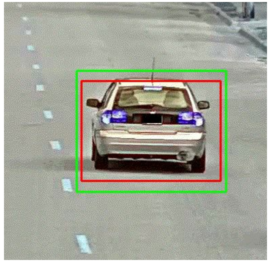
T1.1 IoU & mAP for (Team 5) - (1/3)

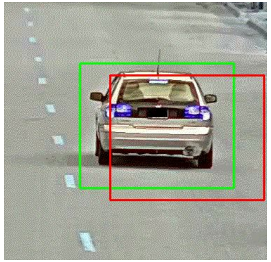
a) Alteration of the characteristics of “existing” boxes

Ground truth (GT) boxes   Altered boxes



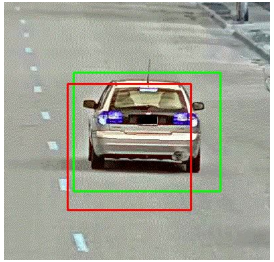
Resizing

Changing the width and height of the GT boxes



Translation

Changing the position of the GT boxes



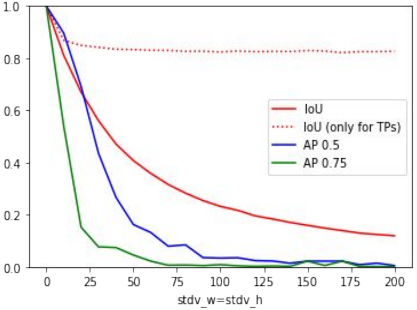
Both combined

Combining resizing with translation

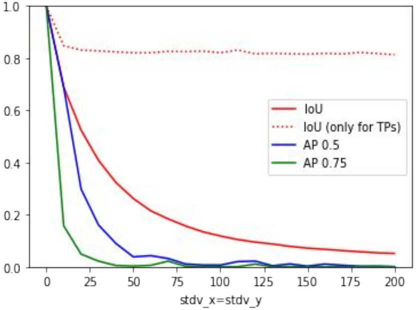
Adjustable parameters	Resizing	Translation	Both combined
	ΔWidth: Normal dist. (0, **stdv_w**) *ΔHeight*: Normal dist. (0, **stdv_h**)	*Trans. x*: Normal dist. (0, **stdv_x**) *Trans. y*: Normal dist. (0, **stdv_y**)	*All of the previous four*

Note: All standard deviation values relate to pixel units

Resizing stdv



Translation stdv



Both alterations lead to similar results:

- mIoU decreases in an exponential manner as the stdvs for resizing/displacement increase, as either the intersection between altered and GT boxes is reduced or their union grows.

- AP scores go down particularly fast when the mIoU first starts decreasing towards the threshold values of 0.75 and 0.5, as more FP appear. The $AP_{0.75}$ case suffers a more sudden decrease than the $AP_{0.5}$ due to its more restrictive threshold.

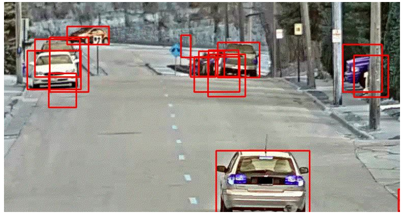
Combined	stdv_w = stdv_h	stdv_x = stdv_y	**mIoU**	**$mAP_{0.5}$**
	10 px	10 px	**0.659**	**0.600**
	15 px	5 px	**0.695**	**0.743**
	5 px	15 px	**0.596**	**0.445**

- The value of the stdvs for translation alterations appears to have a more drastic effect in lowering both the mIoU and AP.

T1.1 IoU & mAP for (Team 5) - (2/3)

b) Deleting / generating bounding boxes

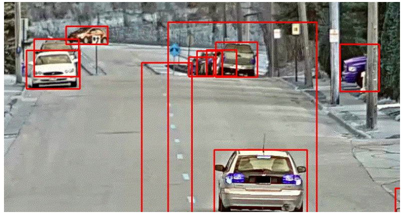
Ground truth (GT) boxes   Unaltered detections & generated boxes



Box generation (*“over”*)

Generating altered* boxes based on existing detections

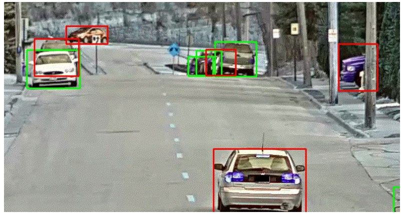
Probability: **probGen**



Box generation (*“random”*)

Generating boxes with random sizes and positioning

Probability: **probGen**

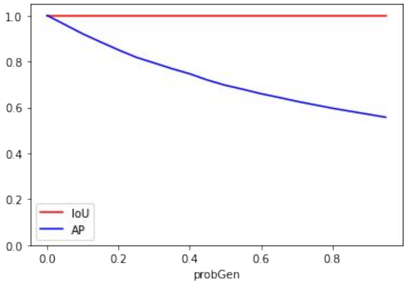


Box deletion

Deleting some of the detected bounding boxes

Probability: **probDel**

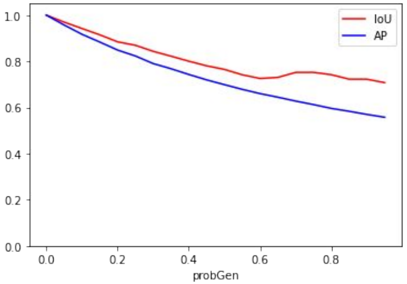
probGen (*“over”*)



- mIoU at 1 since original detects. that coincide with the GT are chosen over the new boxes.

- More FP as the prob. increases, leading to a decrease in AP until ~0.5 (50% of FPs).

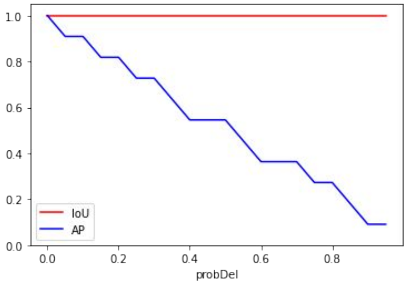
probGen (*“random”*)



- mIoU decreases as the prob. increases, since some of the new boxes get associated to GT ones before the more precise matches.

- The AP behaves the same way as in the *“over”* case.

probDel



- mIoU at 1 since all remaining detects. coincide with the GT.

- Consistent decay of AP due to number of missed detections increasing in approximately a linear manner.

(*) Altered as in the combined case of a) in the previous slide (all stdvs set to 30 px).

Note: All displayed tests considering $AP_{0.5}$

T1.1 IoU & mAP for (Team 5) - (3/3)

c) Combinations of alterations + generation/deletion of bounding boxes

	Alterations in size and position		Generation/deletion		**Measures**	
	stdv_w = stdv_h	stdv_x = stdv_y	probGen*	probDel	**mIoU**	**$\mathbf{mAP_{0.5}}$**
Ref.	10 px	10 px	0.1	0.1	**0.645**	**0.559**
	20 px	10 px	0.1	0.1	**0.567**	**0.402**
	10 px	**20 px**	0.1	0.1	**0.496**	**0.261**
	10 px	10 px	**0.4**	0.1	**0.612**	**0.465**
	10 px	10 px	0.1	**0.4**	**0.644**	**0.427**

- As previously proved, larger stdvs for the alterations in size in position lead to both mIoU and mAP values being decreased.

- While increasing the probability of generating new images kept the mIoU at 1 in the previous slide, since the reference boxes now contain alterations some of the further altered new boxes go beyond the 0.5 threshold, slightly decreasing the mIoU. The mAP suffers a more noticeable decrease due to the presence of more FPs.

- Again, increasing the probability of deleting detections maintains the mIoU intact, but leads to a decrease in the mAP due to the higher number of missed detections.

(*) Only "over" box generations considered.

Regarding the calculation of the provided measures

mIoU

Calculated taking into consideration intersection and union values for all bounding boxes in relation to corresponding GTs.

mIoU (only for TPs)

Calculated taking into consideration intersection and union values only for TP cases in relation to corresponding GTs.

$\mathbf{mAP_x}$

Calculated through the area under the precision-recall curve using the Pascal VOC with Detectron2.

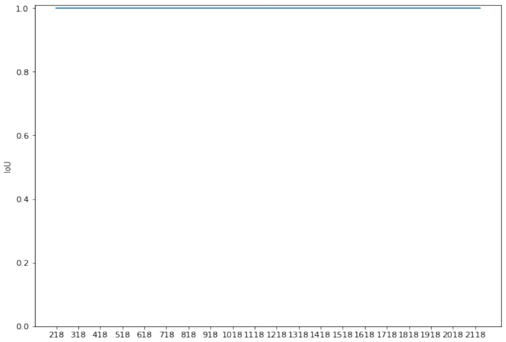
Since there is only one class being considered, the mAP is the same as the AP of the class.

Only cases with IoU larger than x are considered.

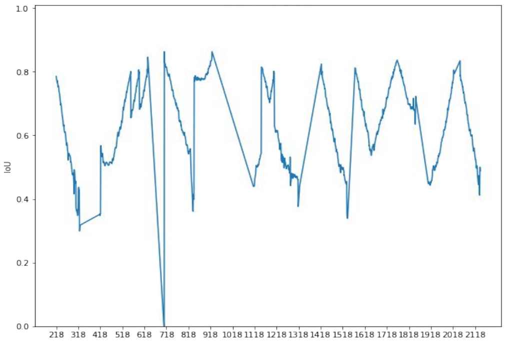
T2 IoU vs Time (Team5) - (1/2)

IoU evolution over frames for different detections

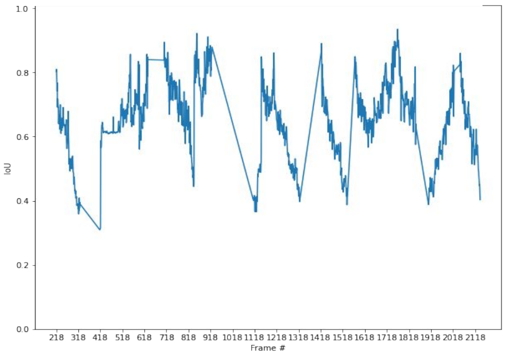
GT



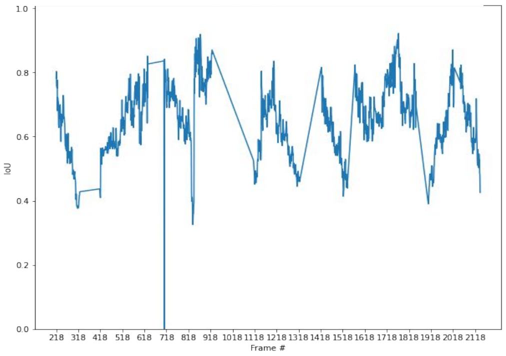
YOLO3



Mask - RCNN



SSD512



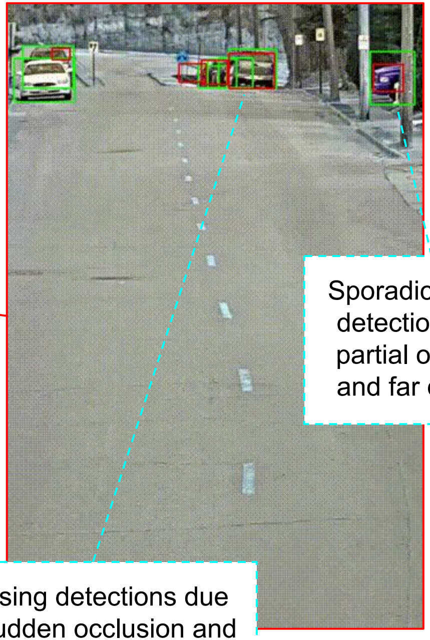
- The general progression of the IoU over the frames for each of the detection cases is quite similar.

- The range of values of all detection cases for the IoU is also rather consistent, oscillating between values of ~0.4 and 0.9.

- Mask RCNN detection appears to be the most stable alternative in terms of avoiding large drops like the one present around frames ~670 and 720.

- Similar patterns can nevertheless be observed across the different detectors, as drops in IoU caused by missed detections appear in the same frame ranges

T2 IoU vs Time (Team5) - (2/2)



Sporadic drops in detection due to partial occlusion and far distance

Missing detections due to sudden occlusion and far distance

T3.3 Analysis (Team 5) - (1/3)

Motion sequence visualisation

Sequence 45

Sequence 157

This slide aims to visualise the **motion** for both sequence 45 & 157. The resulting motion fields are likely consist mostly of **camera motions** as the result of a traveling vehicle.

A greater degree of movement can be observed in sequence 45 compared to sequence 157. We can assume it will be more difficult to estimate sequence 45 than sequence 157.

Lucas-Kanade algorithm is used for estimating the optical in this section of work. The data used is provided by **Kitti**.

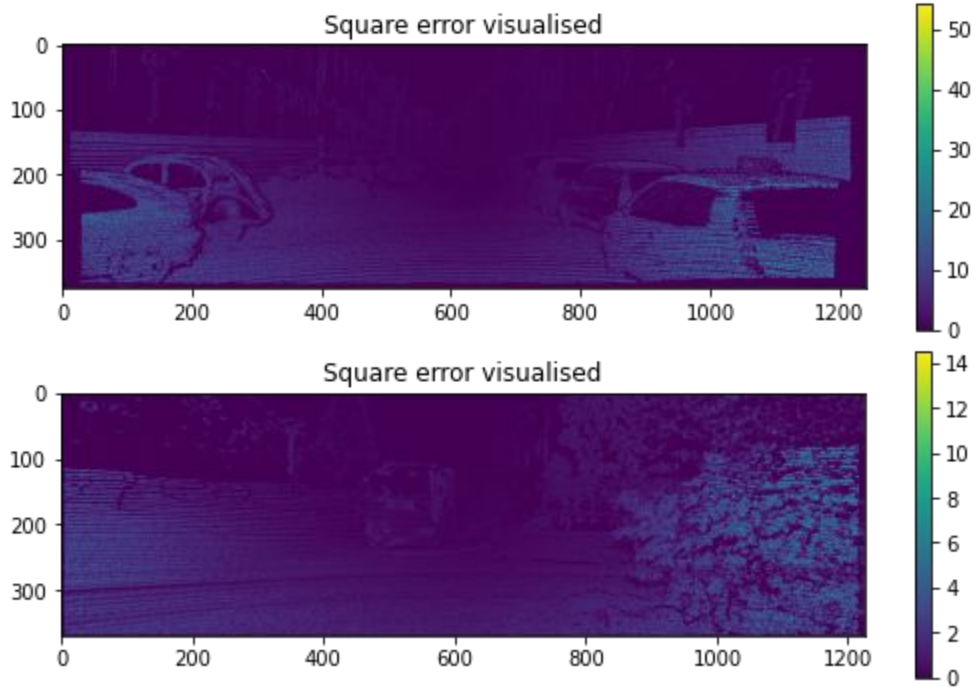
T3.3 Analysis (Team 5) - (2/3)

	MSEN	PEPN
Seq_45	10.62	78.56%
Seq_157	2.75	34.05%

Square error visualisation

Sequence 45

Sequence 157



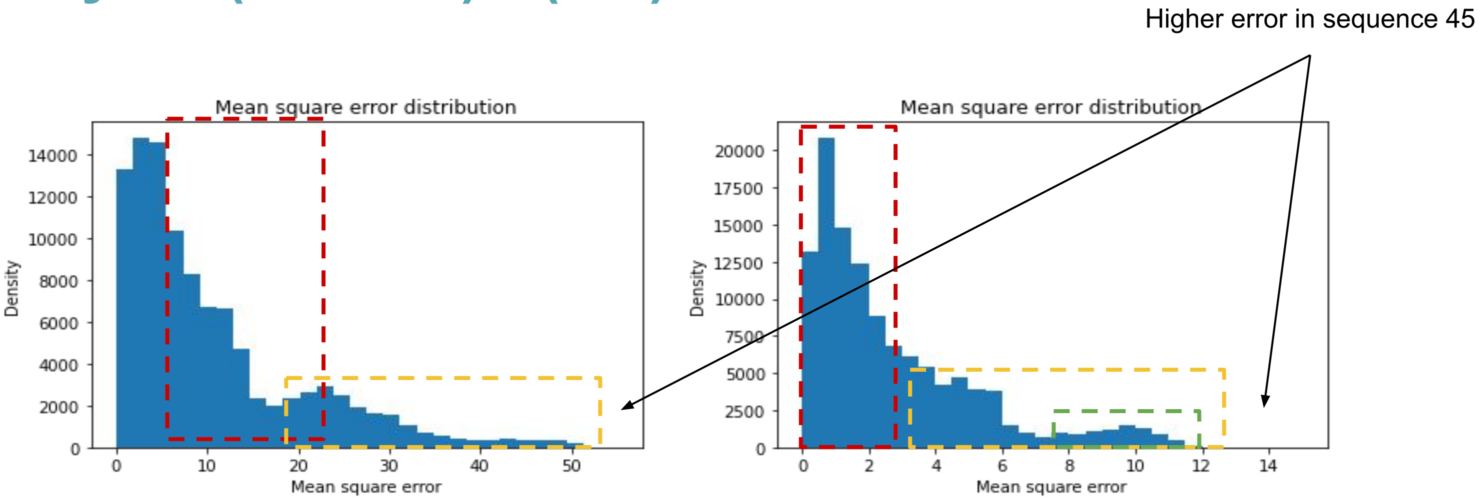
To correctly load optical flow based on Kitti dataset standard: Convert the u-/v-flow into floating point values, then subtract 2**15 before dividing the result by 64.

When calculating the mean square error and percentage of erroneous pixels in the estimation we only consider pixels in **non-occluded areas**.

Both **mean square error** in non-occluded areas and **percentage of erroneous pixels** in non-occluded areas are higher for **sequence 45**.

A **similar pattern** can be observed from both sequences, the error is higher when the pixel is further away from the center, especially around objects with high contrast.

T3.3 Analysis (Team 5) - (3/3)



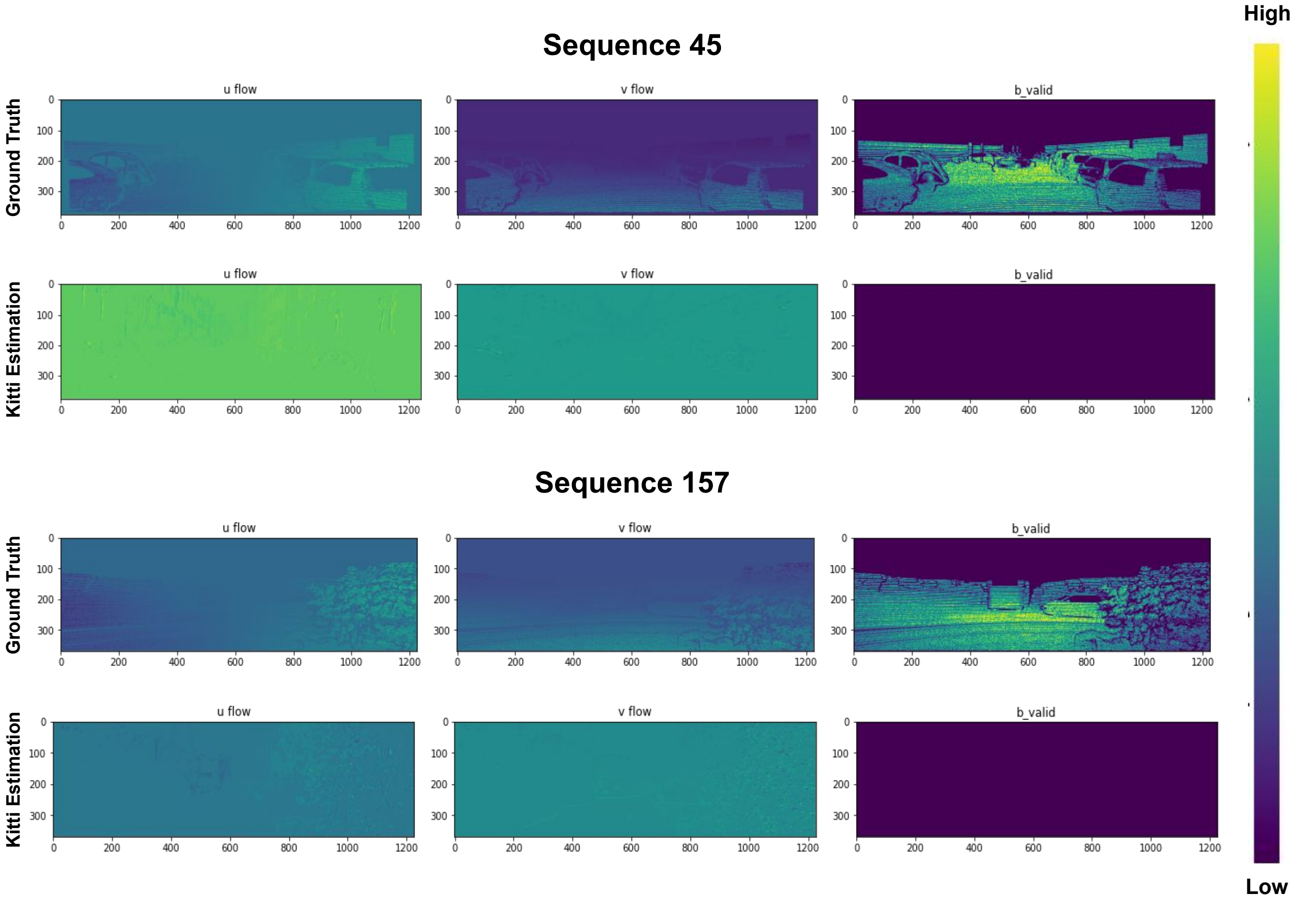
Sequence 45 **Sequence 157**

As mentioned earlier, there is a similar pattern between the two sequences and this can be further illustrated by the mean square error distribution histogram. The **red bounding boxes** indicate the error around the center of the image while the **yellow bounding boxes** show higher error near the edge of the image.

The errors within the **green bounding box** is likely the result of the bushes in sequence 157, where there are lots of textures with similar colour.

As mentioned in the lectures, **Lucas-Kanade algorithm** is prone to **camera motions**, therefore, we can assume that the higher error observed in sequence 45 is likely the result of greater camera movement.

T4 Optical flow plot (Team 5) - (1/3)

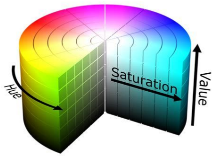


U flow and **v flow** refer to the horizontal and vertical components of the optical flow indicating motions in these directions.

Information regarding **non-occluded areas** is stored in the **b_valid** channel. This allows us to **exclude** all surface points falling outside the image plane and points occluded by objects within the same image when capturing the ground truth.

The estimated optical flow values are a lot **smaller** compared to the ground truth, and this is also true for the amount of **variation** between pixels.

T4 Optical flow plot (Team 5) - (2/3)

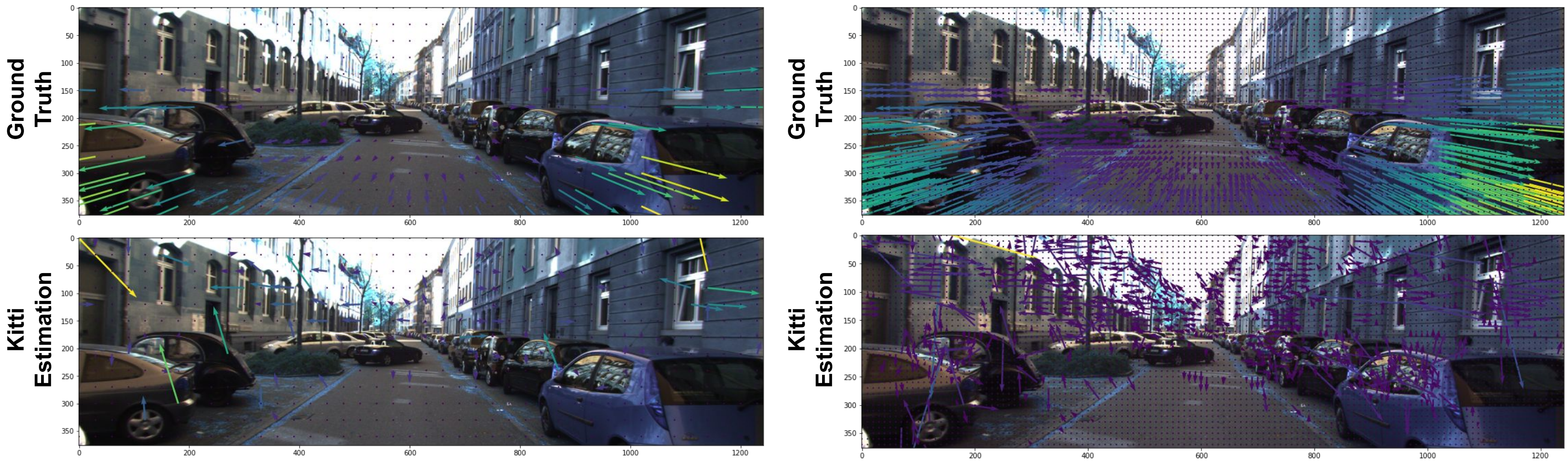


Arrows

H: Angle

S: Magnitude

Sequence 45



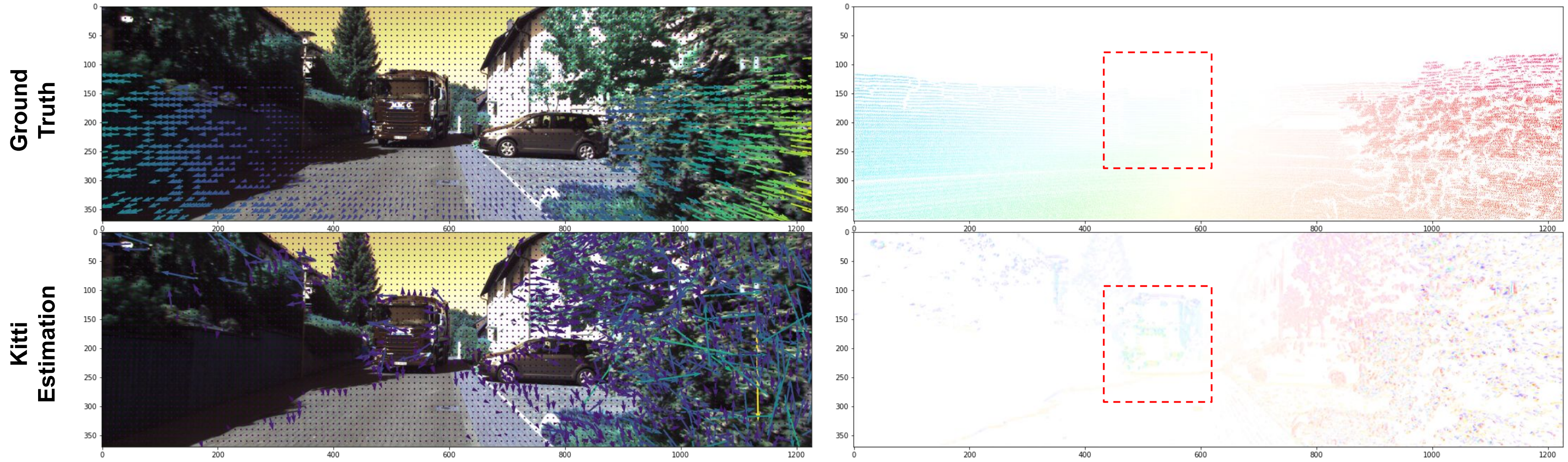
Larger grid size | **Smaller grid size**

In contrast, to the ground truth where most the motion are detected away from the center, the estimated optical flow from KL-algorithm are more evenly scattered around the image.

Two different **grid sizes** were used to visualise the optical flow. Larger grid is useful for visualizing the motion when there are lots of noise in the estimation, particularly in the case of kitti estimation.

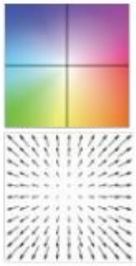
T4 Optical flow plot (Team 5) - (3/3)

Sequence 157



Using arrows to visualise the optical flow can be messy at times. Here, we experiment with the use of colour wheels with the **flow_viz** library. It is obvious that the **ground truth** has a **smooth transition of colours** showing a consistent movement of the camera. On the other hand, the colours in the estimation are scattered around indicating inconsistent movements.

It is interesting to note that in the estimation it managed to detect the movement of the incoming truck, but it was less noticeable in the ground truth.



Color wheel