TEXAS A&M UNIVERSITY CSE DEPARTMENT

CSCE 643 Final Project
Implementation of the Automatic Map Initialization in ORB-SLAM

Yuan-Peng Yu

The goal of the project is to to find whether the homography matrix or the fundamental matrix fits the given two frames. The selected model can be decomposed to compute the relative camera poses. The algorithm of the project should satisfy the following conditions.

- For each point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, the algorithms should be executed automatically without human intervention.
- The algorithm will select the homography matrix model in a planar scene or low parallax condition.
- The algorithm will select the fundamental matrix model in a non-planar scene.

Figure 0.1 is the flowchart of the automatic map initialization and each step will be explained in each section.

1 FIND CORRESPONDENCE FEATURES

The ORB-SLAM algorithm is categorized into the feature-based method. So far, the ORB-SLAM system presented by Mur-Artal has the best performance. The name of the algorithm comes from the feature descriptor: ORB (Oriented FAST and Rotated BRIEF). ORB feature descriptor is evolved from the FAST and BRIEF descriptor. Rublee proposed the ORB descriptor to replace the traditional SIFT (Scale-Invariant Feature Transform) feature detector to improve the performance of a SLAM system. The main advantages of adopting the ORB descriptor include rotation invariance and high noise intolerance. ORB descriptor is faster and more accurate than the original FAST descriptor. When compared to BRIEF, ORB has a lower consumption on computing. The efficiency of ORB feature descriptor enables a better image-matching ability and fast computation.

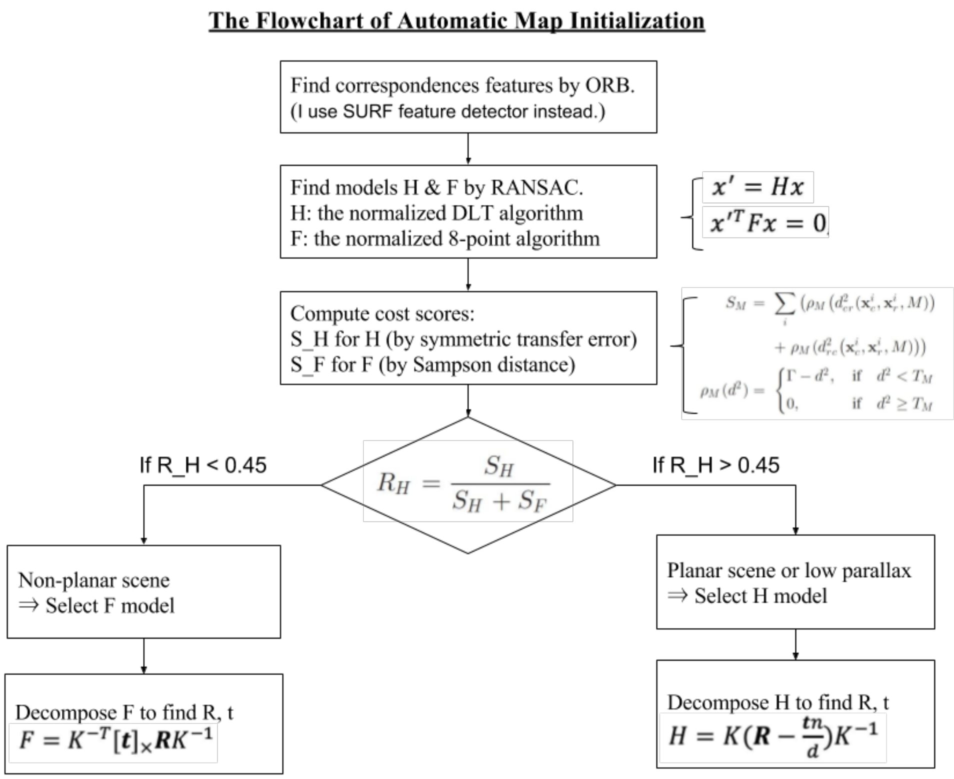

Figure 0.1: The flowchart of the automatic map initialization.

In the project, I adopt the speeded up robust features (SURF) detector to find correspondence features because it is contained in the Matlab Computer Vision toolbox. The difference between SURF and ORB is not our focus in the project. Using SURF can help us find feature points to compute two models: the homography matrix and the fundamental matrix. It is important that the two frames must contain enough feature points so that the model can be computed. The homography matrix needs four correspondence pairs whereas the fundamental matrix needs at least seven correspondence pairs.

2 Compute H & F

2.1 Compute H

A set of points in the second image can be regard as a projective transformation of a set of corresponding points in the first image. According to Hartley and Zisserman, each image is in the projective space $\mathbf{P}^2$. A linear transformation mapping from $\mathbf{P}^2$ to $\mathbf{P}^2$can be represented by a non-singular matrix.

$$\mathbf{x}' = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{x} \tag{2.1}$$

The matrix multiplication can be written as the following equation.

$$\mathbf{x}' = H\mathbf{x} \tag{2.2}$$

The 2D homography matrix contains nine elements. Because is defined up to scale, it is generally set as one. The number of degrees of freedom becomes eight for the projective transformation. For a point in the image, the number of degrees is two. To solve the homography matrix, we need to find at least four corresponding points to provide eight equations. Thus, an exact solution for the homography matrix requires four points correspondences in two consecutive images.
According to Hartley and Zisserman, the Gold Standard algorithm is used to compute the 2D homography matrix. First, four corresponding points can determine the homography matrix . An initial estimate of can be computed by the normalized Direct Linear Transformation (DLT) algorithm, or the Random Sample Consensus (RANSAC) algorithm. Second, an error vector is evaluated by the selected cost function. Iterative minimization methods, such as Newton algorithm or Levenberg-Marquardt algorithm, will be adopted to minimize the error vector. When the cost function obtains a minimized error vector, the optimal solution for H is acquired.

2.2 Compute F

For any correspondence points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, the fundamental matrix is defined by the equation

$$x'^T F x = 0.$$

In the homogeneous coordinates, $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$. The equation can be extended as

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'x_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0,$$

where $f = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T$ and

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \tag{2.3}$$

We can rewrite the equation into matrix form.

$$\begin{bmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0. \tag{2.4}$$

Because f_{33} is up to scale, the rank of the matrix is at most 8. Because of the singularity constraint of F, the matrix needs at least 7 points to solve.

$$\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & n \end{bmatrix} f = 0$$
$$\Rightarrow Af = 0.$$

Because the fundamental matrix is singular, the rank of F is two. To achieve this condition, we first use the SVD to decompose F.

$$\mathrm{F} = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} V^T$$

Then, let the new fundamental matrix

$$\mathrm{F}' = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} V^T = U_1\sigma_1V_1^T + U_2\sigma_2V_2^T$$

The Frobenius distance between F' and F

$$\min \parallel \mathrm{F} - \mathrm{F}' \parallel_F,$$

can be minized.

2.3 The score functions for H & F

To obtain an optimal result, we use RANSAC method to find a robust estimation. A threshold is set based on the measurement noise. A typical value of a threshold is 3σ. Both the homography matrix and the fundamental matrix are acquired inside a RANSAC scheme. At each RANSAC iteration, we compute scores S_H and S_F for the homography matrix and the fundamental matrix, respectively. The score S_H is a symmetric transfer error.

$$S_H = \sum_i p_H(d(\mathbf{x}_i, H^{-1}\mathbf{x}_i)^2) + p_H(d(\mathbf{x}_i^{'}, H\mathbf{x}_i)^2) \tag{2.5}$$

where

$$p_H(d^2) = \begin{cases} 5.99 - d^2, & \text{if } d^2 < 5.99 \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

The score S_F is a first-order geometric error, which is also called as Sampson distance.

$$S_F = \sum_i \frac{(\mathbf{x}_i^{'T} F\mathbf{x}_i)^2}{(F\mathbf{x}_i)_1^2 + (F\mathbf{x}_i)_2^2 + (F^T\mathbf{x}_i^{'})_1^2 + (F^T\mathbf{x}_i^{'})_2^2} \tag{2.7}$$

where

$$p_H(d^2) = \begin{cases} 5.99 - d^2, & \text{if } d^2 < 3.84 \\ 0, & \text{otherwise} \end{cases} \tag{2.8}$$

The number 5.99 and 3.84 is the outlier rejection threshold where the χ^2 test at 95%. Here, we assume that the measurement error is a standard deviation of 1 pixel. The homography that has the highest score and the fundamental matrix that has the highest will be adopted by RANSAC.

3 Model selection

The requirement of the algorithm is to select the homography matrix model if the scene has more planar features. Also, pure rotation and low parallax will be explained by the homography matrix. If the scene contains more non-planar features, the model will be the fundamental matrix. After we compute the cost scores S_H and S_F, we can compute R_H to decide the model. ORB-SLAM suggest a robust heuristic function as below

$$R_H = \frac{S_H}{S_H + S_F} \tag{3.1}$$

$$\text{Model} = \begin{cases} H, & \text{if } R_H > 4.5 \\ F, & \text{otherwise} \end{cases} \tag{3.2}$$

If $R_H > 4.5$, the model will be the homography matrix. Otherwise, the fundamental matrix will be adopted as the model.

4 Simulation method and Results

To verify the algorithm, we generate a world scene. Cameras and feature are placed in the world coordinates. The simulation help us to verfify our computation without worrying the measurement noises on images. There are two configuration of cameras: co-centered and non-cocenter. Also, the correspondence features can be divided into two types: planar and non-planar features. In our simulation, the feature distribution are designed in Figure 4.1 and Figure 4.2. There are total four combinations. We tested the algorithm in the four scenarios. We compute the homography matrix, the fundamental matrix, and R_H so that the proper model in the two keyframe can be selected. The following subsection will show our results.

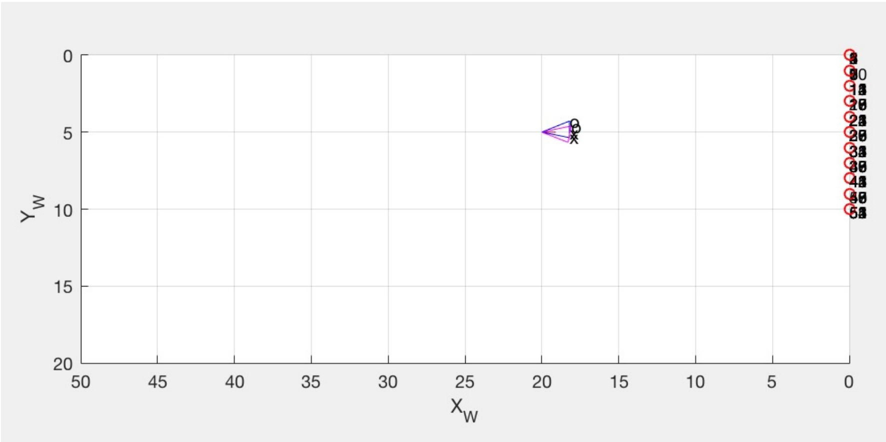

Figure 4.1: The top view shows that the features are planar.

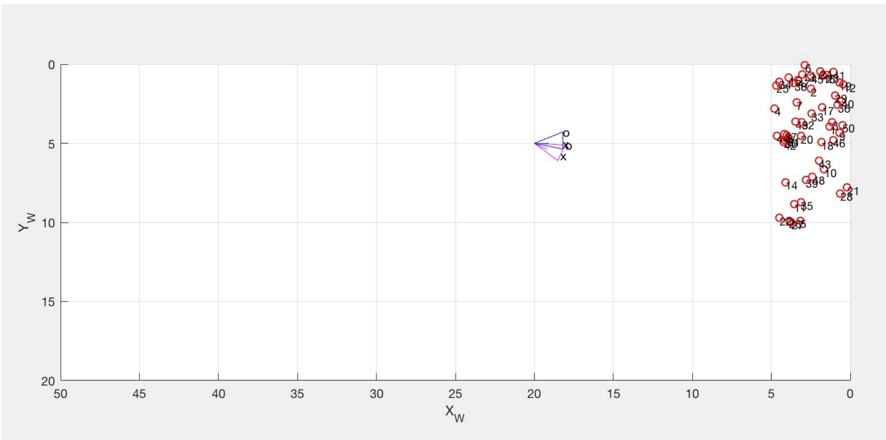

Figure 4.2: The top view shows that the features are distributed randomly. Features are not co-planar.

Figure 4.3 shows the world that contains the cameras and features. Figure 4.4 displays the image views from the camera poses.

Figure 4.5 show a planar features scene and the cameras are co-cented. We use SURF feature detector to match the correspondence features. The result of the model selection is displayed in Figure 4.6.

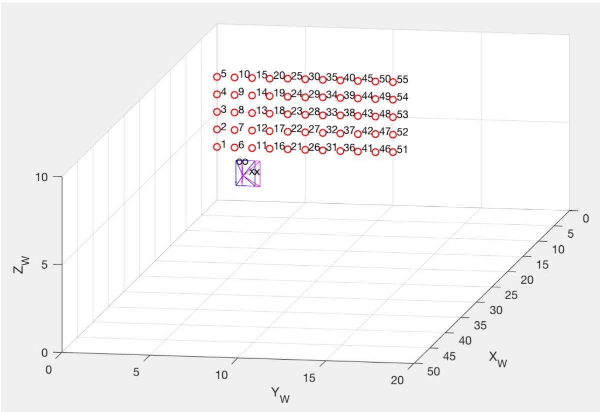

Figure 4.3: Co-centered camera in the planar features scene.

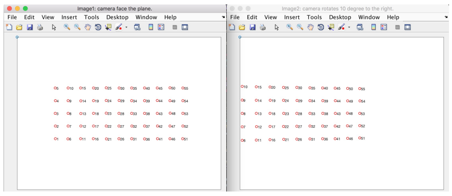

Figure 4.4: The two image view from camera positions.

Figure 4.5: Co-centered camera in the planar features scene.

```
num_H_inliers =

   705

num_F_inliers =

   891

H =

   -0.4354    0.0056 -205.7192
    0.0672   -0.5707  -26.2444
    0.0002   -0.0000   -0.6579

F =

    0.0000    0.0000   -0.0058
   -0.0000    0.0000   -0.0030
    0.0038    0.0014    1.9198

scoreH =

   8.9329e+03

scoreF =

   5.0568e+03

R_H =

    0.6385

Model selection result ==> The homography matrix.
```

Figure 4.6: The result of H, F, and the model selection.

4.2 Co-centered camera in the non-planar features scene

Figure 4.7 shows the world that contains the cameras and features. Figure 4.8 displays the image views from the camera poses.
Figure 4.9 show a planar features scene and the cameras are co-cented. We use SURF feature detector to match the correspondence features. The result of the model selection is displayed in Figure 4.10.

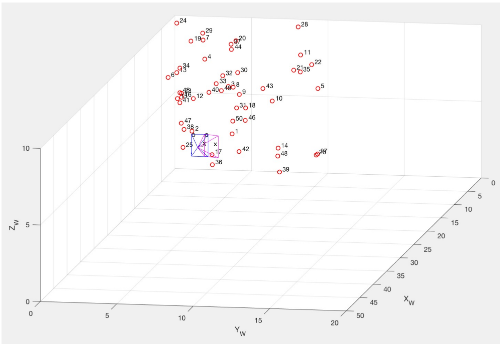

Figure 4.7: Co-centered camera in the non-planar features scene.

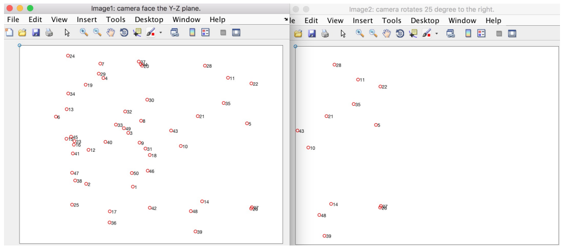

Figure 4.8: The two image view from camera positions.

Figure 4.9: Co-centered camera in the non-planar features scene.

```
num_H_inliers =

   175


num_F_inliers =

   282


H =

    0.6137   -0.0662 -936.8768
    0.1698    0.5711 -300.5788
    0.0002   -0.0000    0.2971


F =

   -0.0000   -0.0000    0.0004
    0.0000   -0.0000    0.0016
   -0.0017   -0.0041    2.8377


scoreH =

   1.4792e+03


scoreF =

   1.5639e+03


R_H =

    0.4861

Model selection result ==> The homography matrix.
```

Figure 4.10: The result of H, F, and the model selection.

4.3 Camera rotates and translates in the planar features scene

Figure 4.11 shows the world that contains the cameras and features. Figure 4.12 displays the image views from the camera poses.
Figure 4.13 show a planar features scene and the cameras are co-cented. We use SURF feature detector to match the correspondence features. The result of the model selection is displayed in Figure 4.14.

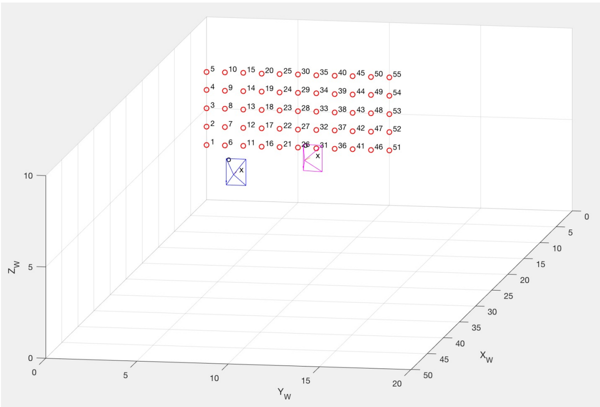

Figure 4.11: Camera rotates and translates in the planar features scene

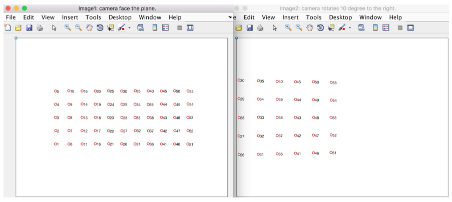

Figure 4.12: The two image view from camera positions.

Figure 4.13: Camera rotates and translates in the planar features scene

```
num_H_inliers =

   356

num_F_inliers =

   419

H =

   -0.4523   -0.0378    8.6400
    0.2151   -0.6249 -118.5689
    0.0004   -0.0000   -0.7815

F =

    0.0000   -0.0000    0.0098
    0.0000    0.0000   -0.0110
   -0.0114    0.0077    1.7916

scoreH =

   3.5460e+03

scoreF =

   2.2669e+03

R_H =

    0.6100

Model selection result ==> The homography matrix.
```

Figure 4.14: The result of H, F, and the model selection.

4.4 Camera rotates and translates in the non-planar features scene

Figure 4.15 shows the world that contains the cameras and features. Figure 4.16 displays the image views from the camera poses.
Figure 4.17 show a planar features scene and the cameras are co-cented. We use SURF feature detector to match the correspondence features. The result of the model selection is displayed in Figure 4.18.

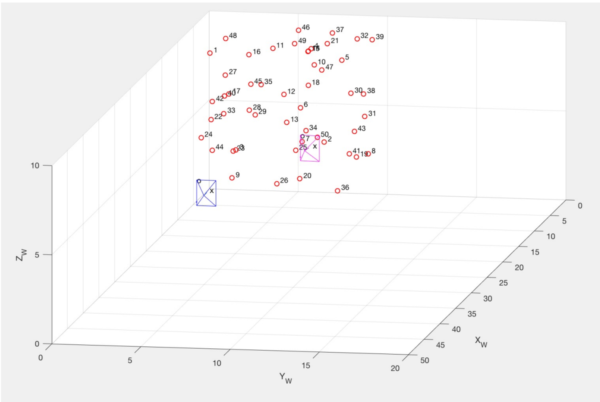

Figure 4.15: Camera rotates and translates in the non-planar features scene.

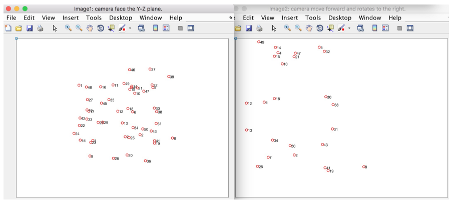

Figure 4.16: The two image view from camera positions.

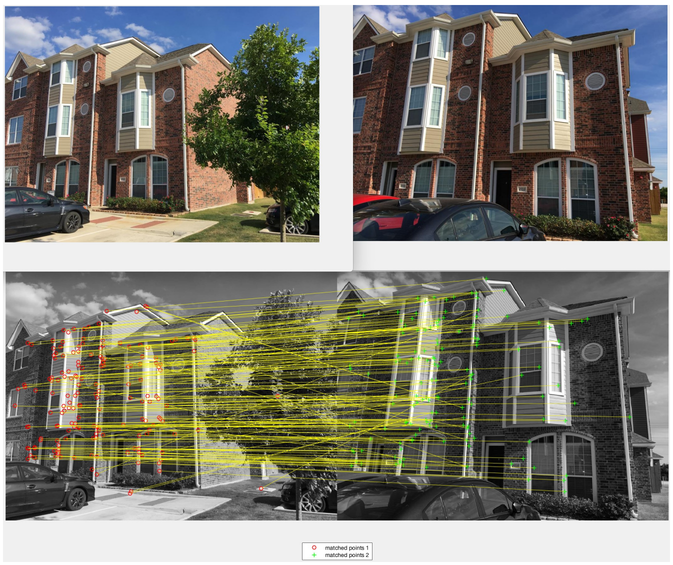

Figure 4.17: Camera rotates and translates in the non-planar features scene.

```
num_H_inliers =

    48


num_F_inliers =

   120


H =

    0.9133   -0.0409  -15.1022
    0.0514    0.7277  -90.9050
    0.0000   -0.0001    0.5908


F =

    0.0000    0.0000   -0.0007
   -0.0000    0.0000   -0.0046
    0.0013    0.0056   -0.7144


scoreH =

  336.1267


scoreF =

  599.9153


R_H =

    0.3591

Model selection result ==> The fundamental matrix.
```

Figure 4.18: The result of H, F, and the model selection.

4.5 Source code & instruction for compiling

The source code is included in the submitting zip file. The codes are written in Matlab. The executeable main files are:

- *main_cocenter_plane_dominated.m* : Compute the H, F, and the scores of model selection for the planar features. The cameras are co-centered.
- *main_cocenter_random_features.m* : Compute the H, F, and the scores of model selection for the non-planar features. The cameras are co-centered.
- *main_plane_dominated.m* : Compute the H, F, and the scores of model selection for the planar features. Two cameras are not co-centered.
- *main_random_features.m* : Compute the H, F, and the scores of model selection for the non-planar features. The cameras are co-centered. Two cameras are not co-centered.
- *image_test.m* : Verifies the algorithms by images.

5 Reference

- Andrew Zisserman, MATLAB Functions for Multiple View Geometry (http://www.robots.ox.ac.uk/vgg/hzbook/code/).
- Peter Kovesi, MATLAB and Octave Functions for Computer Vision and Image Processing (http://www.peterkovesi.com/matlabfns/index.html)
- Raul Mur-Artal, J. M. M. Montiel and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, October 2015. (http://webdiis.unizar.es/ raulmur/orbslam/)