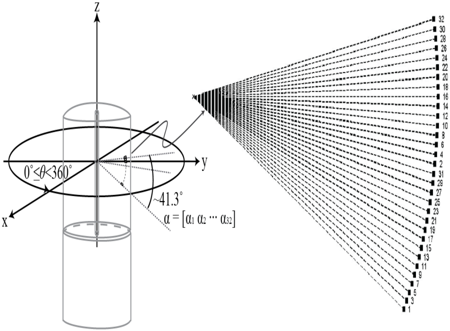
Алгоритмы сопровождения динамических целей в трехмерных облаках точек

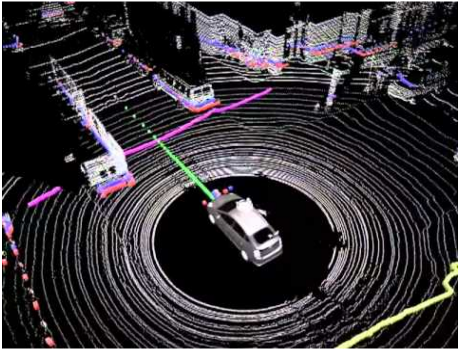
Щелчков Дмитрий

17 мая 2018 г.

Overview

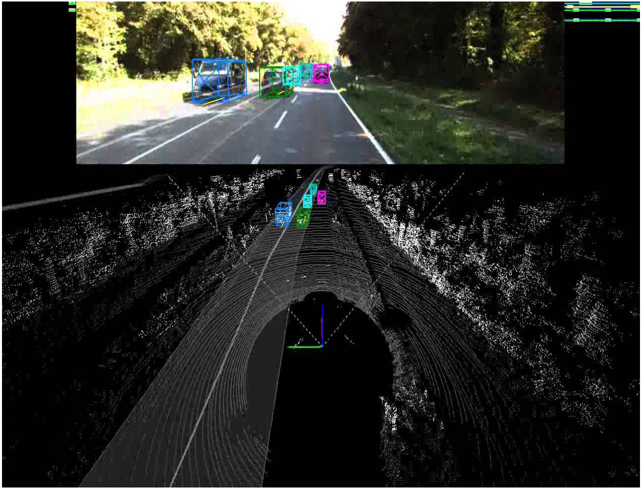
LIDAR

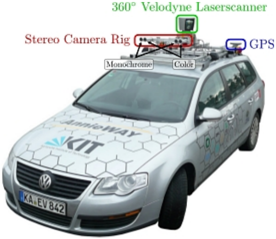




- 64 beams
- 10 Hz

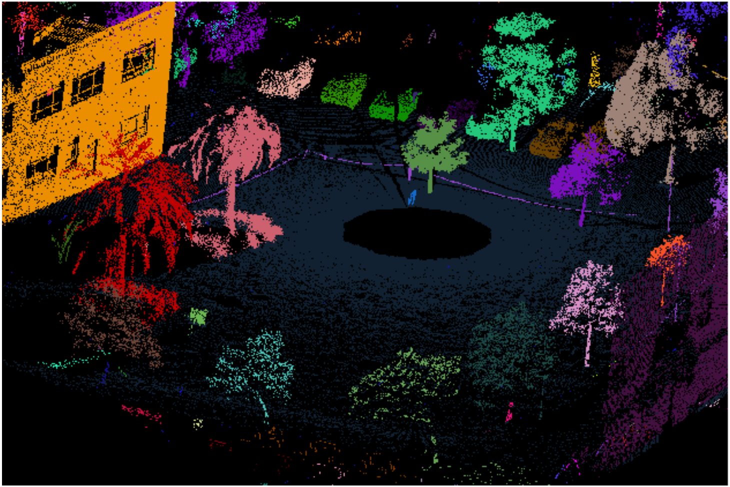
KITTI





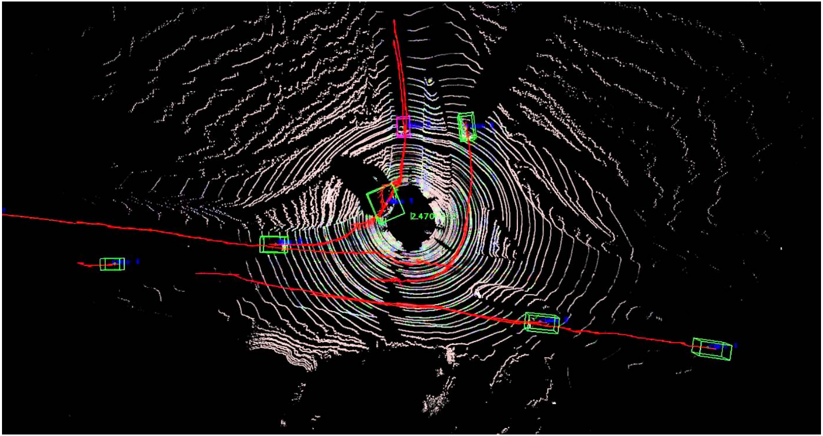
- 50 tracklets from 10 to 45 seconds each
- Bbox for each object

Задача сегментации



- Cars and people are the only important segments

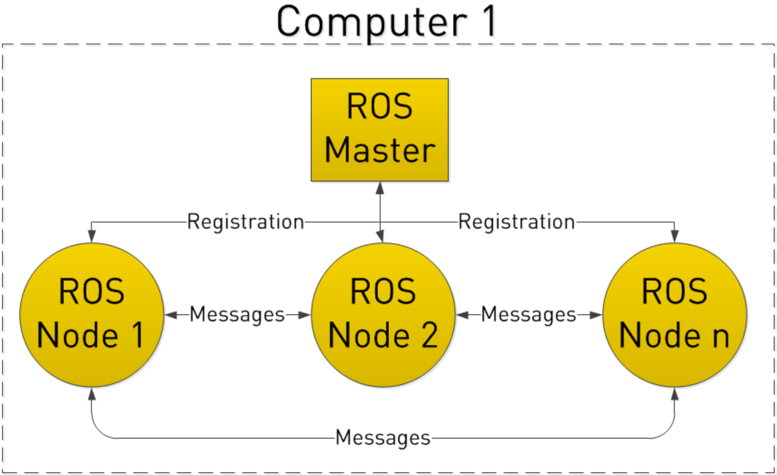
Задача детекции и трекинга



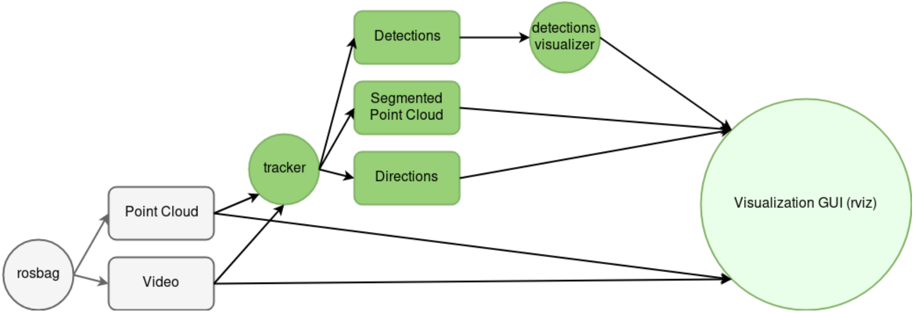
- Problems: occlusion, mismatch between frames
- One may be or may not be interested in direction, acceleration and speed

Корреляционные фильтры - пример из картинок, хочется кернел денсити овер юнион

Robot Operating System (ROS)



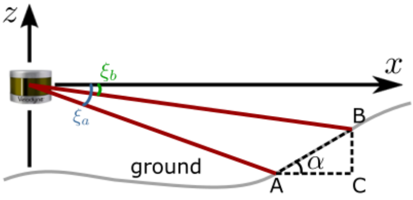
Robot Operating System (ROS): tracker graph

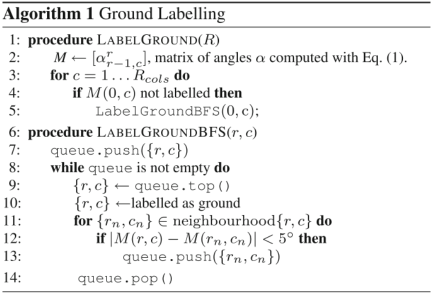


Pipeline

1. Segmentation
2. Association
3. Tracking

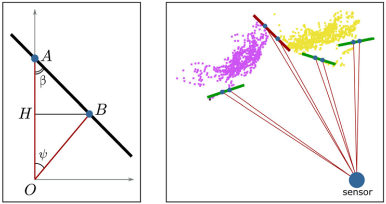
Segmentation

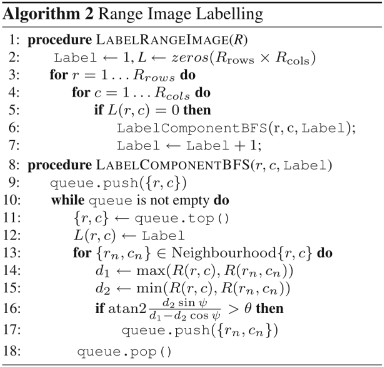




Algorithm 1 Ground Labelling

```
procedure LabelGround($R$)
    $M \leftarrow [\alpha^r_{r-1,c}]$, matrix of angles $\alpha$ computed with Eq. (1).
    for $c = 1 \ldots R_{cols}$ do
        if $M(0, c)$ not labelled then
            LabelGroundBFS(0, c);
procedure LabelGroundBFS($r, c$)
    queue.push($\{r, c\}$)
    while queue is not empty do
        $\{r, c\} \leftarrow$ queue.top()
        $\{r, c\} \leftarrow$labelled as ground
        for $\{r_n, c_n\} \in$ neighbourhood$\{r, c\}$ do
            if $|M(r, c) - M(r_n, c_n)| < 5°$ then
                queue.push($\{r_n, c_n\}$)
        queue.pop()
```

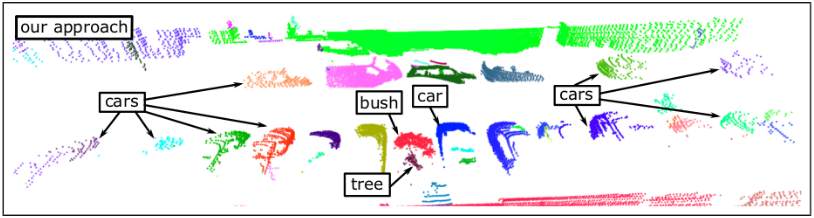




Algorithm 2 Range Image Labelling

```
procedure LabelRangeImage($R$)
    Label $\leftarrow 1, L \leftarrow zeros(R_{\text{rows}} \times R_{\text{cols}})$
    for $r = 1 \ldots R_{rows}$ do
        for $c = 1 \ldots R_{cols}$ do
            if $L(r, c) = 0$ then
                LabelComponentBFS(r, c, Label);
                Label $\leftarrow$ Label $+ 1$;
procedure LabelComponentBFS($r, c$, Label)
    queue.push($\{r, c\}$)
    while queue is not empty do
        $\{r, c\} \leftarrow$ queue.top()
        $L(r, c) \leftarrow$ Label
        for $\{r_n, c_n\} \in$ Neighbourhood$\{r, c\}$ do
            $d_1 \leftarrow \max(R(r, c), R(r_n, c_n))$
            $d_2 \leftarrow \min(R(r, c), R(r_n, c_n))$
            if $\operatorname{atan2} \frac{d_2 \sin \psi}{d_1 - d_2 \cos \psi} > \theta$ then
                queue.push($\{r_n, c_n\}$)
        queue.pop()
```

Segmentation: troubleshooting



We need only good segmentation of moving objects

There's a number of problems:

1. Undersegmentation of the ground
2. Incapability to work in the presence of plants

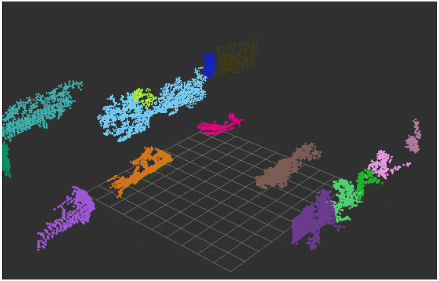
1. Normal based approach
2. Trees and grass removal algorithm

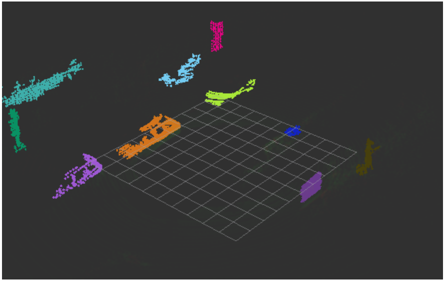
Segmentation: trees

Segmentation: trees removal

- Create local approximation of an object shape
- Check deviation of a point
- Remove points with many outliers in neighbourhood

Runtime $<$ 50ms





Segmentation: notes

- Not moving obstacles doesn't matter
- Only cars and pedestrians segmentation necessary
- Car shapes should be accurate

Трекинг - ICP + kernel correlation filter + kalman filter as baseline Нет данных чтобы обучить корреляционный фильтр (

Корреляционный фильтр на PointNet