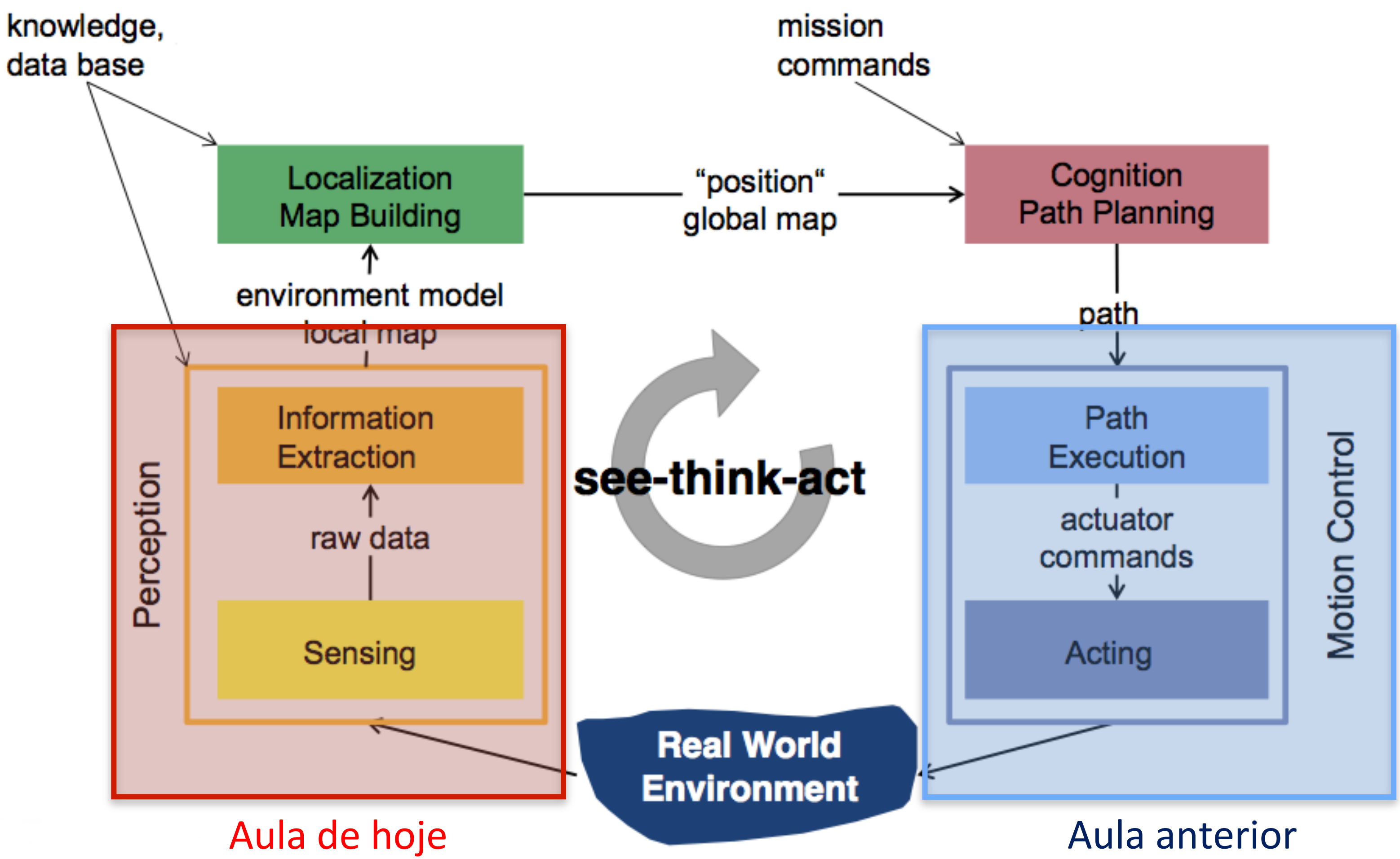
Universidade Tecnológica Federal do Paraná (UTFPR)
Disciplina: PPGCA/CPGEI - Robótica Móvel

Percepção

Prof. André Schneider de Oliveira
Prof. João Alberto Fabro

Controle de Robôs Móveis



Percepção

- Capacidade do robô sentir (ou perceber) as características do ambiente em que está inserido

- Podem ser detectadas diferentes informações
  - distância, cores, temperatura, força, altura ...

- A percepção é utilizada para que o robô interaja com o ambiente (movimente-se)

Classificação dos Sensores

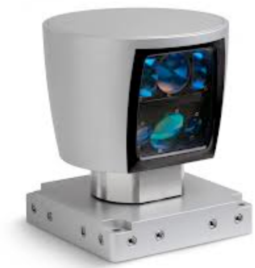
- Quanto a medição:
  - **Sensores Proprioceptivos**: realizam a medição de grandezas internas ao robô
  - **Sensores Exteroceptivos**: medem informações do ambiente

- Quanto ao funcionamento:
  - **Sensores Passivos**: medem a energia do ambiente e sofrem muita influência deste
  - **Sensores Ativos**: emitem sua própria energia e medem a reação.

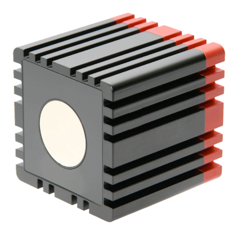
Sensores para percepção

•Sonar

•Laser Scanner ou LIDAR(Light Detection and Ranging)



•Time of Flight (TOF) Camera



•Luz Estruturada

LIDAR

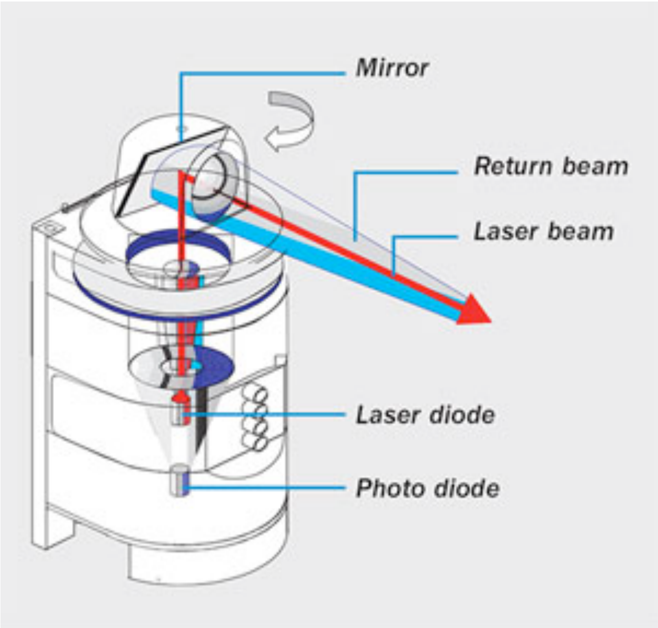
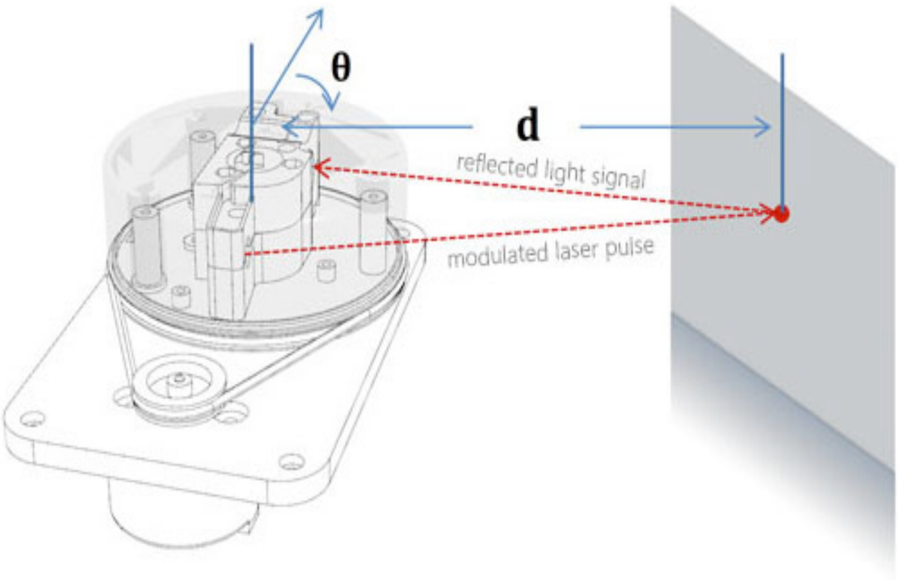
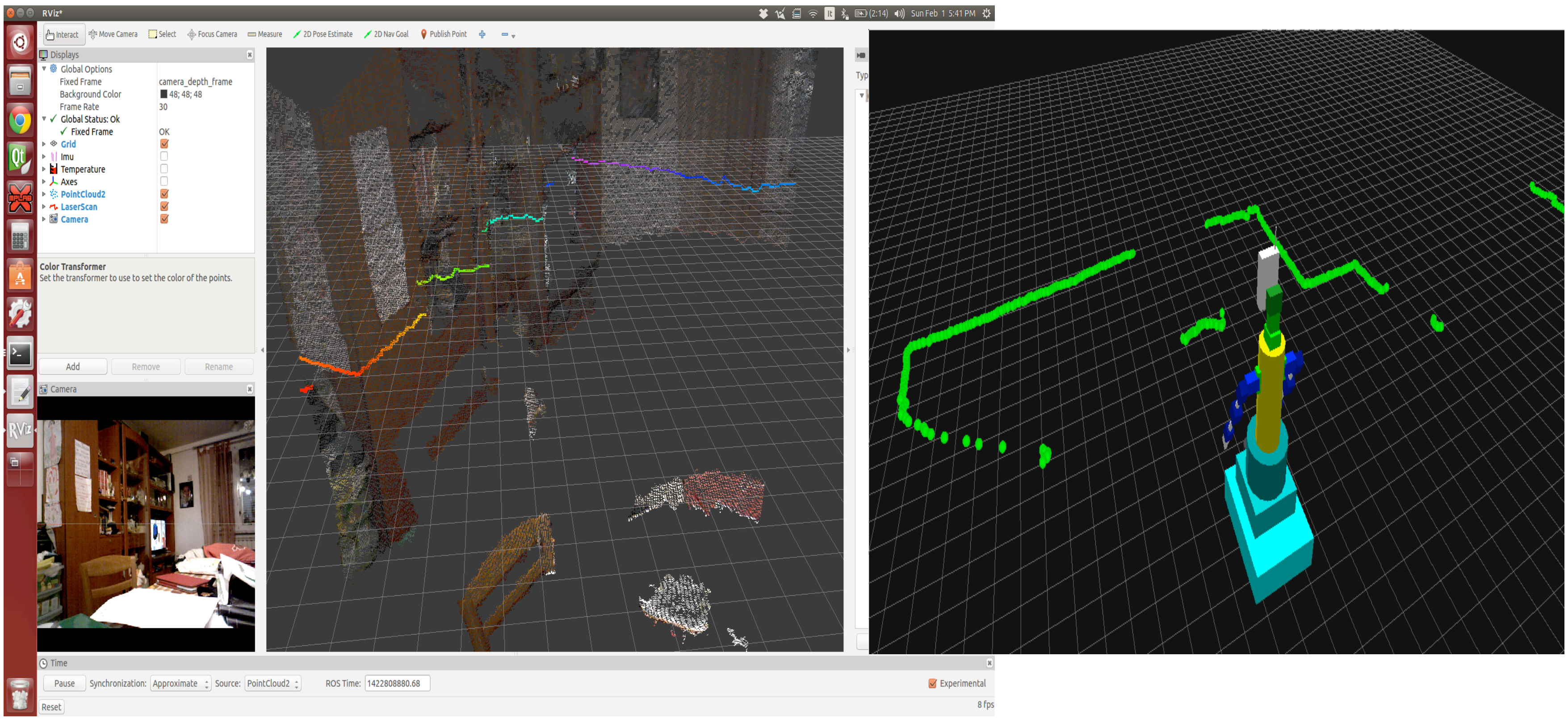


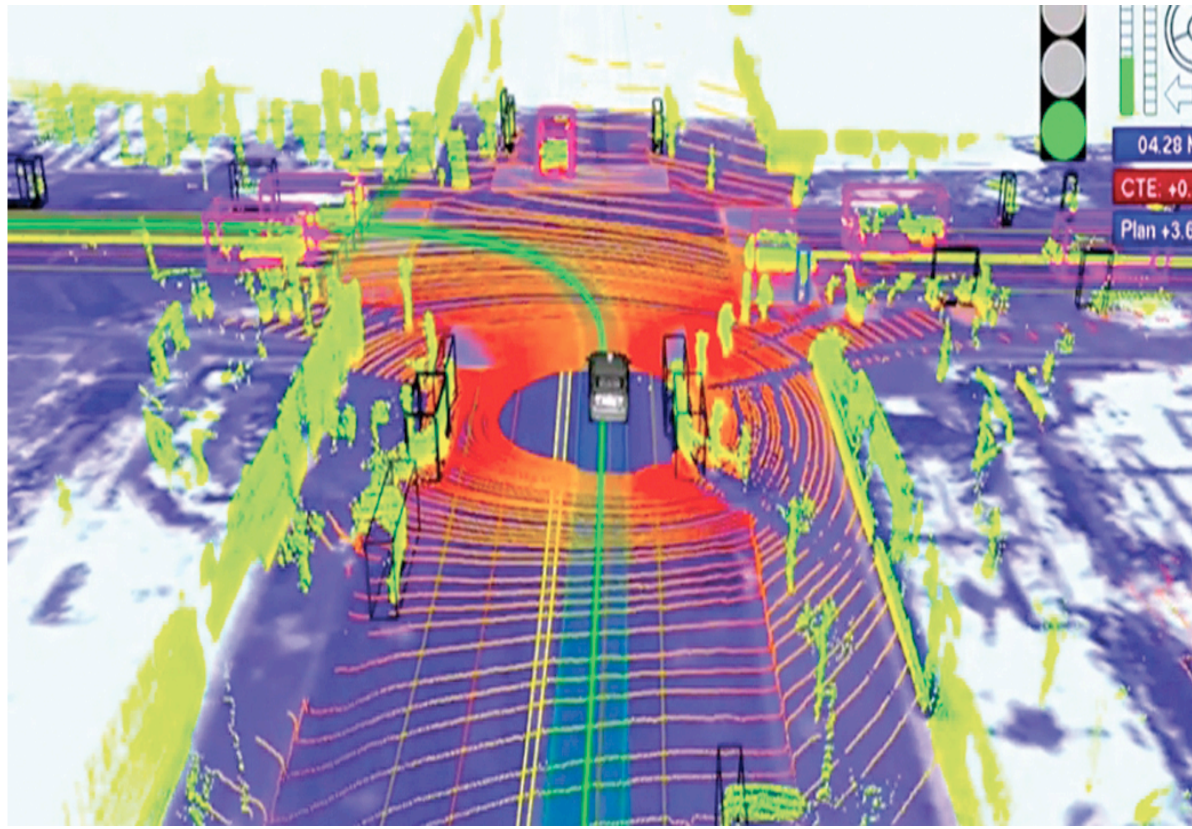
Illustration of LIDAR sensor demonstrating the time of flight principle. *(Courtesy of SICK, Inc.)*



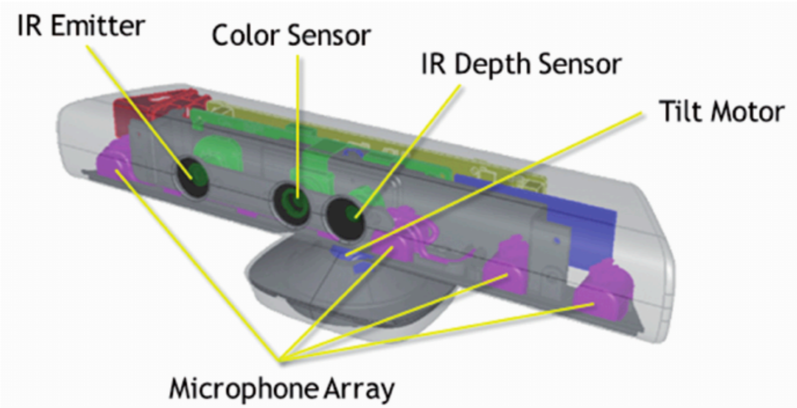
Informações Laserscan 2D

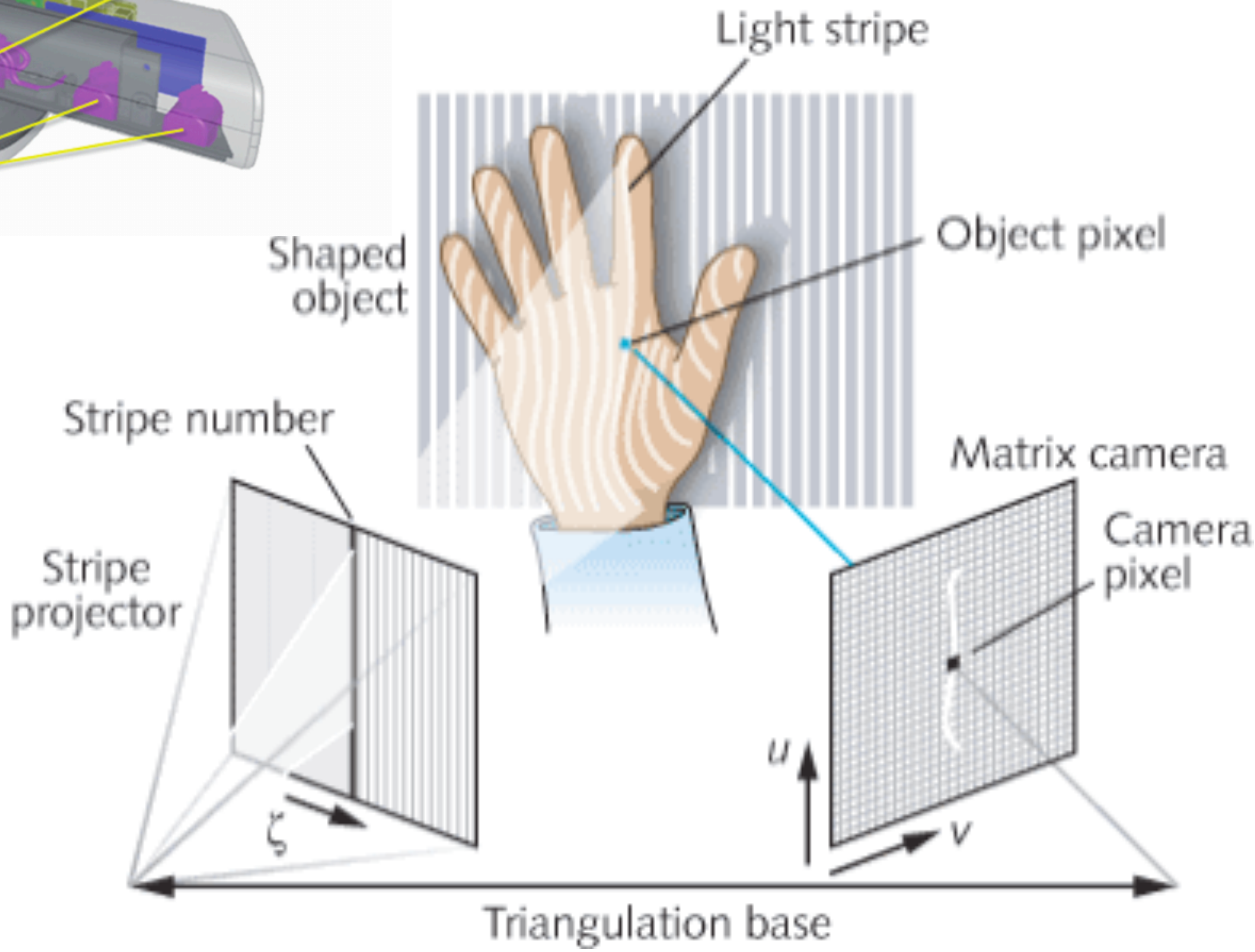


Informações Laserscan 3D

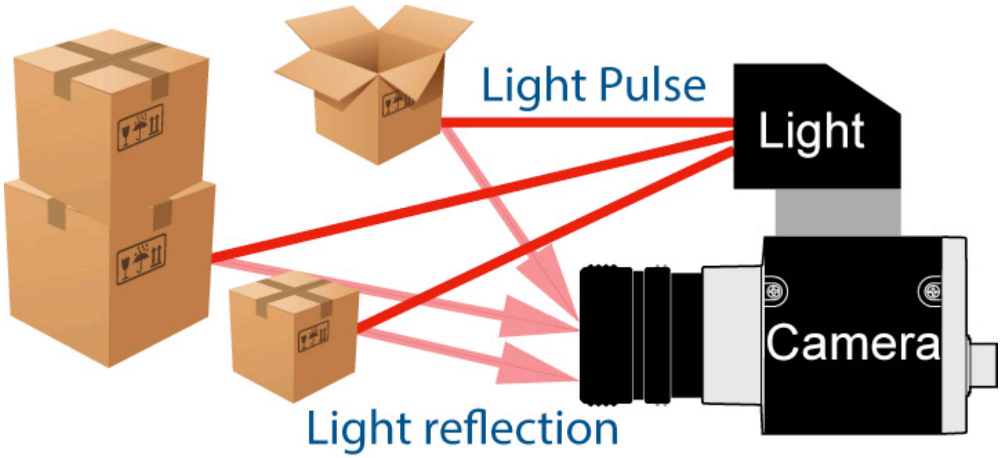


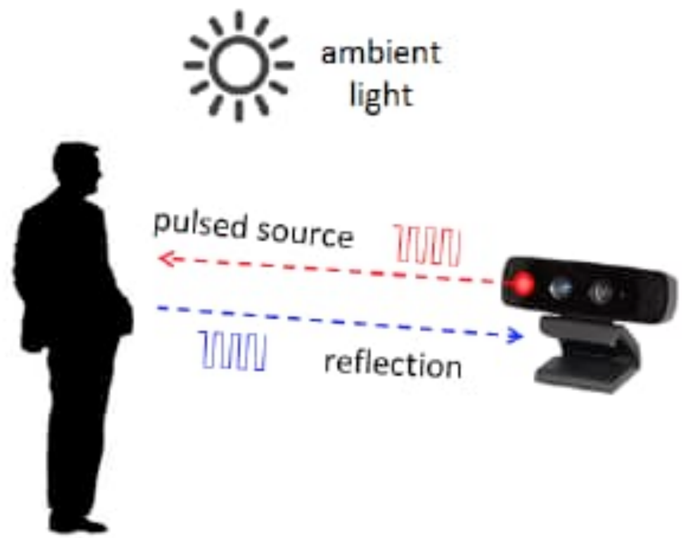
Luz Estruturada



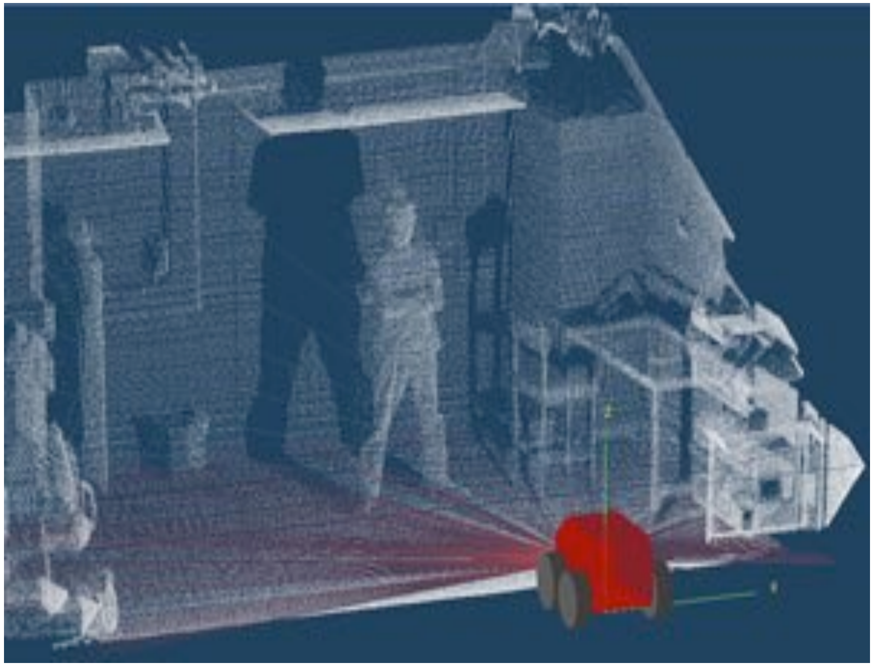


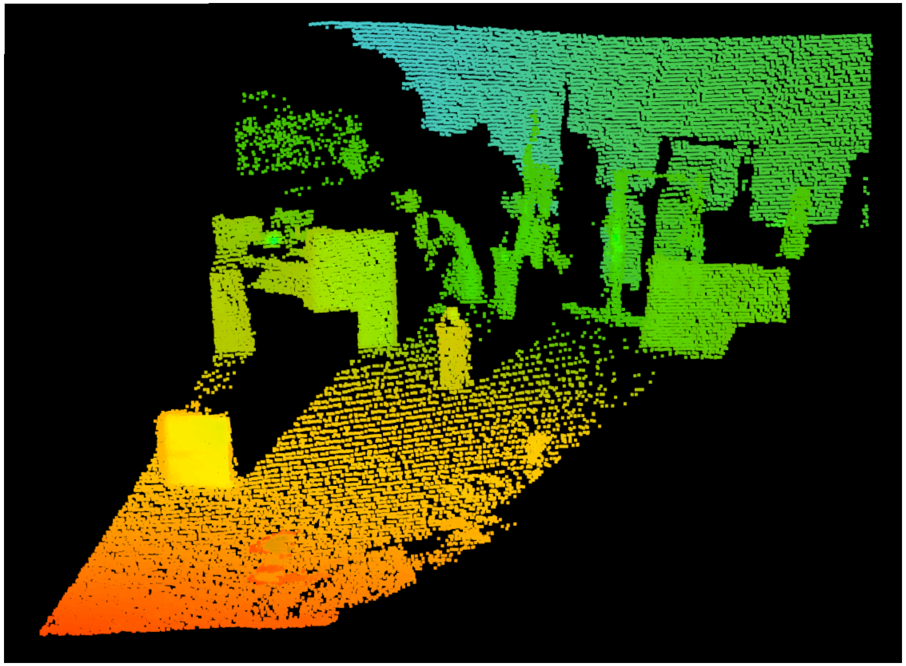
Time-of-flight



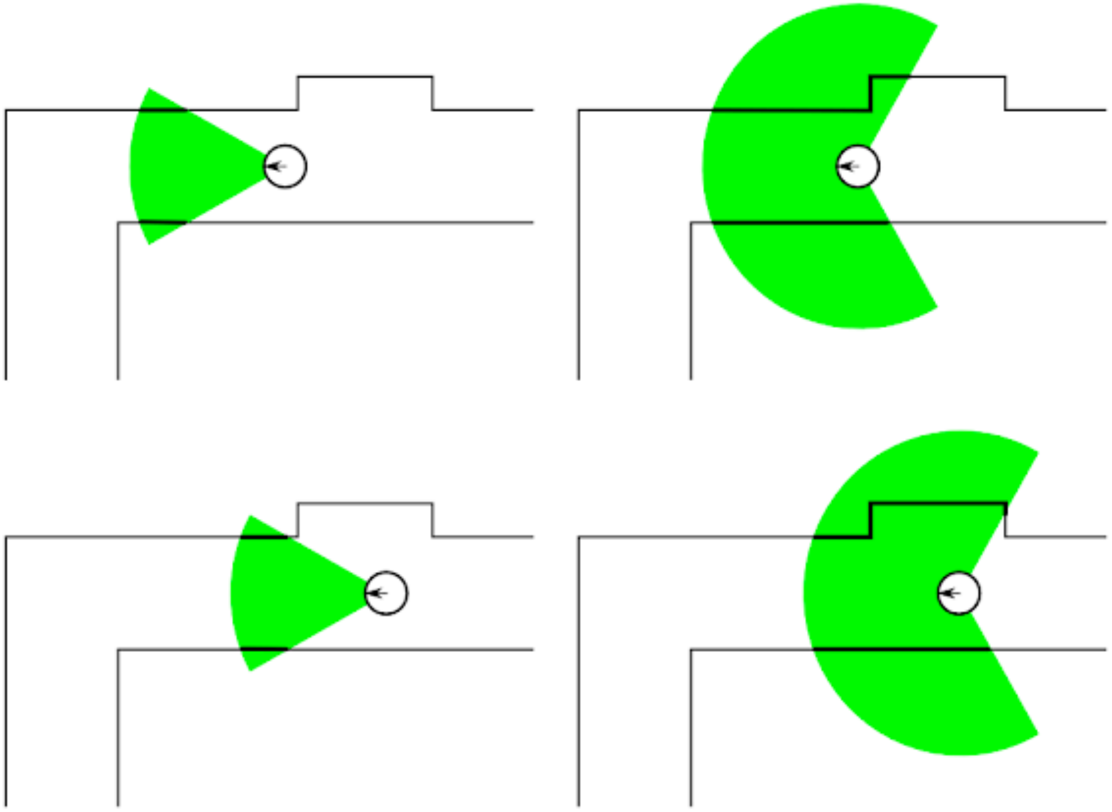


Informações Pointcloud

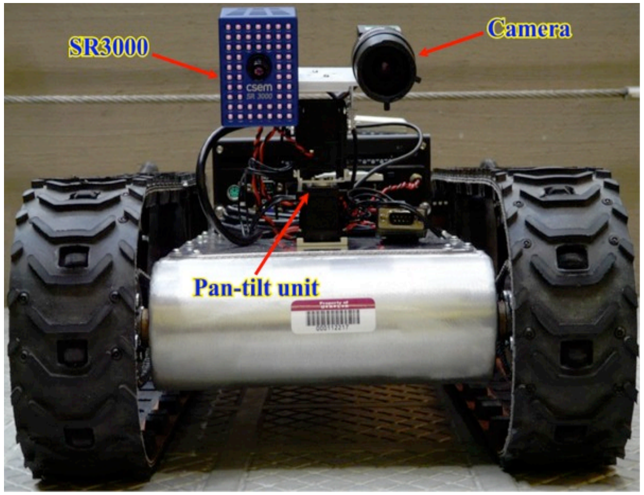




Pointcloud vs Laserscan



Exemplos de aplicação dos sensores



Laserscan

```
# Single scan from a planar laser range-finder
#
# If you have another ranging device with different behavior (e.g. a sonar
# array), please find or create a different message, since applications
# will make fairly laser-specific assumptions about this data

Header header            # timestamp in the header is the acquisition time of
                         # the first ray in the scan.
                         #
                         # in frame frame_id, angles are measured around
                         # the positive Z axis (counterclockwise, if Z is up)
                         # with zero angle being forward along the x axis

float32 angle_min        # start angle of the scan [rad]
float32 angle_max        # end angle of the scan [rad]
float32 angle_increment  # angular distance between measurements [rad]

float32 time_increment   # time between measurements [seconds] - if your scanner
                         # is moving, this will be used in interpolating position
                         # of 3d points
float32 scan_time        # time between scans [seconds]

float32 range_min        # minimum range value [m]
float32 range_max        # maximum range value [m]

float32[] ranges         # range data [m] (Note: values < range_min or > range_max should be discarded)
float32[] intensities    # intensity data [device-specific units].  If your
                         # device does not provide intensities, please leave
                         # the array empty.
```

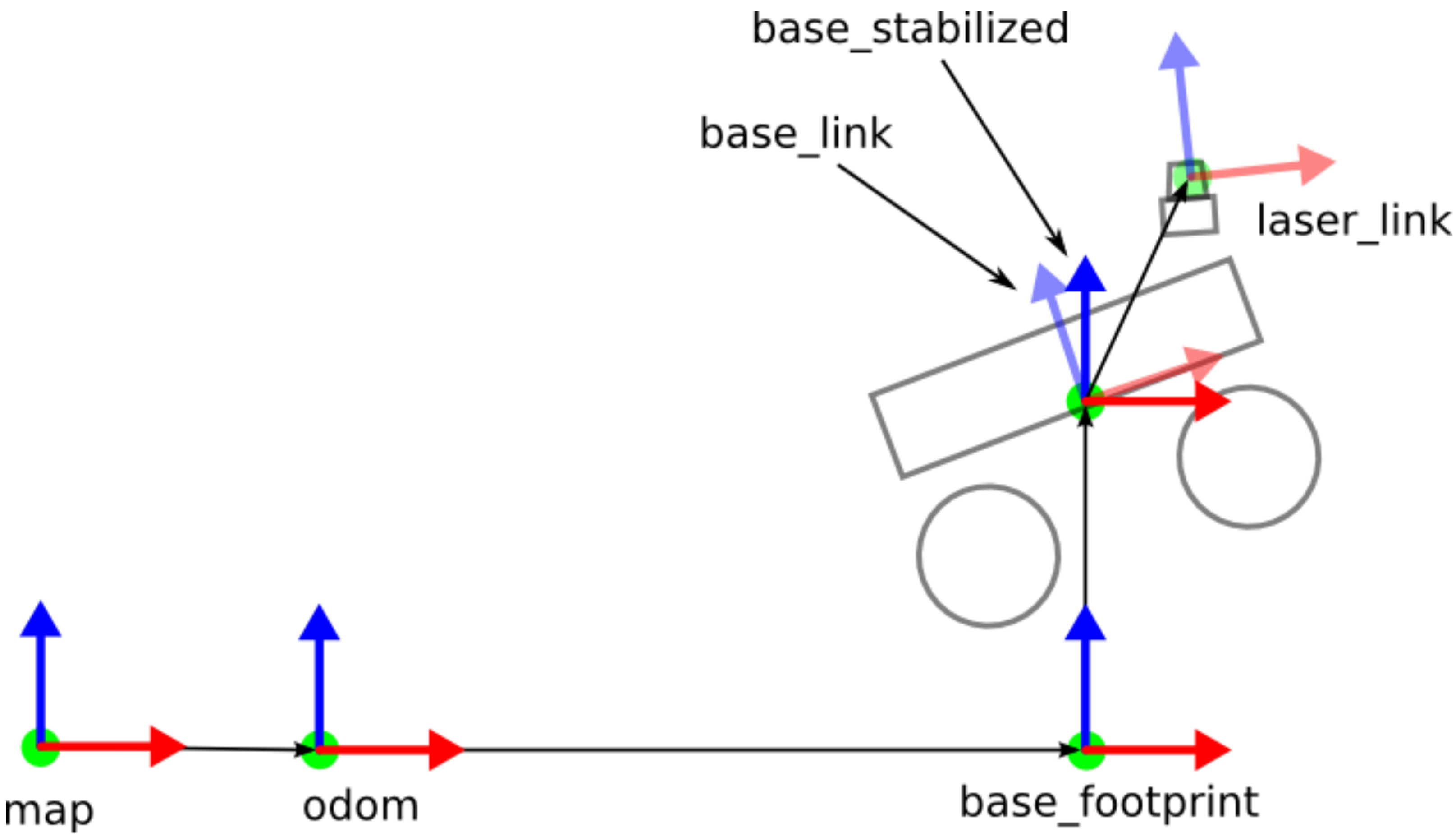
rosmsg show sensor_msgs/LaserScan

PointCloud2

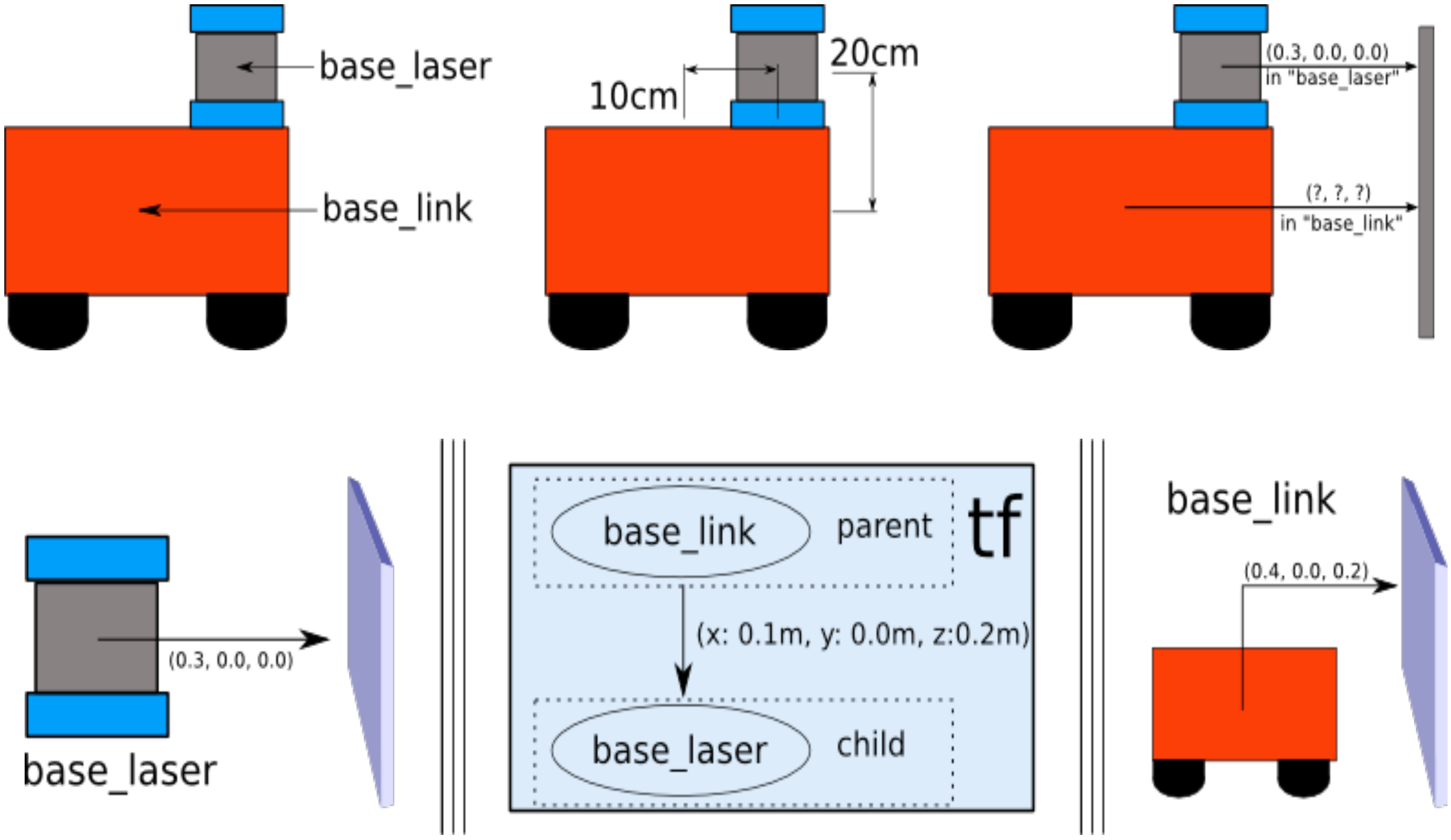
```
# This message holds a collection of N-dimensional points, which may
# contain additional information such as normals, intensity, etc. The
# point data is stored as a binary blob, its layout described by the
# contents of the "fields" array.

# The point cloud data may be organized 2d (image-like) or 1d
# (unordered). Point clouds organized as 2d images may be produced by
# camera depth sensors such as stereo or time-of-flight.

# Time of sensor data acquisition, and the coordinate frame ID (for 3d
# points).
Header header

# 2D structure of the point cloud. If the cloud is unordered, height is
# 1 and width is the length of the point cloud.
uint32 height
uint32 width

# Describes the channels and their layout in the binary data blob.
PointField[] fields

bool    is_bigendian # Is this data bigendian?
uint32  point_step   # Length of a point in bytes
uint32  row_step     # Length of a row in bytes
uint8[] data         # Actual point data, size is (row_step*height)

bool is_dense        # True if there are no invalid points
```

rosmsg show sensor_msgs/PointCloud2

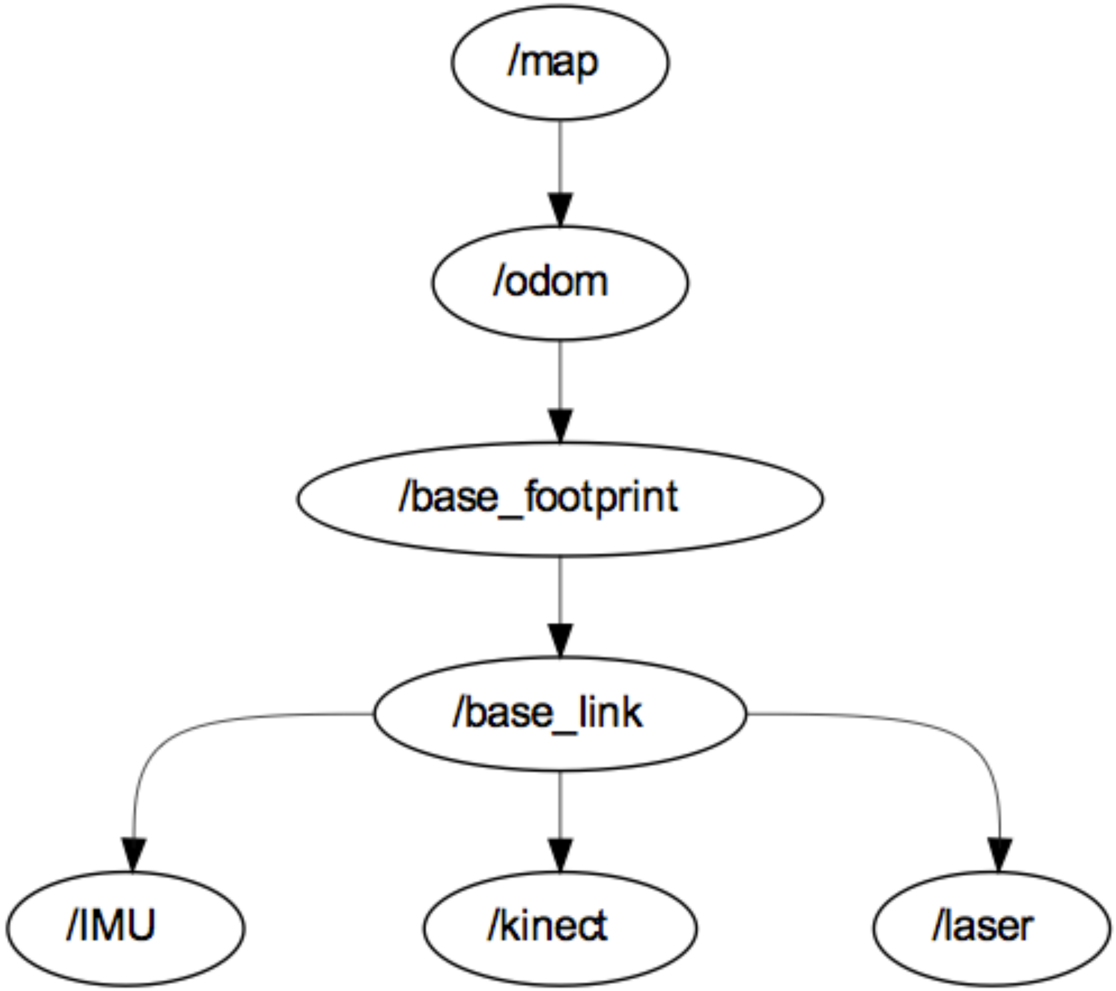
Sistemas de coordenadas



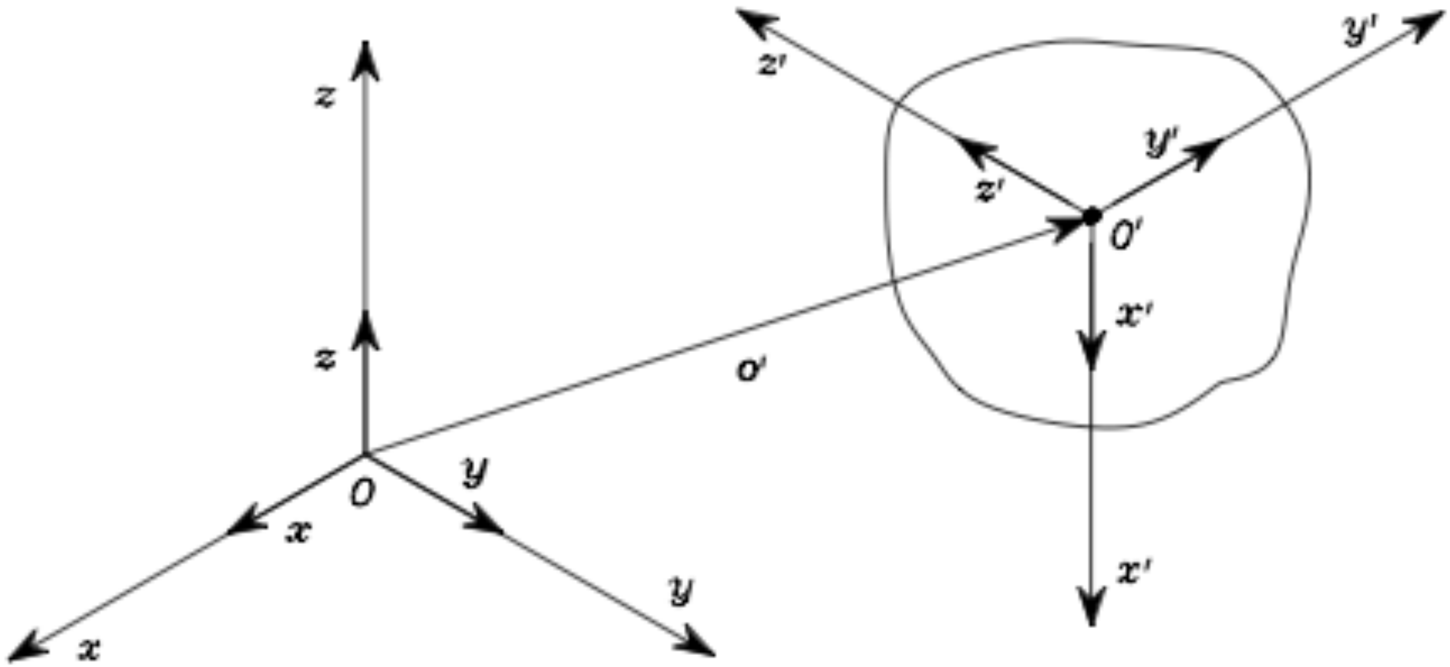
Sistemas de coordenadas



Árvore de transformações



Transformações



Transformações

Matriz de transformação homogênea

$$A_1^0 = \begin{bmatrix} R_1^0 & o_1^0 \\ 0^T & 1 \end{bmatrix}$$

Translação

$$o' = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix}$$

Rotação

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

Vetor Homogêneo

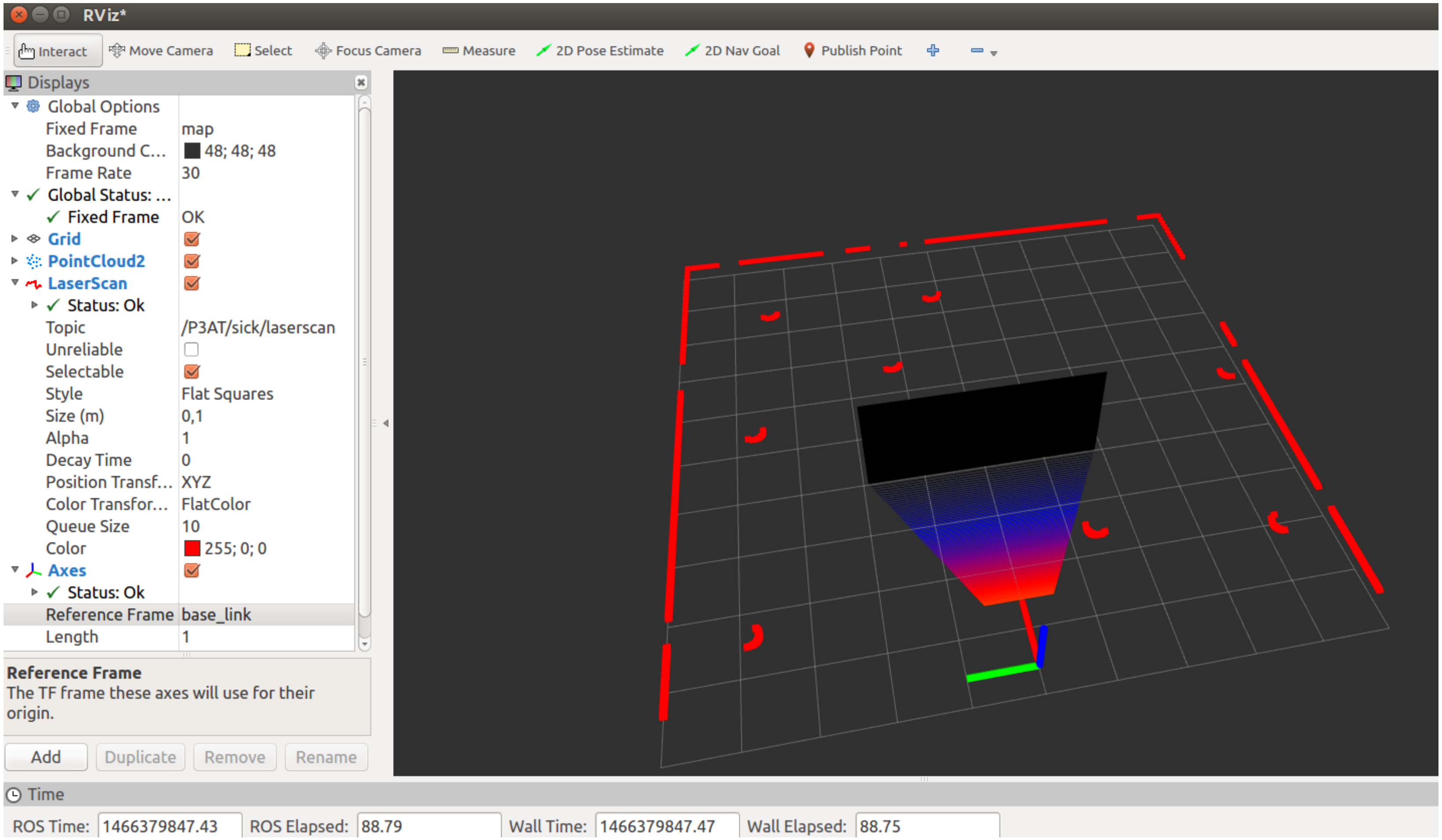
$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \qquad \tilde{p} = \begin{bmatrix} p \\ 1 \end{bmatrix}$$

Transformação

$$\tilde{p}^0 = A_1^0 \tilde{p}^1$$

$$R_2^0 = R_1^0 R_2^1.$$

Visualizador ROS - Rviz



(no terminal)
rviz