make grid & detect key points in empty cells



1 Introduction

1.2 Project pipeline: second and later key frame

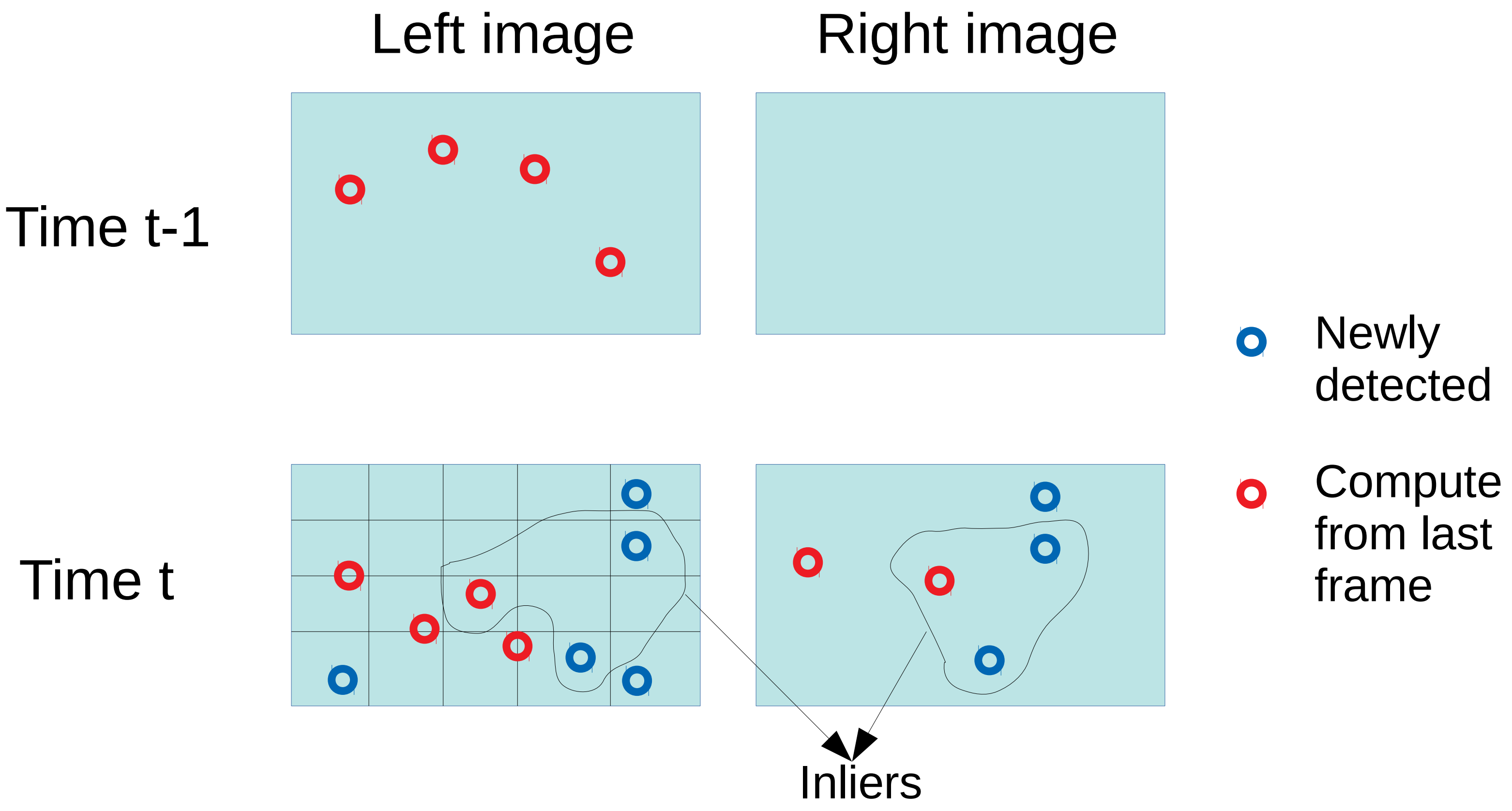
Step 3: optical flow to the right image



1 Introduction

1.2 Project pipeline: second and later key frame

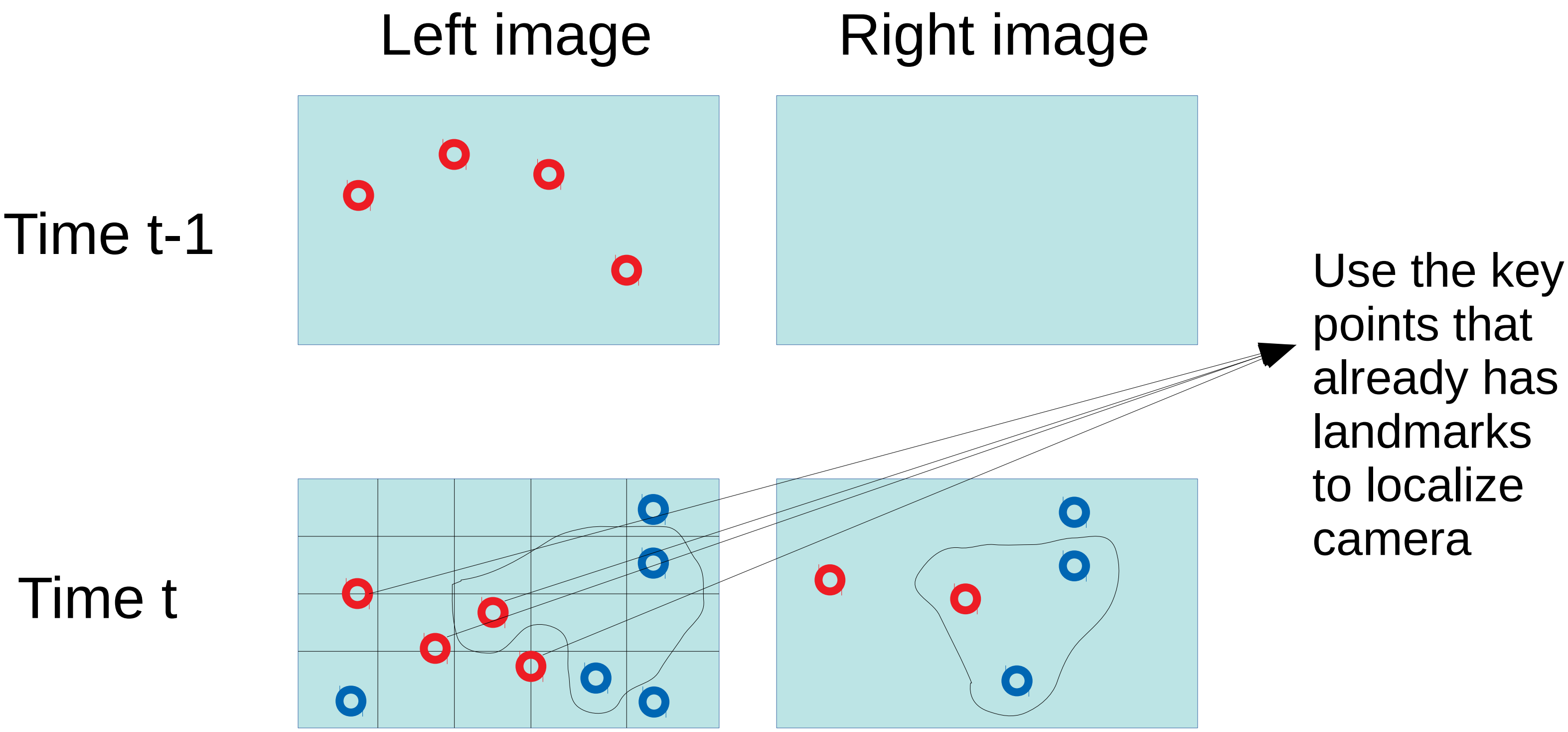
Step 4: use epipolar constraint to find inliers



1 Introduction

1.2 Project pipeline: second and later key frame

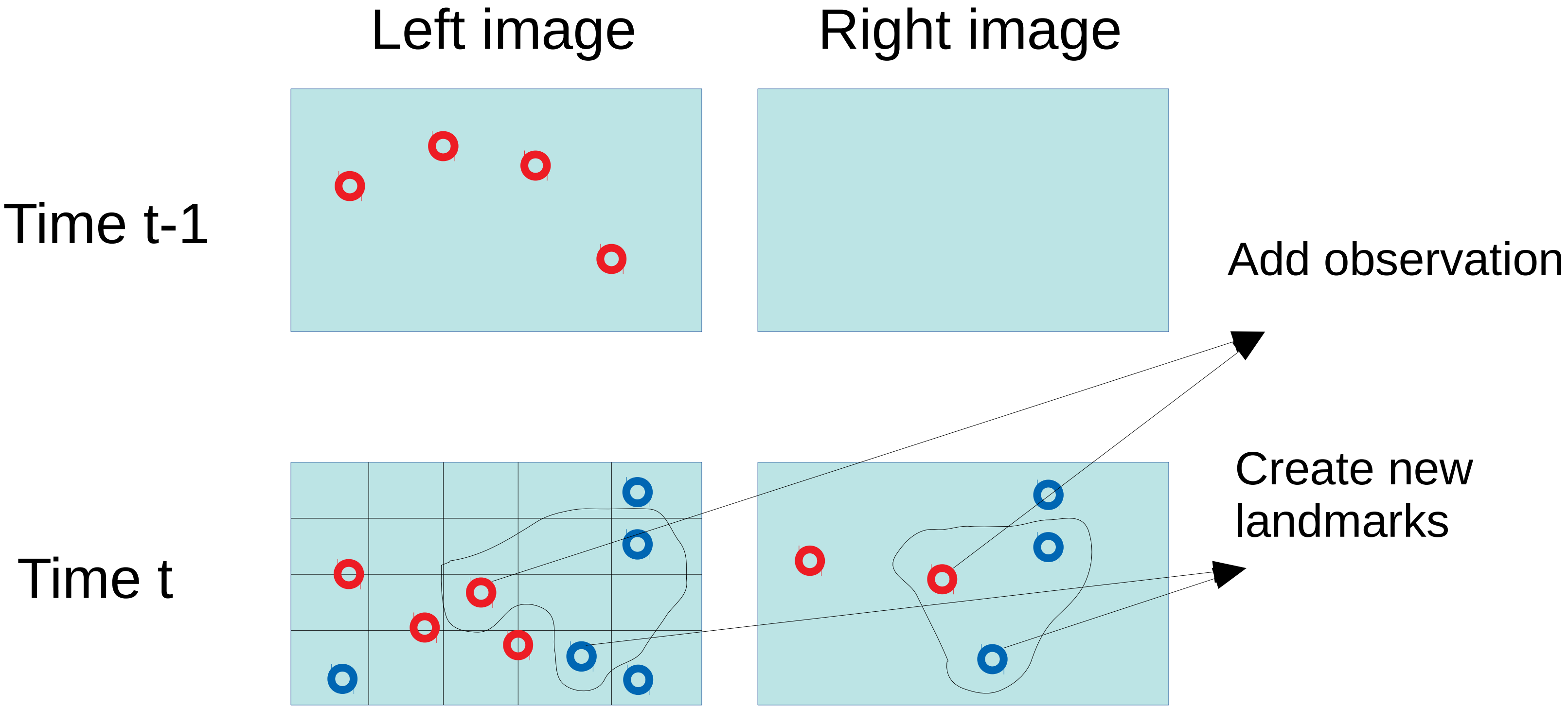
Step 5: localize camera


Left image
Right image
Time t-1
Time t
Use the key points that already has landmarks to localize camera

1 Introduction

1.2 Project pipeline: second and later key frame

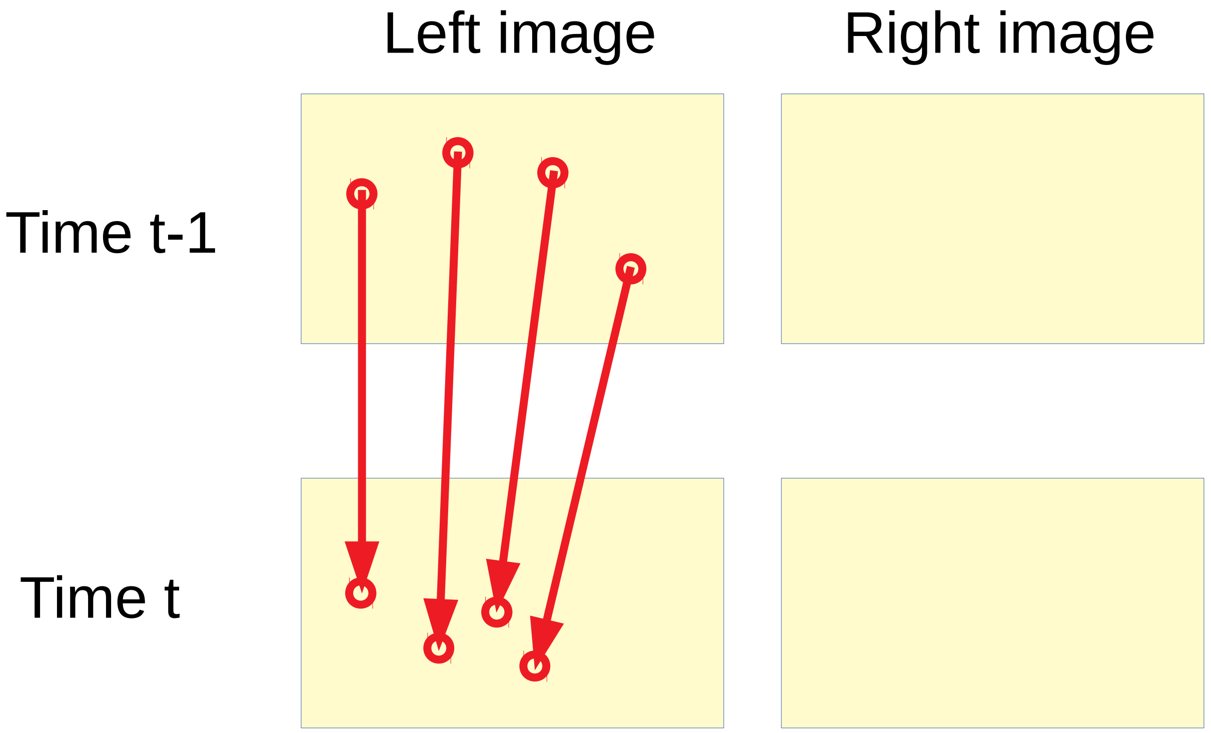
Step 6: add observation and create landmarks



1 Introduction

1.2 Project pipeline: non key frame

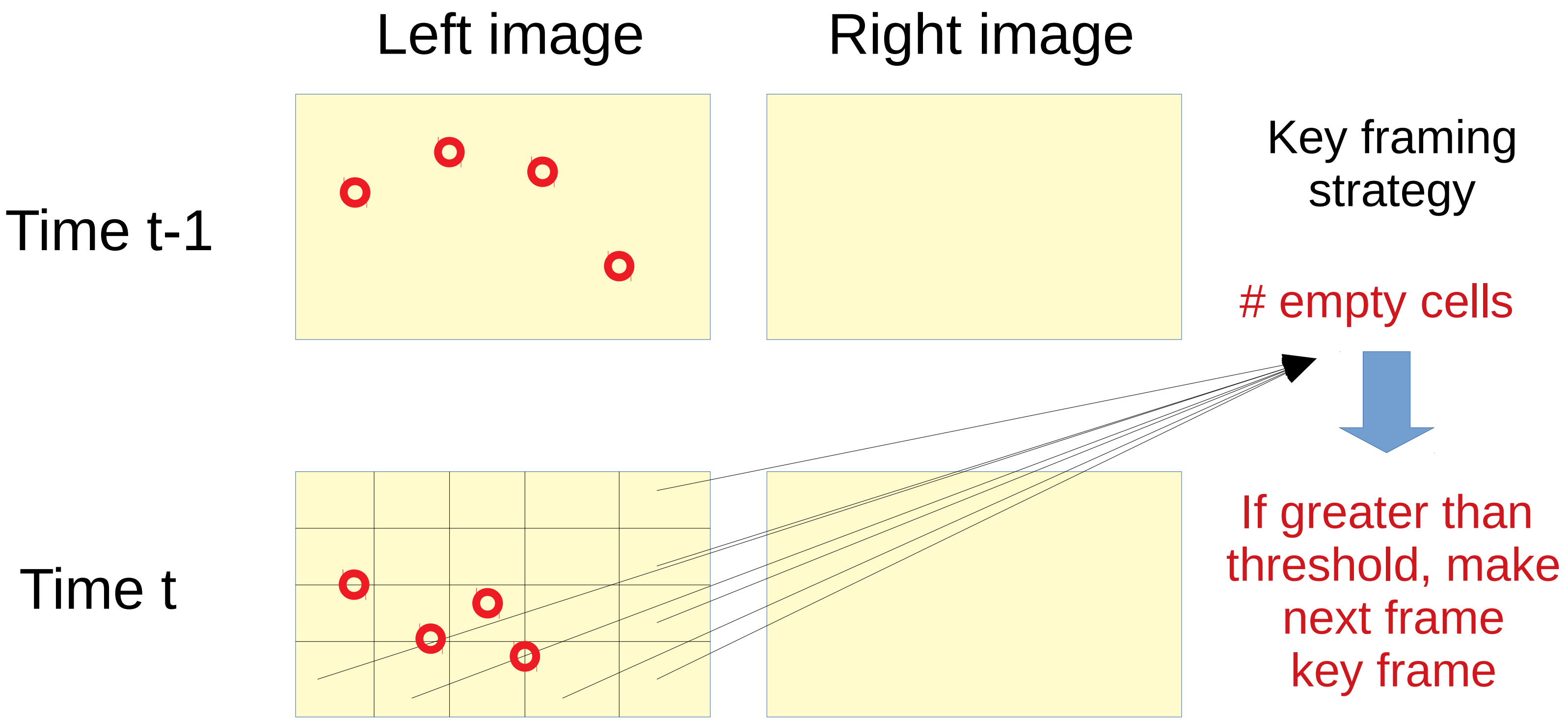
Step 1: optical flow to next frame



1 Introduction

1.2 Project pipeline: non key frame

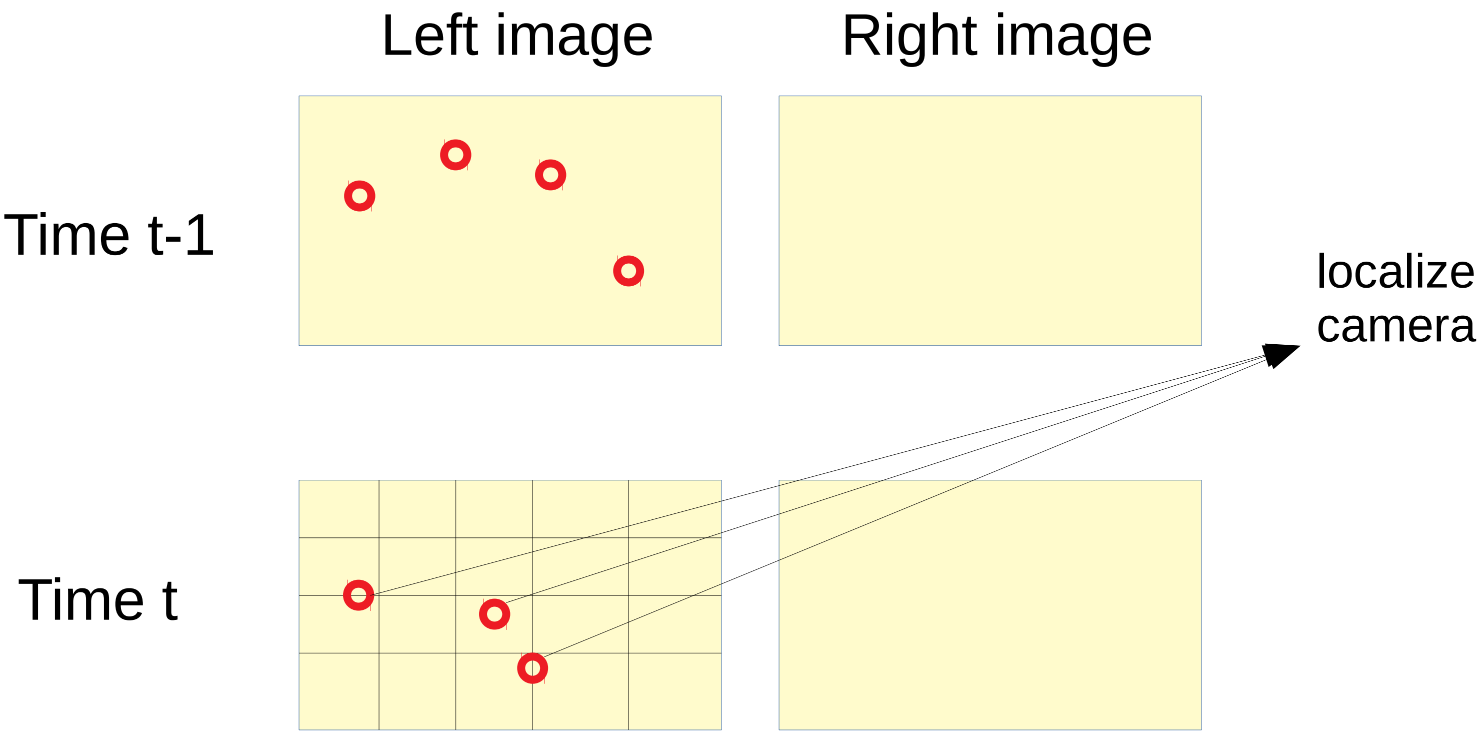
Step 2: make grid & count empty cells



1 Introduction

1.2 Project pipeline: non key frame

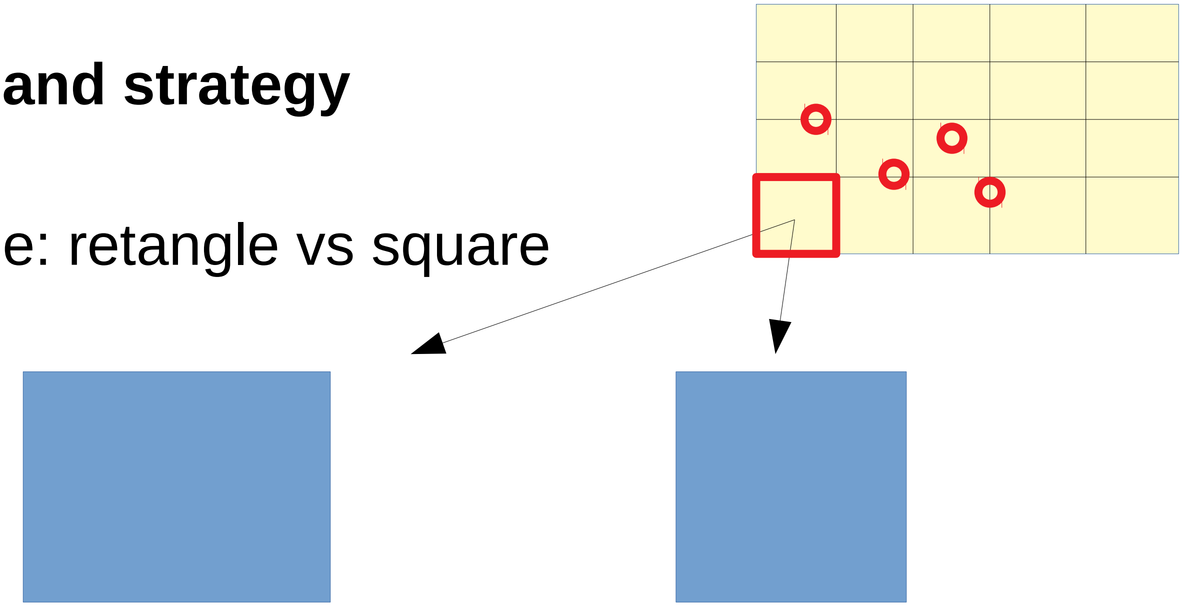
Step 3: localize camera



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

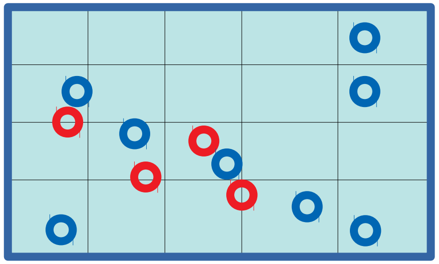
1 Introduction

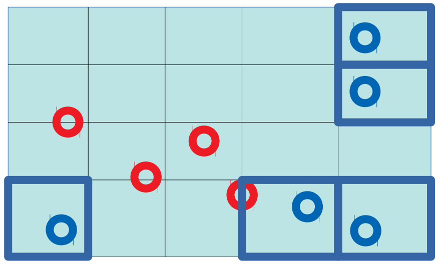
1.3 Parameter and strategy

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

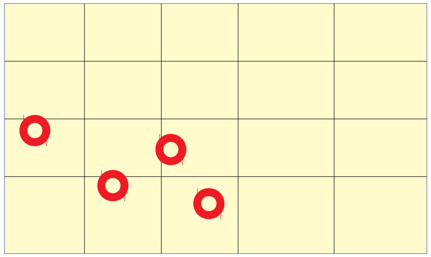
1 Introduction

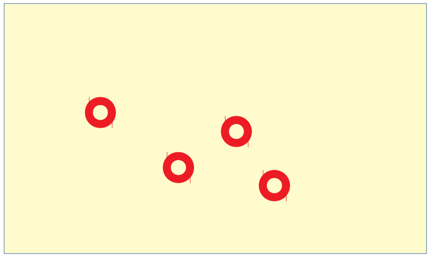
1.3 Parameter and strategy

1.3.3 Key framing: use grid vs use only inliers number

# empty cells

If greater than threshold, make next frame key frame





# localization inliers

If smaller than threshold, make next frame key frame

rmse	0.11	0.43

2 Evaluation

2 Evaluation

2.1 Precision comparison

Basic parameters

image height (pixel)	image width (pixel)	grid size (pixel x pixel)
480	752	32 x 32

Strategy

1. Square grid

2. Key points detection only in empty cell

3. Use number of empty cells to decide key frame

2 Evaluation

2.1 Precision comparison

Variation parameters: maximum empty cells percentage*

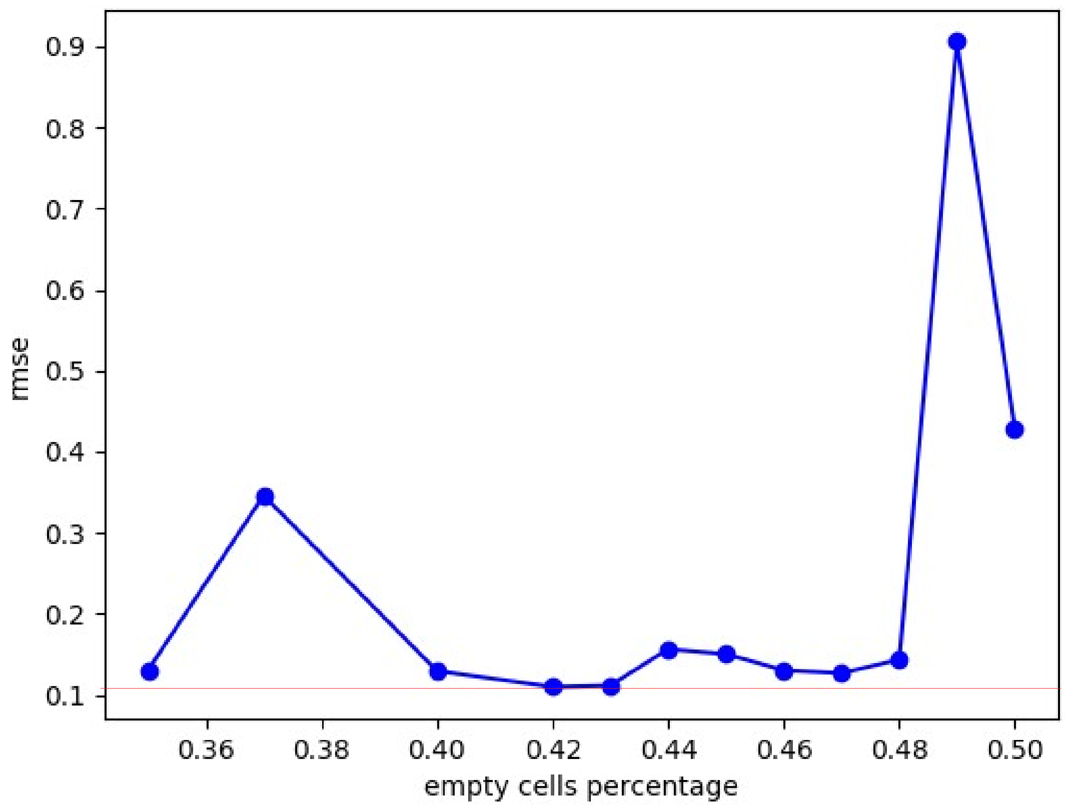
Max empty cells	35%	40%	42%	44%	46%	Exercise 5
rmse	0.1303	0.1295	**0.1099**	0.1563	0.1304	**0.1012**

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

2 Evaluation

2.1 Precision comparison

empty cells percentage vs rmse



2 Evaluation

2.2 Execution time comparison

2.2.1 Detection time

Max empty cells	35%	40%	42%	44%	46%	Exercise 5
detection time (seconds)	3.03643	2.73232	2.43286	2.62328	2.26326	**44.0214**

2 Evaluation

2.2 Execution time comparison

2.2.2 Optimization time

Max empty cells (%)	0.42	Exercise 5
Opt. time(s)	305.043	**65.8619**
# landmarks	290657	**221886**
# observation	1939460	**559563**
# keyframe	414	**170**

2 Evaluation

2.2 Execution time comparison

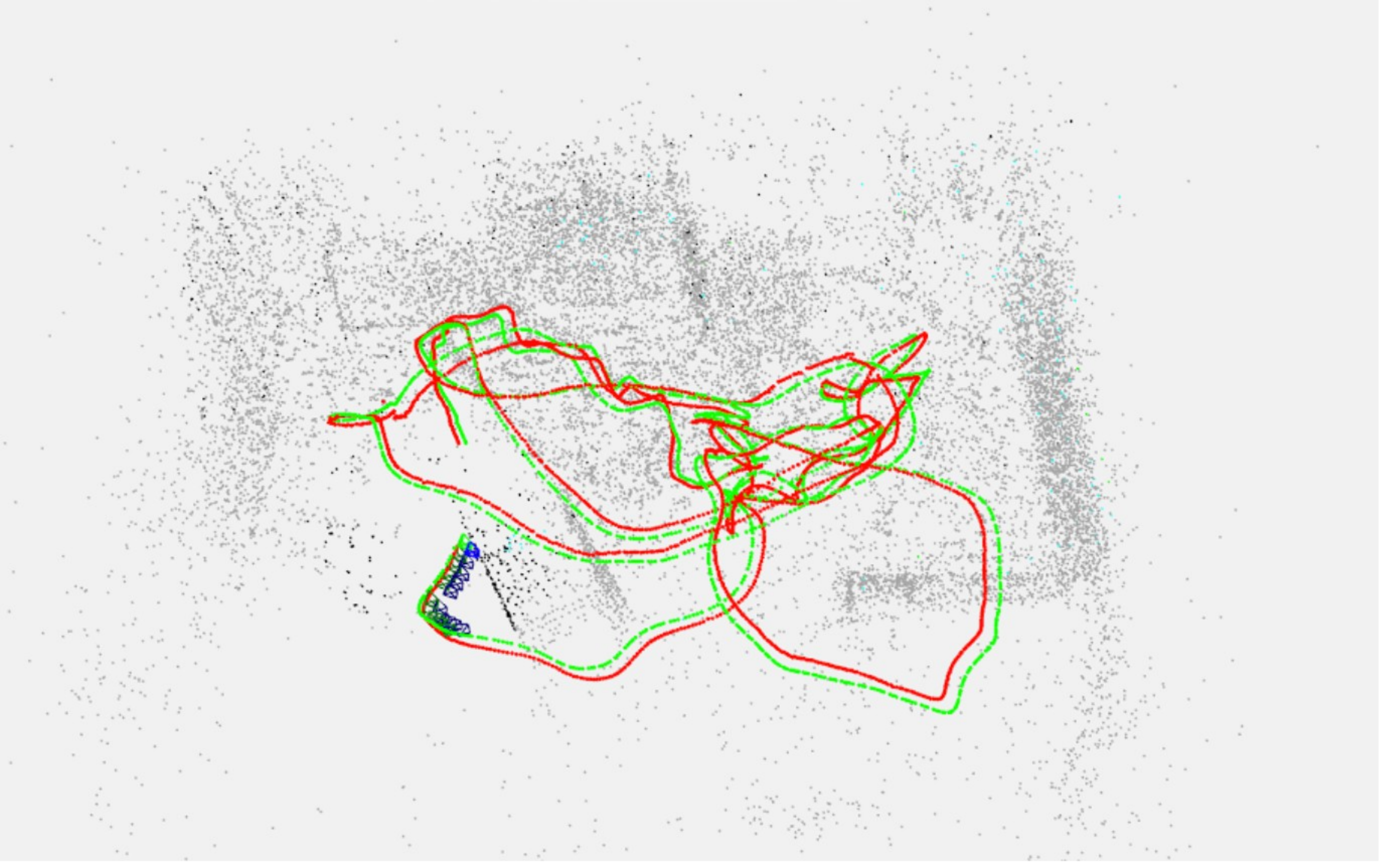
2.2.3 Frame to frame (f2f) optical flow time

Ex. 5 Feature Matching time(s)	5.79
f2f optical flow time(s)	117.4~143.1

2 Evaluation

2.3 Visualization comparison

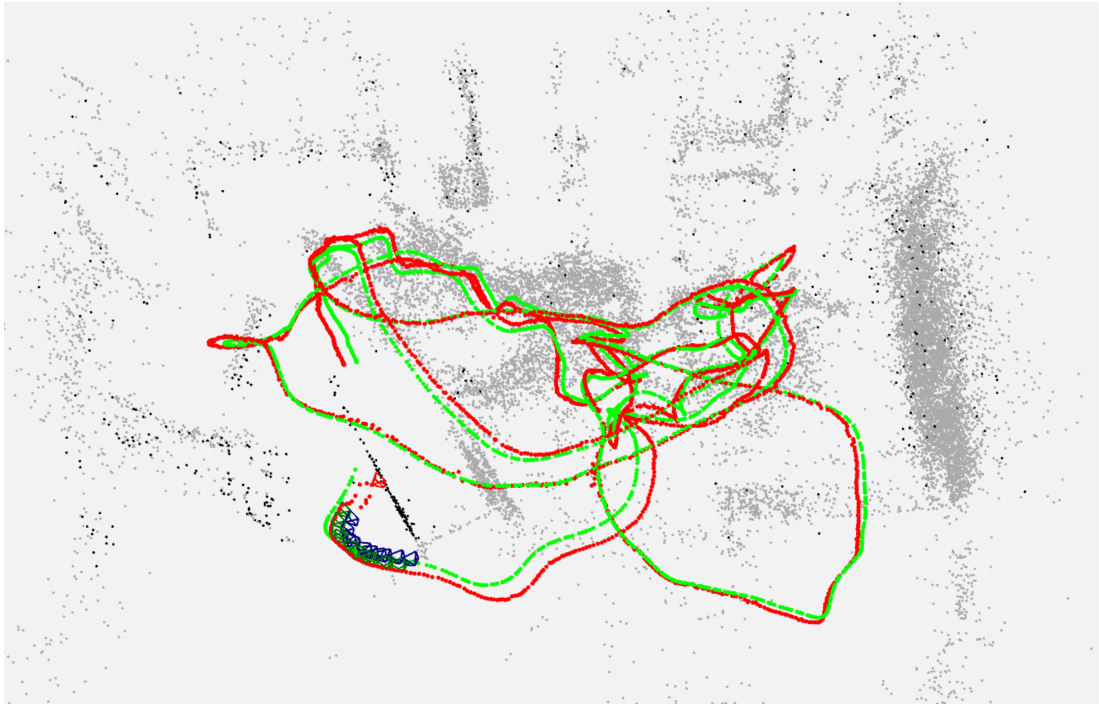
Maximum empty cells percentage: 42%



2 Evaluation

2.3 Visualization comparison

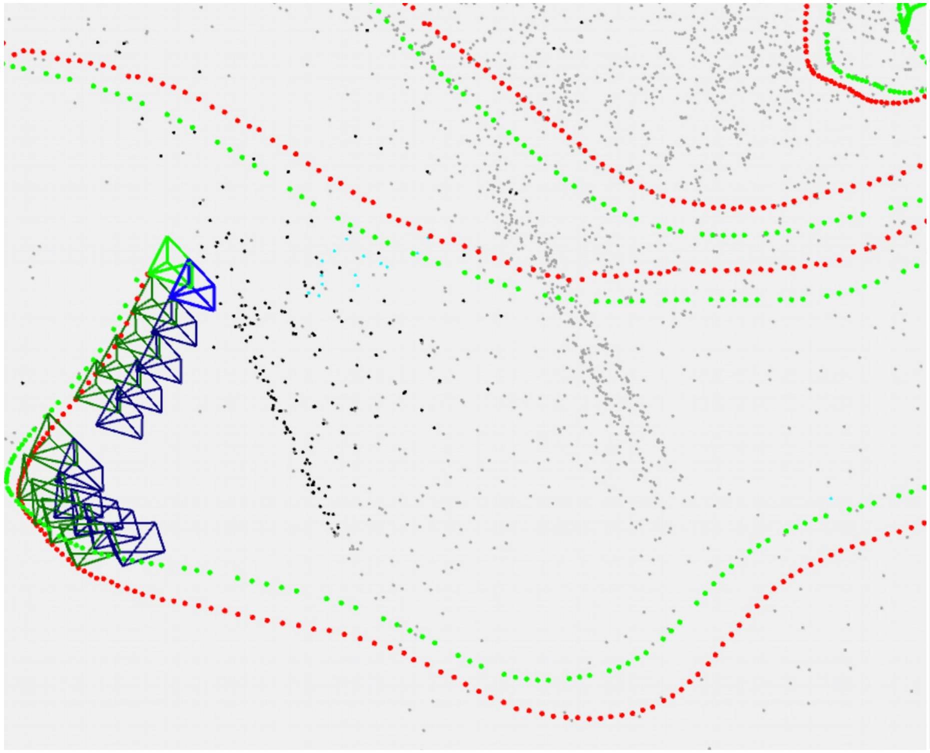
Exercise 5



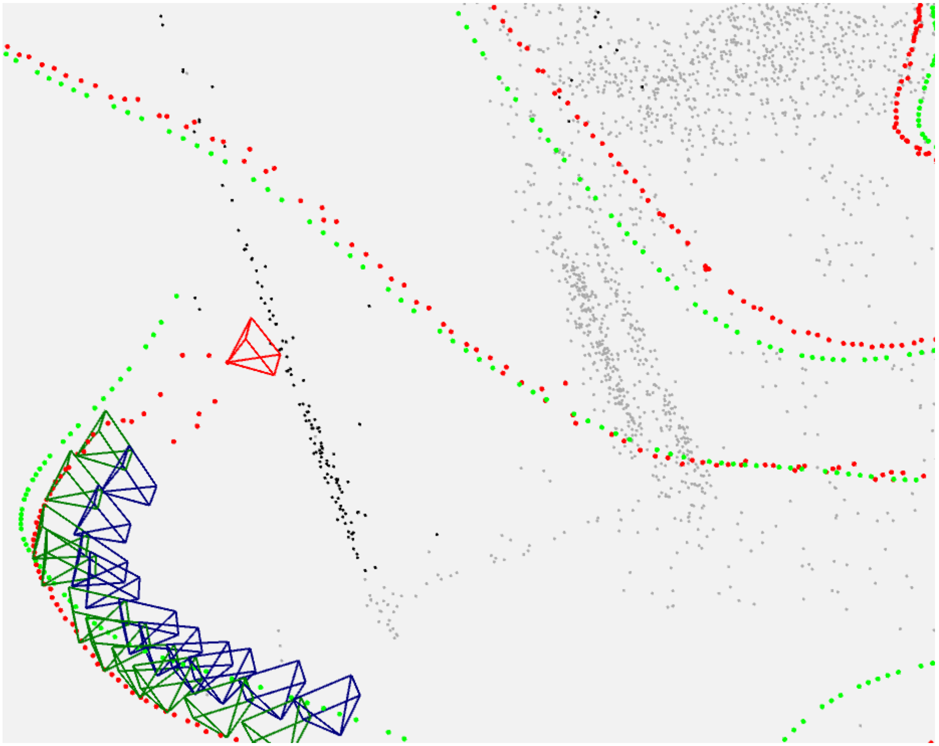
2 Evaluation

2.3 Visualization comparison

Maximum empty cells percentage: 42%



Exercise 5



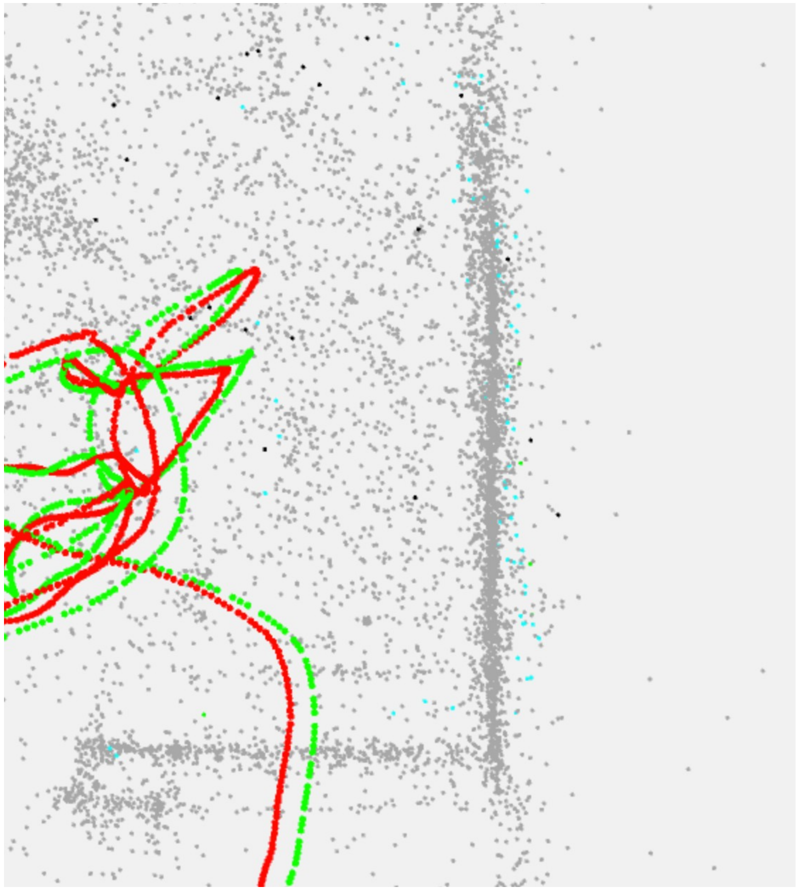
2 Evaluation

2.3 Visualization comparison

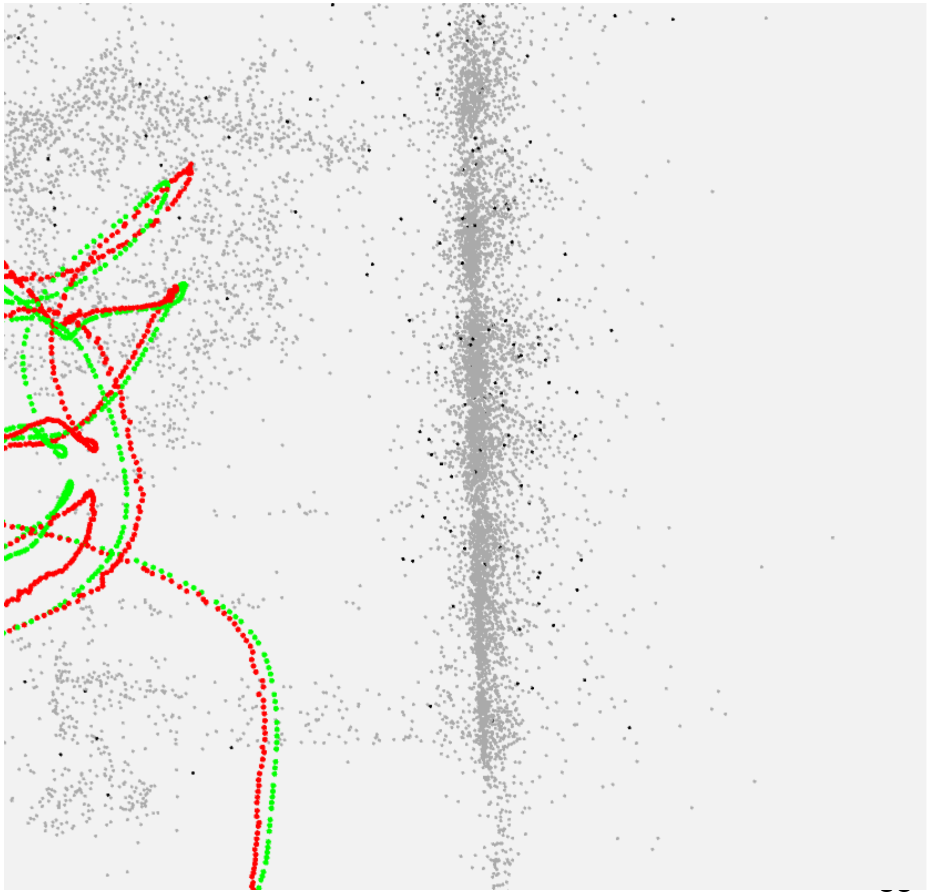
2 Evaluation

2.3 Visualization comparison

Maximum empty cells percentage: 42%



Exercise 5



3 Conclusion

1. Use Lucas and Kanade Optical Flow method to replace normal feature matching.
2. Try different key framing strategies: use grid or only inliers.
3. Try different configurations (e.g. shape of grid, key points detection strategies)
4. Visualize the ground truth and estimated trajectories of camera
5. Evaluation

Q&A

Any question?

Thank you for your attention!