

Tutorial: Image Pipeline

ROS Rescue Summerschool 2012

Johannes Maurer

Institute for Software Technology

August 22, 2012, Graz



Overview

- Image Transport
- Image Pipeline
- CV Bridge

Image Transport

- specialized transport strategies
- abstracting this complexity
- separate plugin packages
- known transport packages
 - `image_transport ("raw")`
 - `compressed_image_transport ("compressed")`
 - `theora_image_transport ("theora")`
- install
 - `sudo apt-get install ros-fuerte-image-pipeline
ros-fuerte-image-transport-plugins`

Image Pipeline

- process raw camera images into useful inputs to vision algorithms
- provides tools
 - Calibration
 - Monocular processing
 - Stereo processing
 - Depth processing
 - Visualization

Image Pipeline Example

- `ROS_NAMESPACE=camera rosrun uvc_camera camera_node _frame_id:=/front_camera _camera_info_url:=file:///home/johannes/camera.yaml`
- `roslaunch image_view image_view image:=/camera/image_raw`
- `roslaunch camera_calibration cameracalibrator.py --size 8x6 --square 0.024 image:=/camera/image_raw camera:=/camera`

Image Pipeline Example

- `roslaunch camera_calibration_parsers convert ost.ini ./camera.yaml`
- `ROS_NAMESPACE=camera roslaunch image_proc image_proc`
- `roslaunch image_view image_view`
`image:=/camera/image_rect_color`

CV Bridge

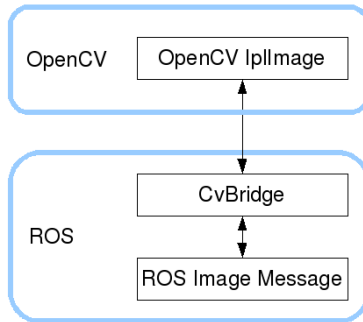


Figure: How to interface OpenCV with ROS using CvBridge

CV Bridge Example

- `rosmake find_circle`
- `roslaunch find_circle find_circle`
- `roslaunch image_view image_view`
`image:=/find_circle`

References:

- http://www.ros.org/wiki/image_transport
- http://www.ros.org/wiki/image_pipeline
- http://www.ros.org/wiki/cv_bridge
- <http://opencv.willowgarage.com>