AR QR code

1. Overview

ARTag (AR tag, AR means "augmented reality") is a fiducial marking system, which can be understood as a reference for other objects. It looks similar to a QR code, but its encoding system is very different from that of a QR code. It is mostly used in applications such as camera calibration, robot positioning, augmented reality (AR), etc. One of its important functions is to identify the pose relationship between an object and a camera. ARTag can be attached to an object, or ARTag tags can be attached to a plane to calibrate the camera. After the camera recognizes the ARTag, it can calculate the position and pose of the tag in the camera coordinates.

ar_track_alvar has 4 main functions:

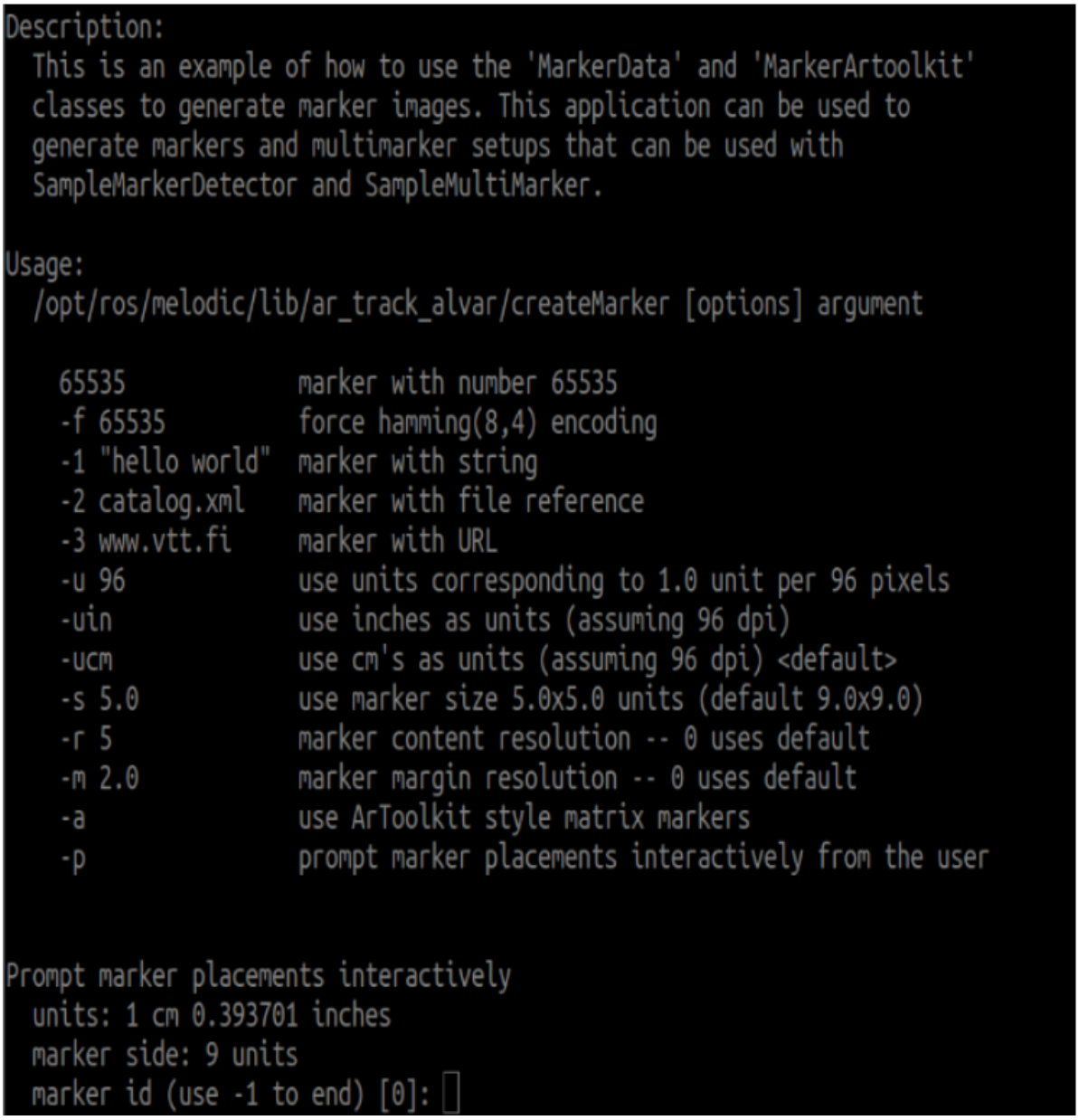
- Generate AR tags of different sizes, resolutions, and data/ID encodings.
- Identify and track the pose of a single AR tag, optionally integrating kinect depth data (when kinect is available) for better pose estimation.
- Identify and track the pose of a "bundle" consisting of multiple tags. This allows for more stable pose estimation, robustness to occlusion, and tracking of multi-sided objects.
- Automatically computes spatial relationships between tags in a bundle using camera images so that users don't have to manually measure and enter tag positions in an XML file to use bundle functionality.

Alvar is newer and more advanced than ARToolkit, which has been the basis for several other ROS AR tag packages. Alvar features adaptive thresholding to handle various lighting conditions, optical flow-based tracking for more stable pose estimation, and an improved tag recognition method that doesn't slow down significantly as the number of tags increases.

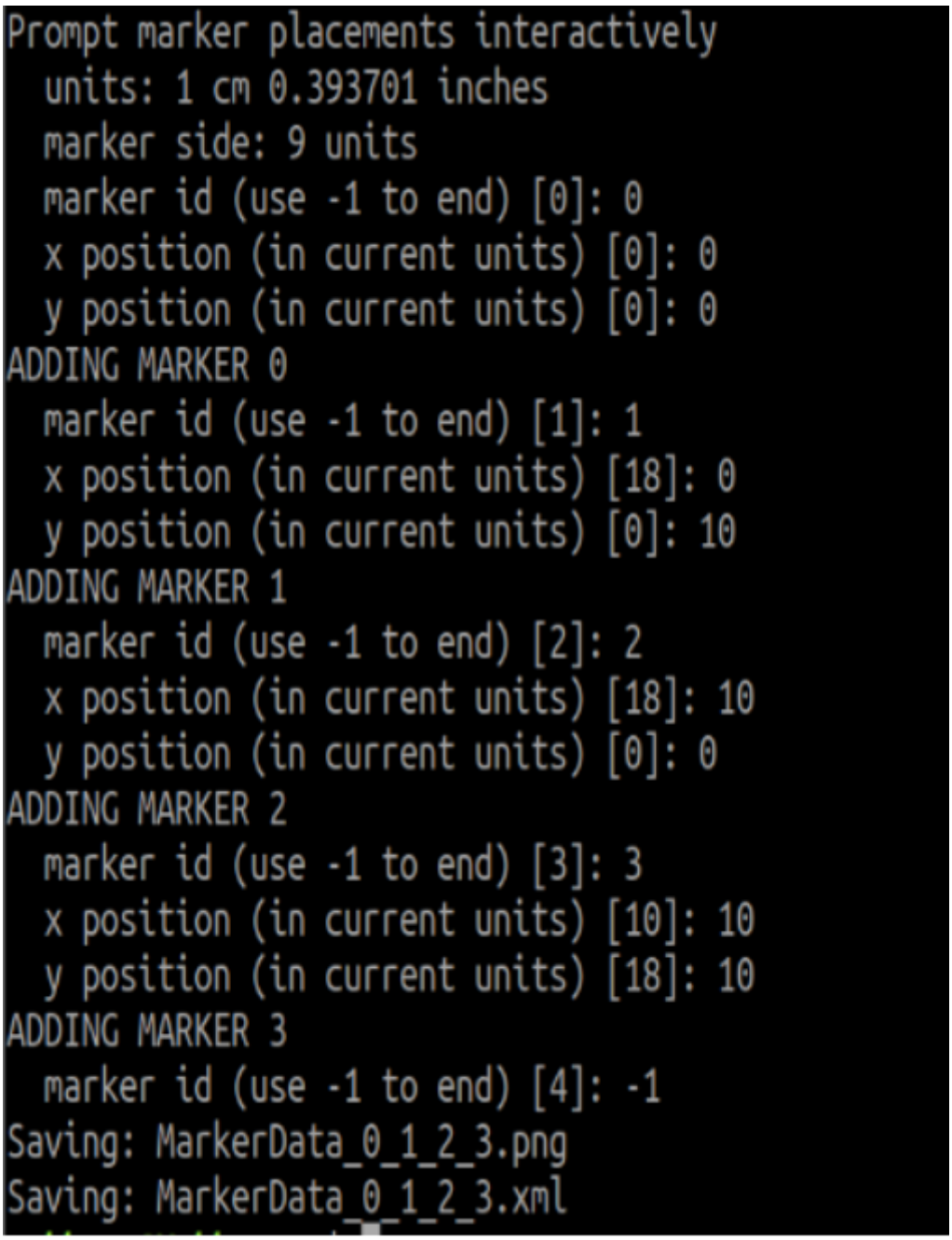
2. Create AR QR code

- Continuously generate multiple labels on one image

```
roscore
rosrun ar_track_alvar createMarker
```



You can enter the 【ID】 and location information here, and enter 【-1】to end. You can generate one or more and design the layout yourself.



- Generate a single number: command + parameter directly generates a digital image.

  ```
  rosrun ar_track_alvar createMarker 11
  rosrun ar_track_alvar createMarker -s 5 33
  ```

  11: The number is the QR code of 11. -s: Specifies the image size. 5: A 5x5 image. 33: The number is the QR code of 33.

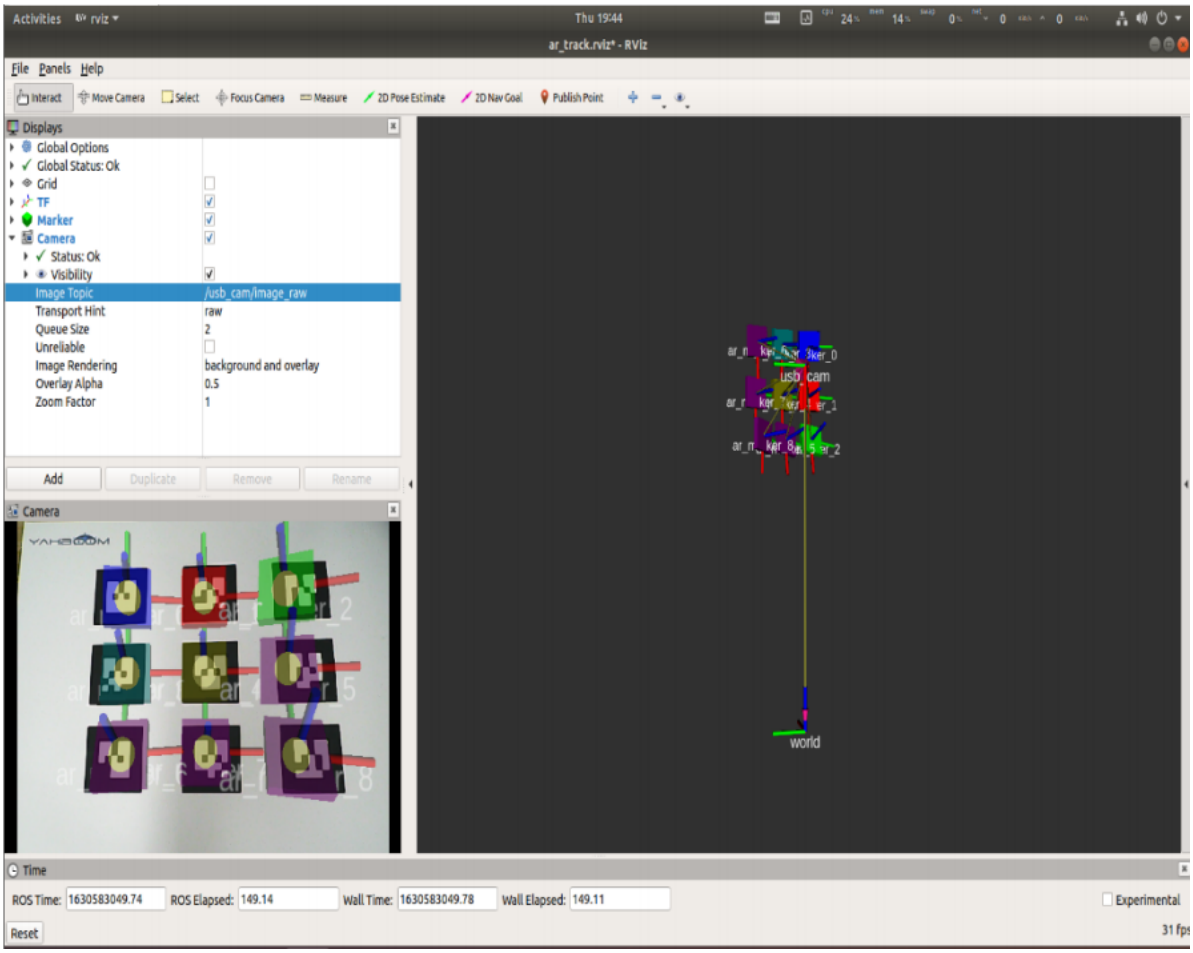
The files generated by the above two generation methods are stored in the terminal directory where the command is run.

3. ARTag Identification

Input following command:

```
roslaunch jetcobot_visual ar_track.launch open_rviz:=true
```

- The open_rviz parameter is turned on by default, and the TAG tag code is placed on the camera screen, as shown in the figure below.



In rviz, you need to set the corresponding camera topic name.

- Image_Topic: The camera topic is [/usb_cam/image_raw].
- Marker: The display component of rviz, different blocks show the location of the AR QR code.
- TF: The display component of rviz, used to display the coordinate system of the AR QR code.
- Camera: The display component of rviz, which displays the camera image.
- world: The world coordinate system.
- usb_cam: The camera coordinate system.

4. ar_track_alvar node

Subscribed topics:

Topic Name	Data Type
/camera_info	(sensor_msgs/CameraInfo)
/image_raw	(sensor_msgs/Image)

Posted topics.

Topic Name	Data Type
/visualization_marker	(visualization_msgs/Marker)
/ar_pose_marker	(ar_track_alvar/AlvarMarkers)

5. View node graph

Input following command.

```
rqt_graph
```