Elective in Robotics

Monocular Visual Odometry

(based on Luca Ricci master thesis)

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Monocular vs. Stereo: examples from nature

predator

- predators' eyes face forward
- the field of view of each eye overlaps to create binocular vision (**stereo vision**)

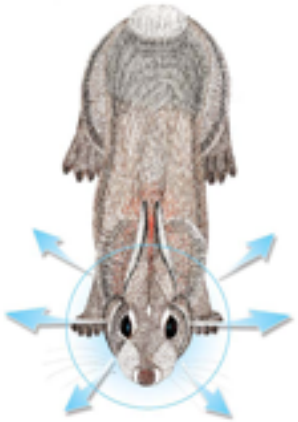
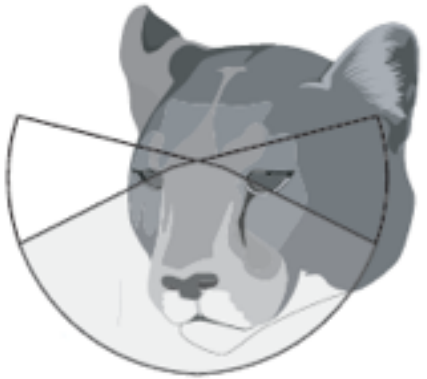
prey

- preys' eyes that sideways
- only a small overlap area between each eye field of view (**monocular vision**)

Monocular vs. Stereo: examples from nature

monocular vision features:

- increased field of view (FOV)
- limited depth perception

vision	monocular	stereoscopic
FOV	wide	narrow
eyes FOV overlapping	does not overlap 	overlapping 
estimation of distances	not accurate	accurate

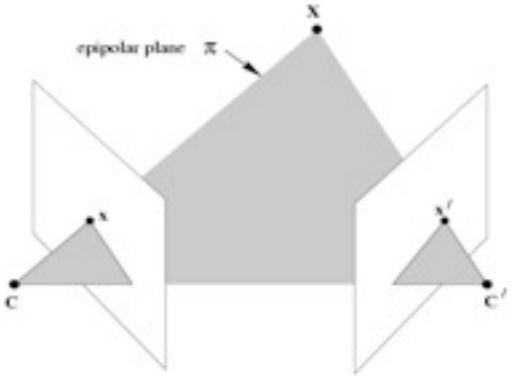
Monocular vs. Stereo: recovering depth

nature

pigeons head bobbing generates parallax motion for depth perception

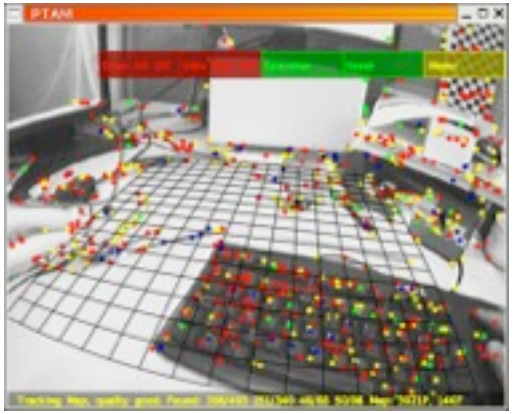
geometry

the geometric relationship between the projections of the same point in 2 different views allows depth computation

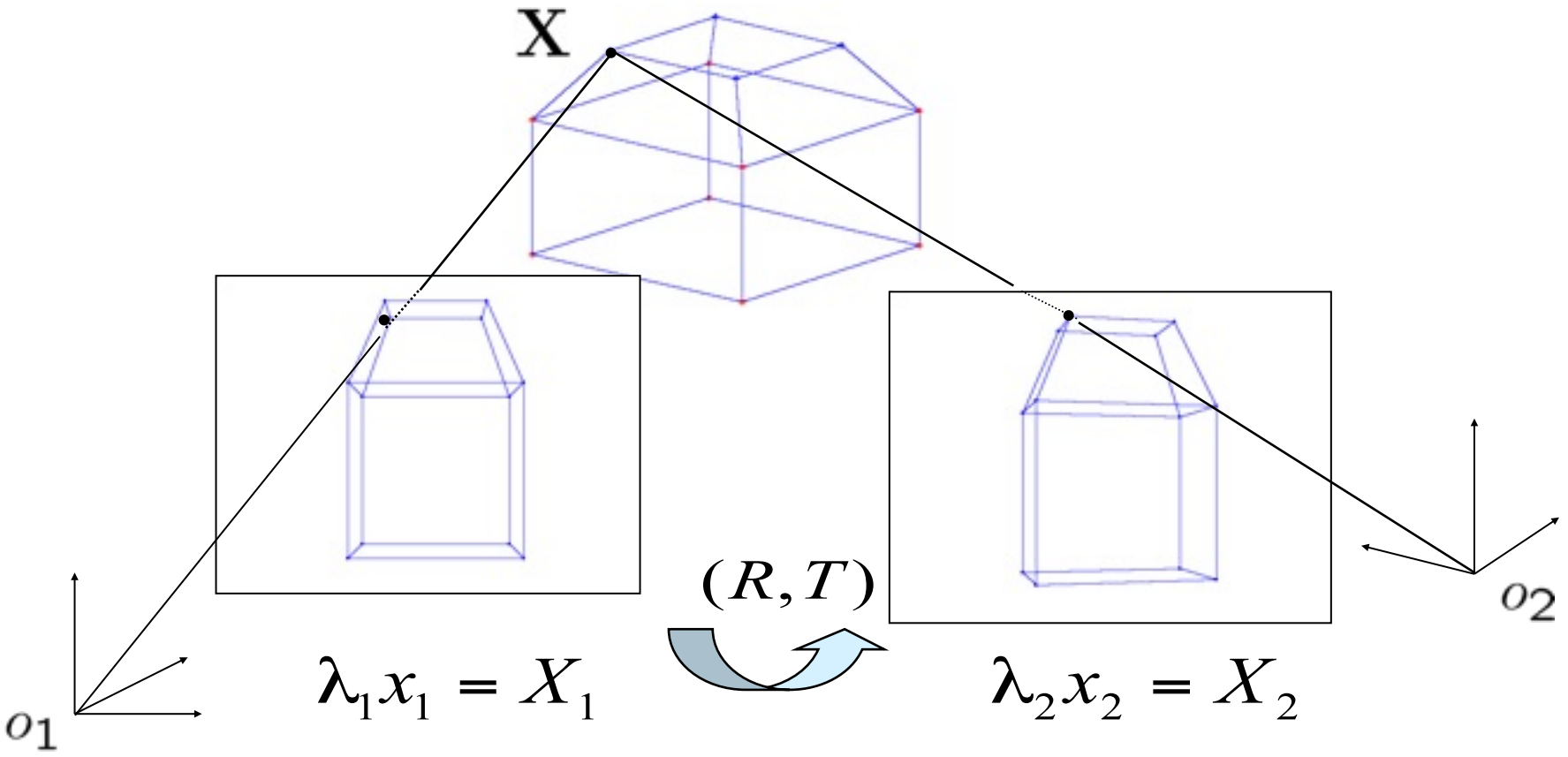


computer vision

Parallel Tracking and Mapping (PTAM) uses geometry to recover depth and provides camera pose estimation through monocular vision



Epipolar Geometry



$$X_2 = RX_1 + T \quad \text{or} \quad \lambda_2 x_2 = R\lambda_1 x_1 + T$$

$\lambda_1,\ \lambda_2$: feature depths in O_1, O_2; x_1, x_2: feature image coordinates in O_1, O_2

The Epipolar Constraint

elimination of depth

1. $\lambda_2 x_2 = R\lambda_1 x_1 + T$

$\times T$ $\times T$

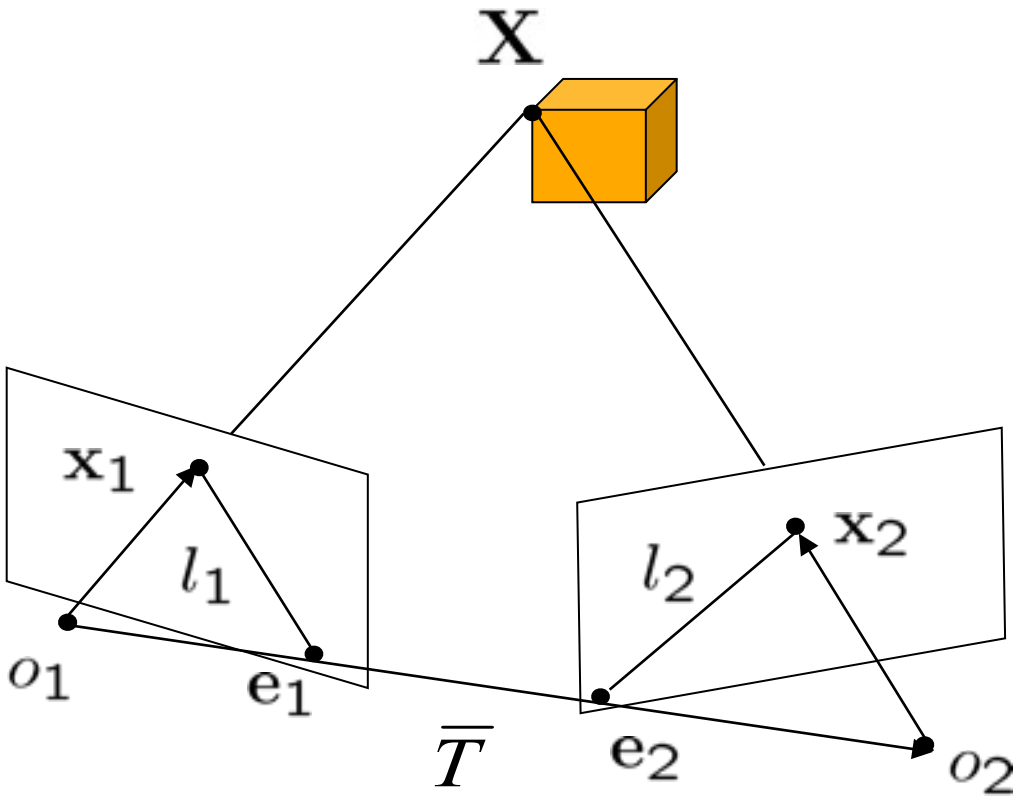
2. $\lambda_2 \hat{T} x_2 = \hat{T} R \lambda_1 x_1$

epipolar constraint

$$x_2^T \hat{T} R x_1 = 0$$

essential matrix

$$E = \hat{T} R$$



epipolar geometry entities

- (O_1, O_2, X) → epipolar plane
- O_1, O_2 → baseline
- l_1, l_2 → epipolar lines
- e_1, e_2 → epipoles

The Essential Matrix

- a special 3 x 3 matrix encoding epipolar geometry of two views

$$E = \left\{ \hat{T}R \mid R \in SO(3), T \in \Re^3 \right\}$$

- apparently 8 dof (9 matrix elements up to scale)
- practically 5 dof (3 – rotation, 2 – translation up to scale)
- given a point in one image, multiplying by the essential matrix will provide the epipolar line to search along in the second view

estimation of the essential matrix from a pair of views

(8 – point algorithm | 5 – point algorithm)

1. rewrite $E^S = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9]^T$

$$a = [x_1x_2, x_1x_2, x_1x_2, x_1x_2, x_1x_2, x_1x_2, x_1x_2, x_1x_2, x_1x_2]^T$$

2. collect epipolar constraints

$$\chi E^S = 0$$

$$\chi = [a^1, \ldots, a^n]^T$$

for uniqueness of solution
χ must have at least rank 5
need parallax motion
$T = 0$ won't work!

Monocular Visual Odometry vs Monocular Visual SLAM

Monocular Visual Odometry

feature matching between image frame

- faster: works in costant time
- accumulated small errors will cause drifts
- cannot mantain a consistent scale from couples of frames (need 3 or more views)
- motion singuralities (pure rotations do not constraint enough the motion)

Monocular Visual SLAM

feature matching between current image frames and a live **map**

- slower but accurate
- repeated observation of the same features ensures no drifts in trajectory estimate
- scale fixed once set the map
- extra cost for expanding and maintaining the map
- method based on EKF are limited by the size of the map

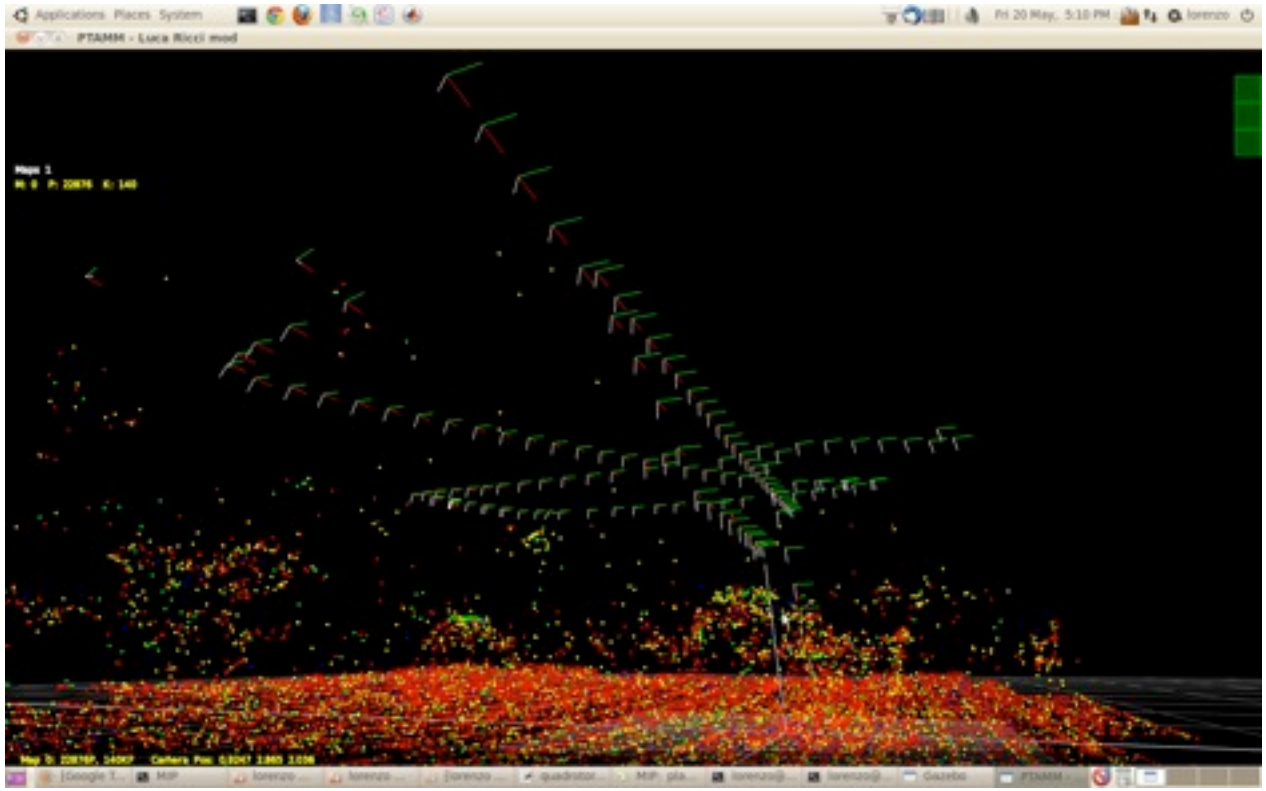
Parallel Tracking and Mapping (PTAM)

- monocular visual SLAM algorithm
- intended for small workspace AR (Augmented Reality) applications
- mapping and tracking are separated and run in two parallel threads
- mapping is based on keyframes
- new points are initialized with an epipolar search
- no feature or map uncertainties model (bundle adjustment on a vast number of image features)
- robust against partial camera occlusions (50 % of features available)

PTAM: what is a map

a collection of M map points and N keyframes

- map point : a 3D point in the world ($p_{j\,w} = (x_{j\,w}, y_{j\,w}, z_{j\,w}, 1)^T$)
- keyframe: a pyramid of greyscale 8bpp images (i.e. 640 x 480, … , 80 x 60) and an associated camera-centred coordinate frame (K_i)



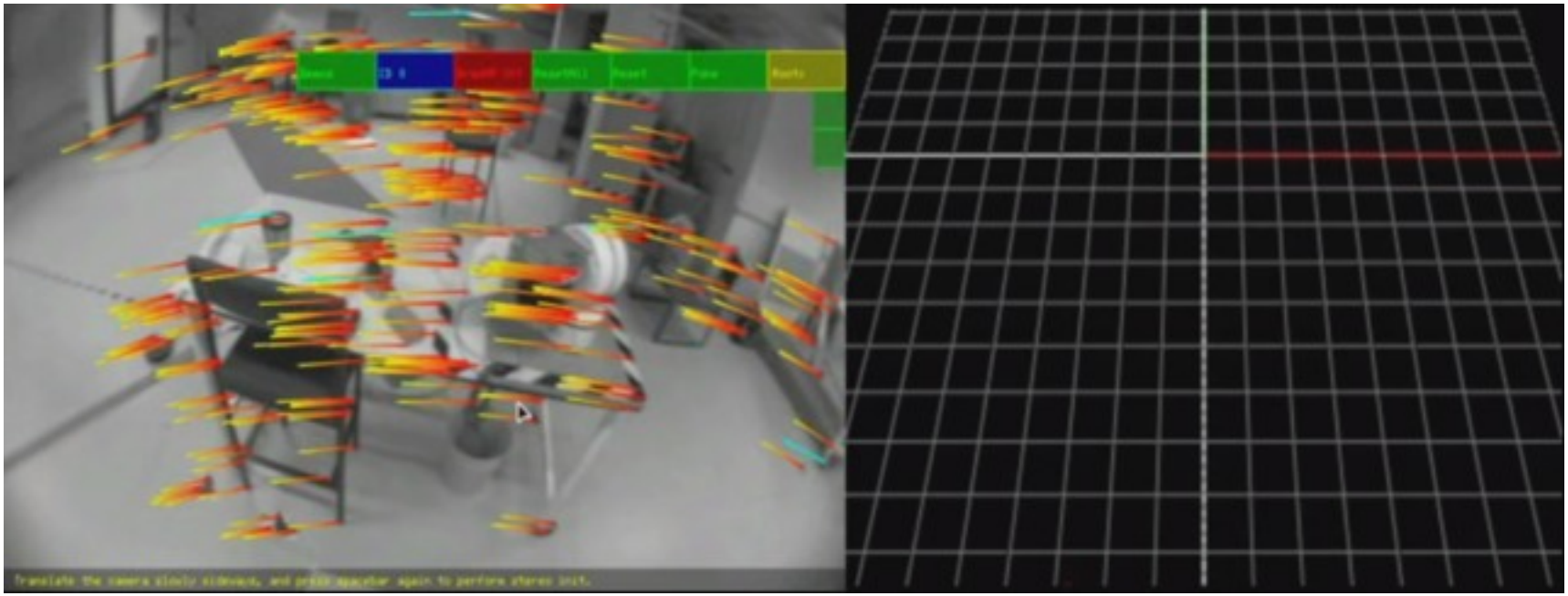
PTAM: map initialization

1. acquire the first keyframe

PTAM: map initialization

1. acquire the first keyframe

2. translate and rotate the camera while tracking the features (hyp. 10 cm translation)

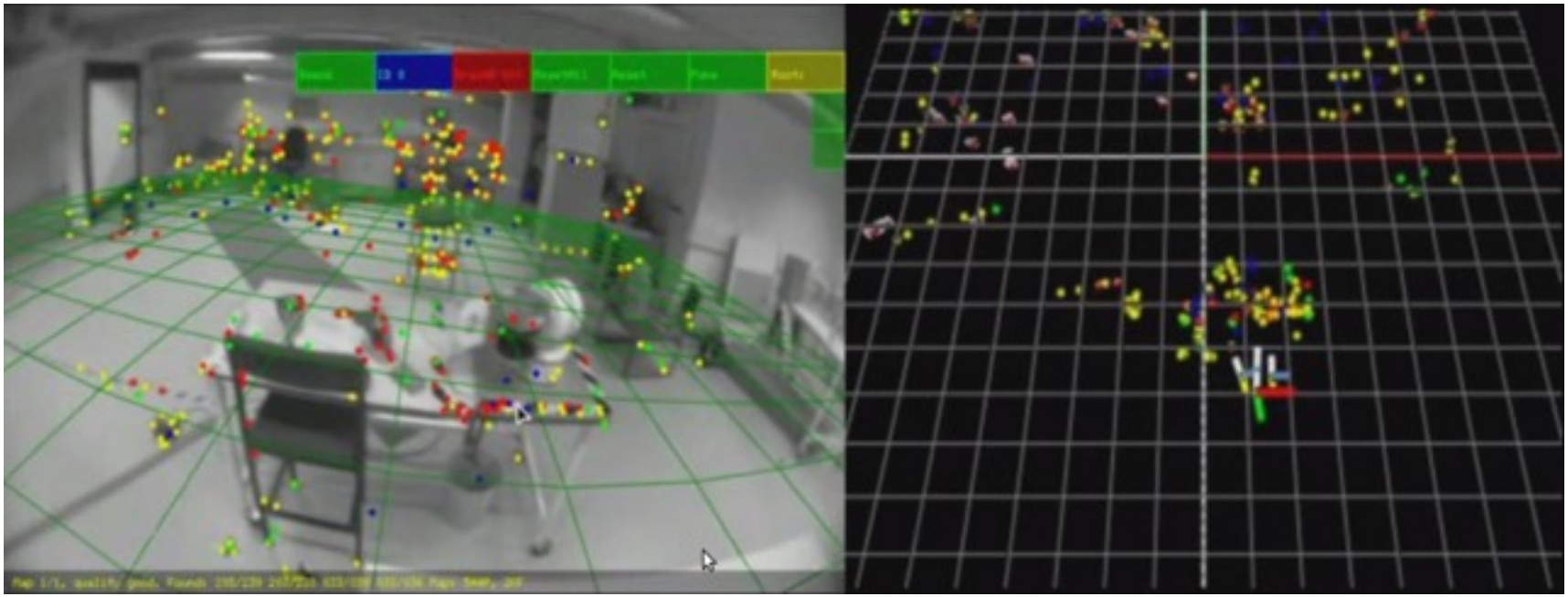


PTAM: map initialization

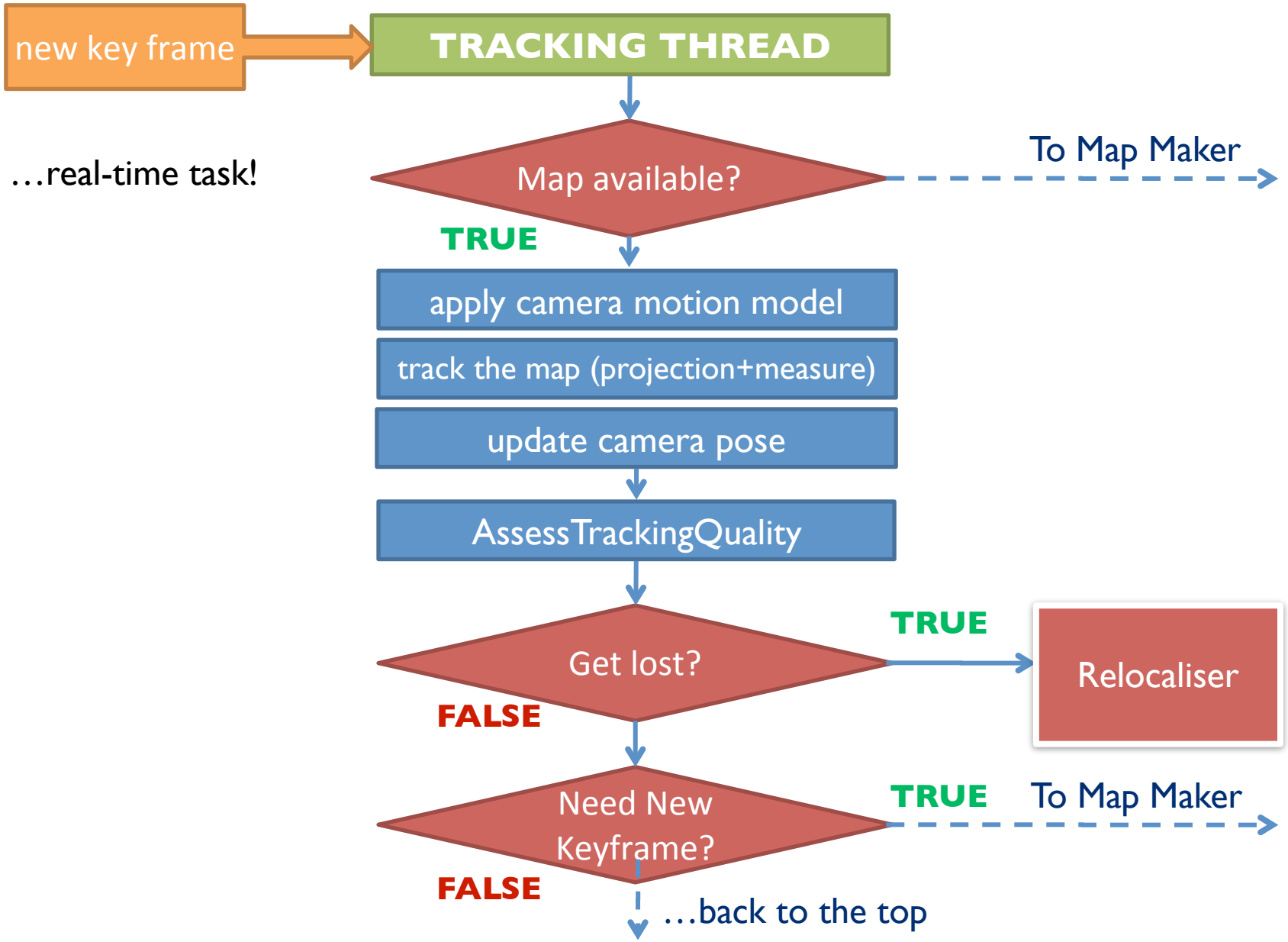
1. acquire the first keyframe (triggered by user)

2. translate and rotate the camera while tracking the features (hyp. 10 cm translation)

3. acquire the second keyframe (user triggered) and build the map (epipolar search)



PTAM: tracking thread



PTAM: more about the tracker

Map point search

Fixed - range image search around the point's predicted image location

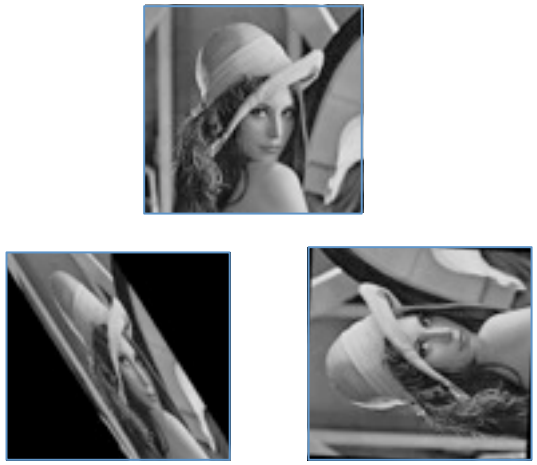
1. Map point in original keyframe (first spotted)

2. Fixed range patch extracted from original image (8 x 8)

3. Affine warp of the patch based on motion model pose estimate

4. Projection on the current view based on the motion model pose estimate

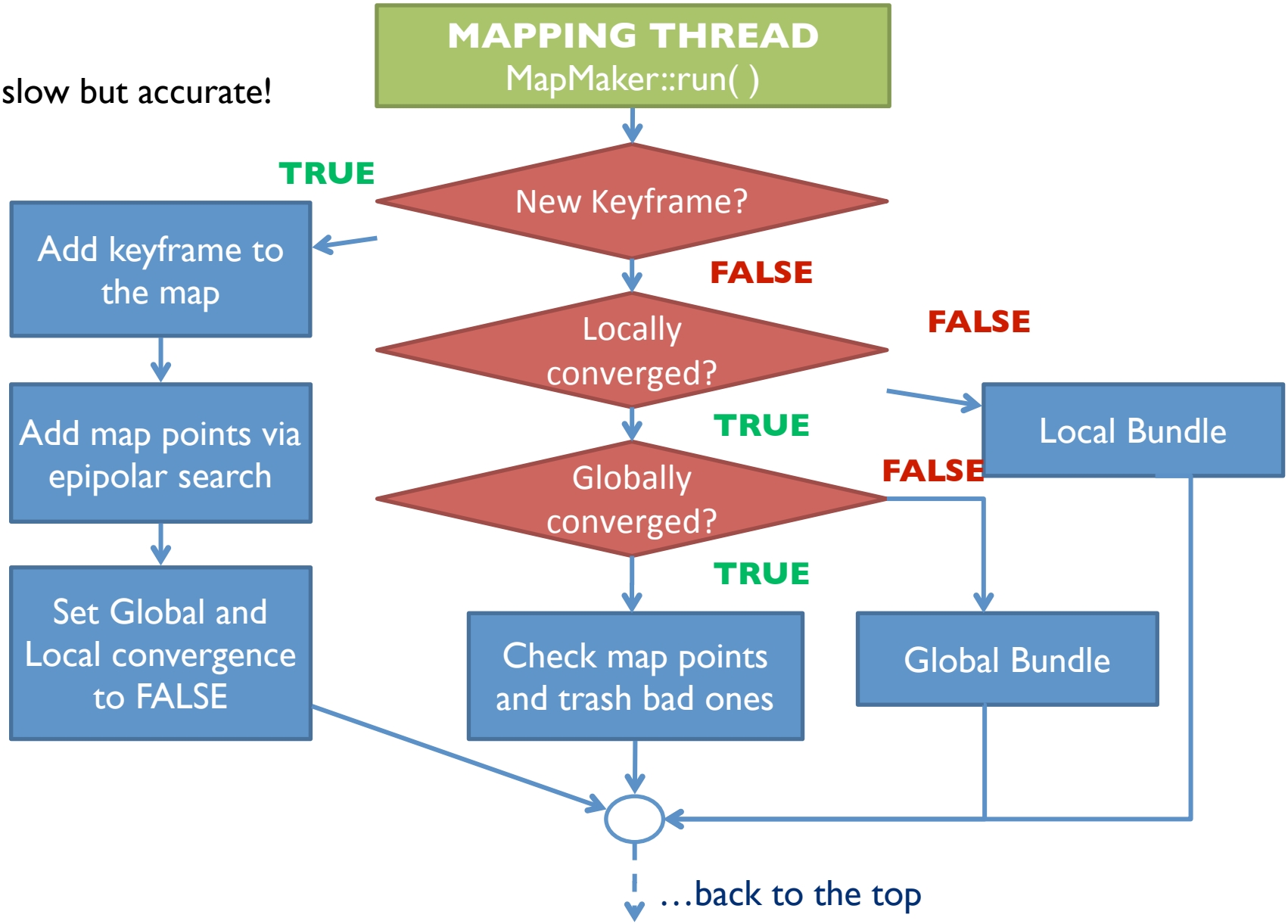
5. Comparison with the current view (Sum of Squared Differences score)



Examples of affine warping

PTAM: mapping thread

…slow but accurate!

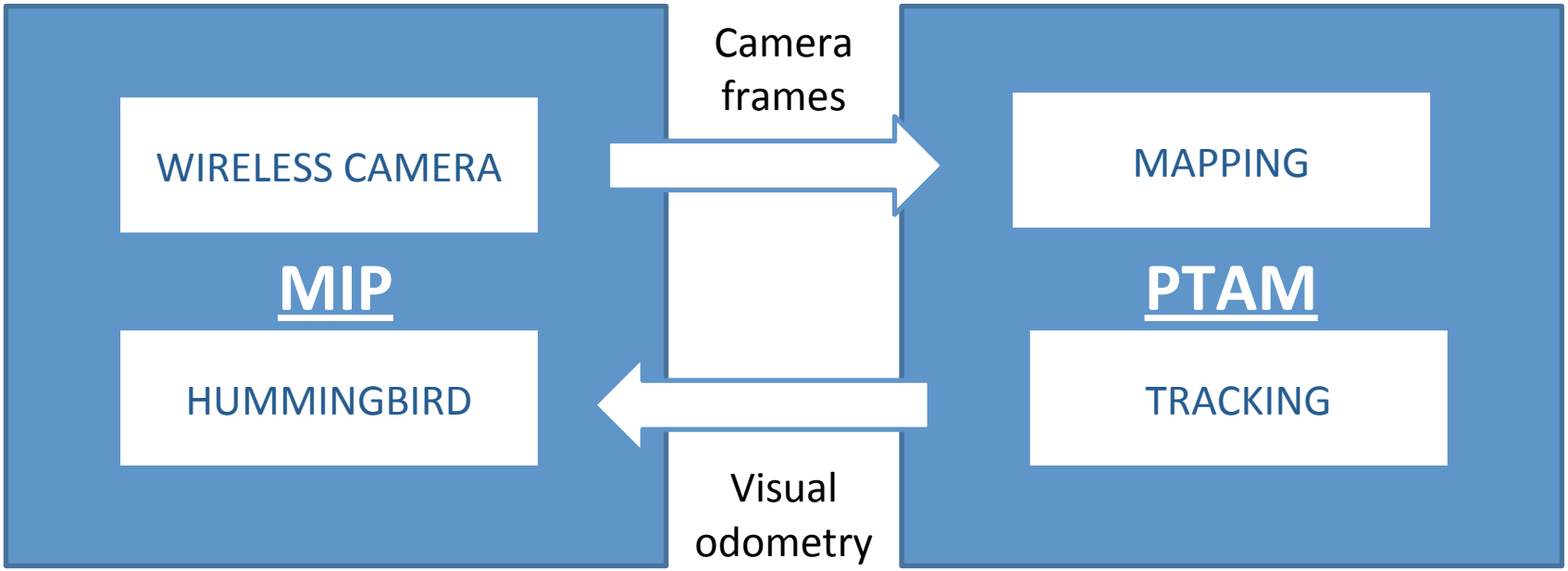


PTAM: monocular SLAM on a quadrotor?

some basic facts:

- Hummingbird quadrotor equipped with a monocular camera facing downwards
- Multi-Robot-Integration Platform (MIP) implements quadrotor communication, wireless camera sensor interface and some other useful stuff

passing camera frames from MIP to PTAM will do the trick...



MIP: an overview

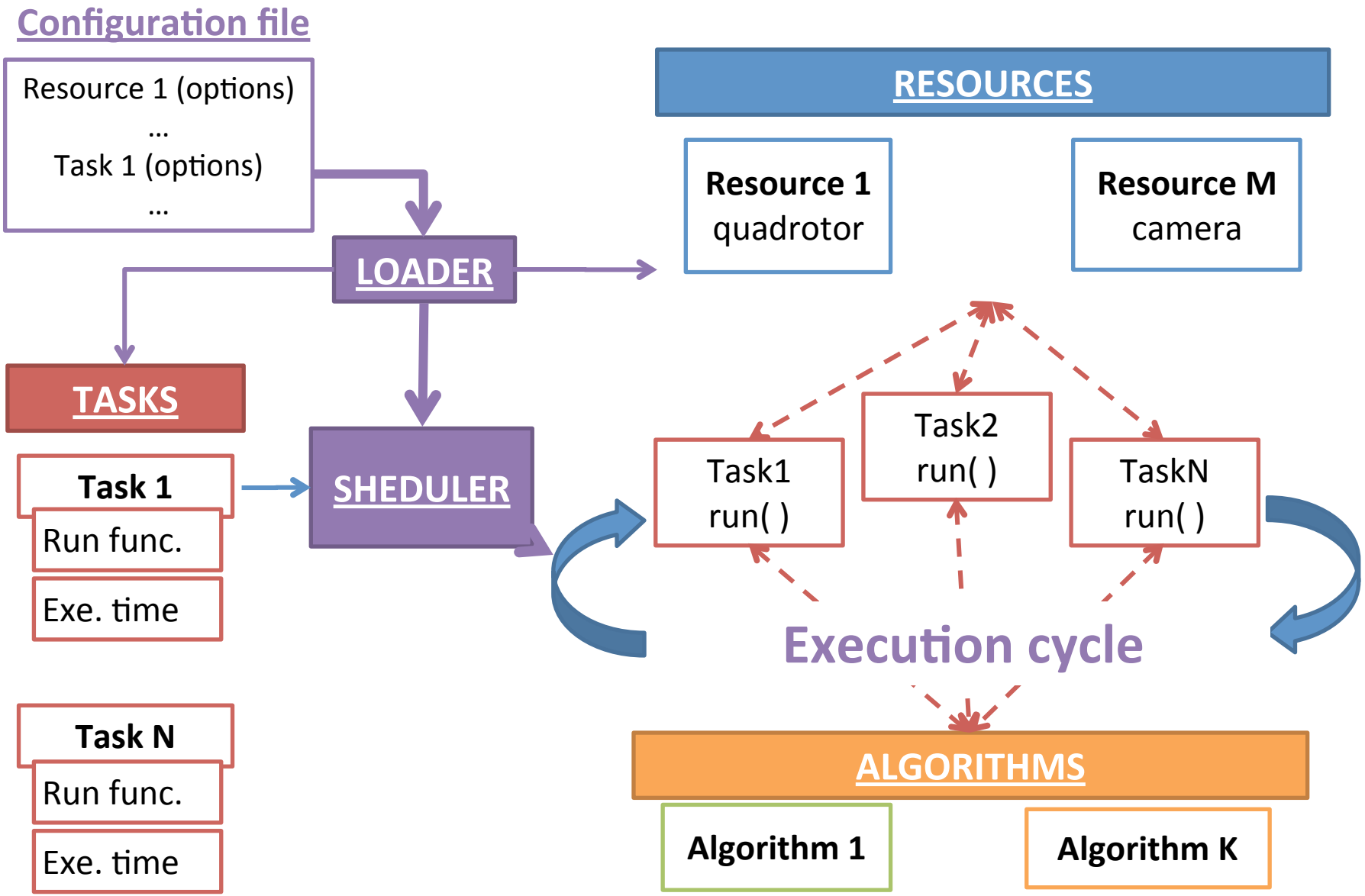
a C++ software aimed to develop control and estimation robotics algorithms

- good level of modularity
- use of abstracted low-level interfaces
- interface with 3D simulation environment (Player/Gazebo)

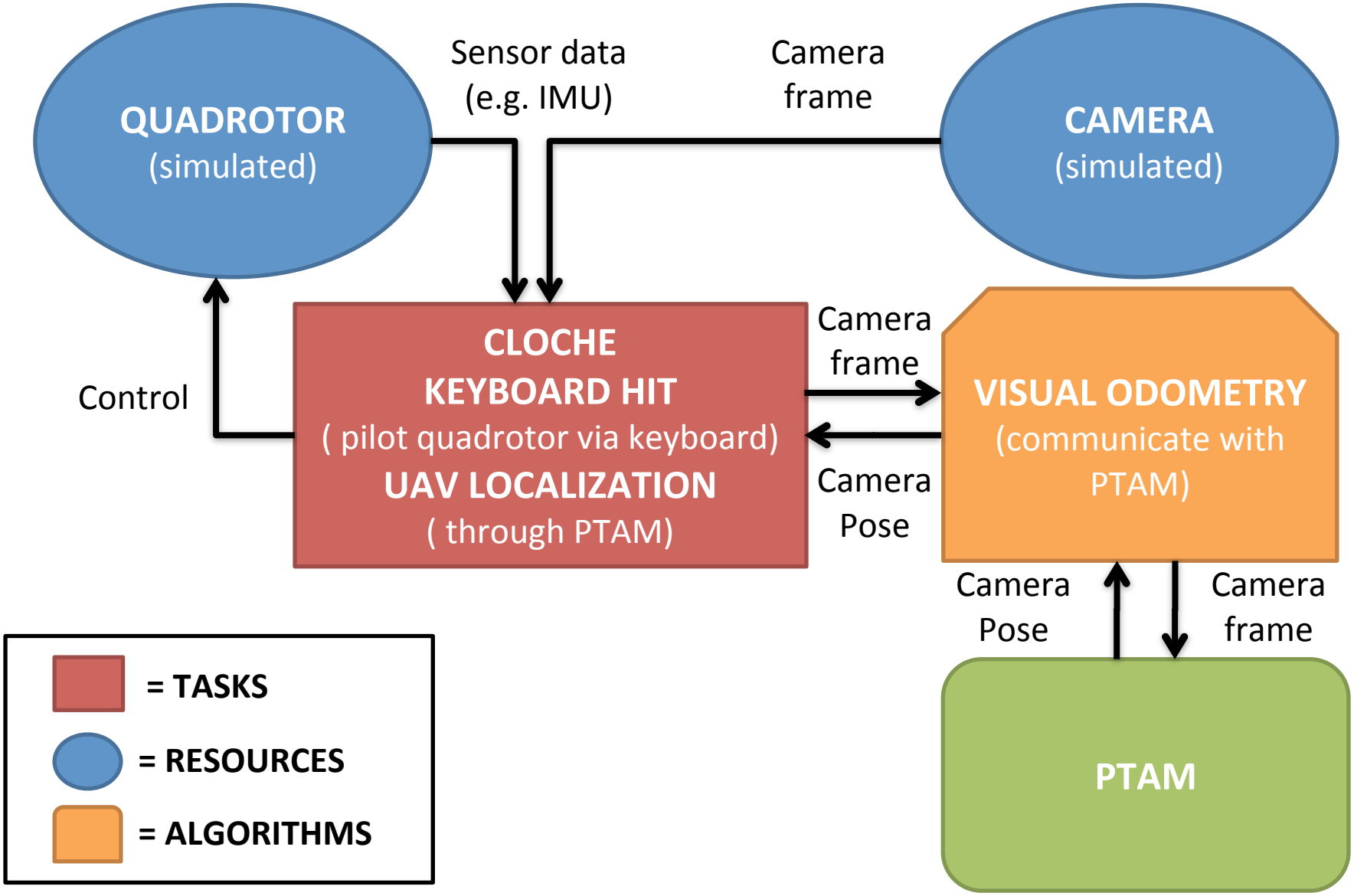
MIP components

Main	main of the program. Here is created and launched the Scheduler
Baselib	basic library for general purpose and robotics functionalities , e.g. IP communication, pose class, matrix class,…
Resources	class providing interface modules respect to the hardware or the MIP platform facilities, e.g. camera, quadrotor, keyboard, ...
Algorithms	class collection of robotics algorithms, e.g. Visual odometry algorithm, estimate (Kalman filtering)
Tasks	High level robot activities that must be execute in parallel, glueing algorithms and resources, e.g. tracking, deployment, target navigation, mutual localization, entrapment, exploration,…

MIP: an overview

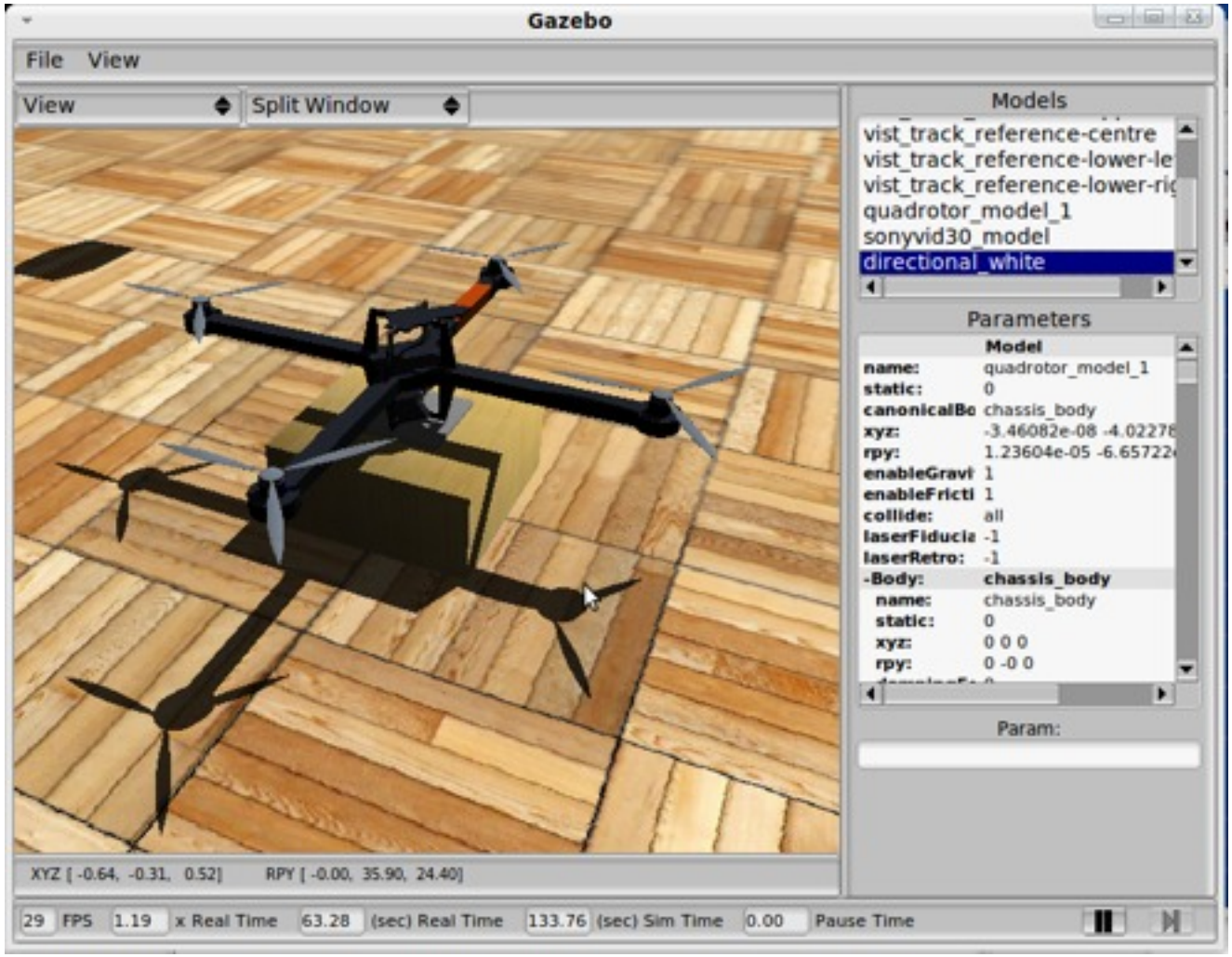


MIP: use with PTAM and quadrotor



MIP: use with PTAM and quadrotor

1. Start 3D simulation environment (Player/Gazebo)



MIP: use with PTAM and quadrotor

1. Start 3D simulation environment (Player/Gazebo)

2. Select the configuration file and run MIP

```
RESOURCES
        Keyboard
        UavPlayerUav
        -usePTAMM false
        CameraPlayer
TASKS
        KeyboardHit
        ClocheTask
        UAVPositioningSystem

-debugLevel 0
```

MIP: use with PTAM and quadrotor

1. Start 3D simulation environment (Player/Gazebo)

2. Select the configuration file and run MIP

```
RESOURCES
        Keyboard
        UavPlayer0uv
        -usePTAMM false
        CameraPlayer
TASKS
        KeyboardKit
        ClocheTask
        UAVPositionSystem

-debugLevel 0
```

3. Start PTAM and navigate

References

[1] Georg Klein and David Murray ,
"*Parallel Tracking and Mapping for Small AR Workspaces*" ,
In Proc. International Symposium on Mixed and Augmented Reality (ISMAR'07, Nara), 2007

[2] R. I. Hartley and A. Zisserman, "*Multiple View Geometry in Computer*",
Vision. Cambridge University Press, second edition, 2004.

[3] Y. Ma and S. Soatto, "*A invitation to 3-D vision, from images to geometric models*",
Springer-Verlag New York, 2004.

[4] J. Shi and C. Tomasi, "*Good features to track*",
In *Proc. IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR '94),*
pages 593–600. IEEE Computer Society, 1994.