

# 研究背景 / 自動車業界の動向

画像引用 : [a],[b],[c],[d]

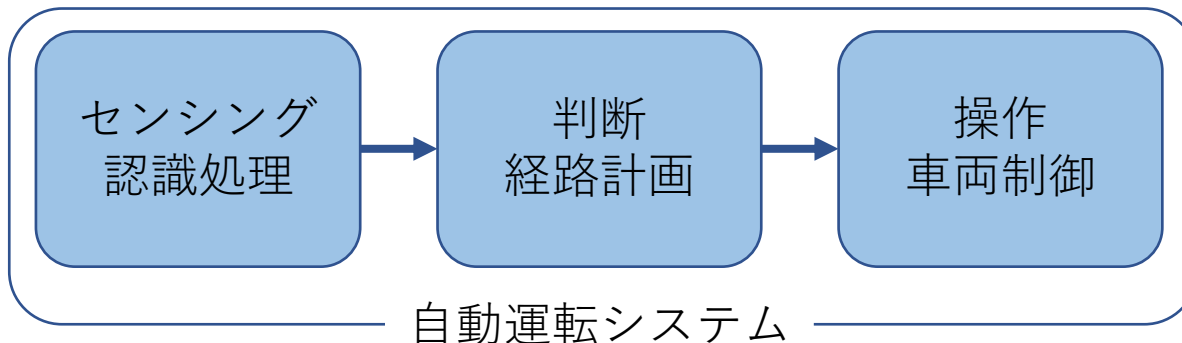
## ➤ 自動運転への注目

- 高齢化や人手不足に伴い需要増加
- 様々な企業が研究開発に取り組む



## ➤ 認識技術の重要性

- 後の判断・制御にも影響を与える
- 各種センサーの組み合わせにより実現を目指す

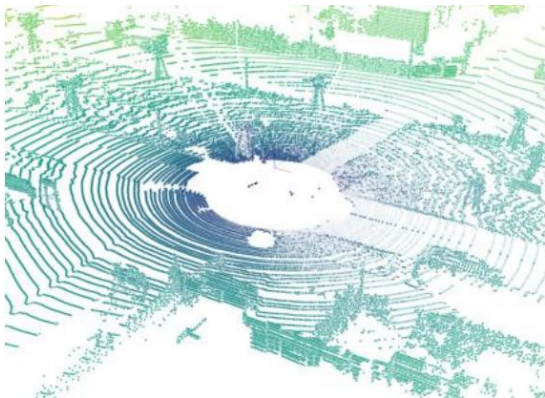
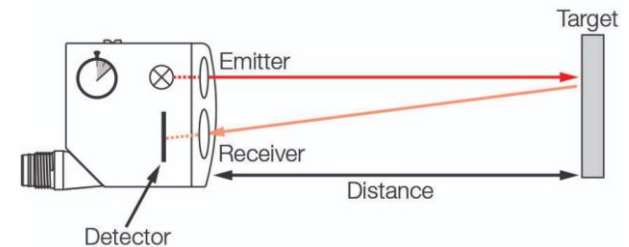


# 研究背景 / 自動車業界の動向

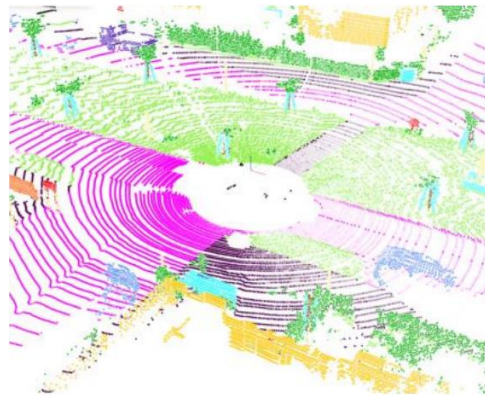
画像引用 : [e],[f],[g]

## ➤ LiDAR(Light Detection and Ranging)の活用

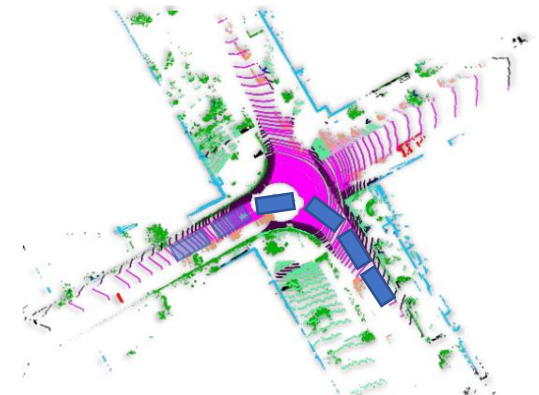
- TOF方式の測距型センサー  
(Time of Flight)
- 3次元情報を点群により直接取得
- 点群をセグメンテーションして環境を認識  
➡ 後の判断・制御につなげやすい



点群取得



セグメンテーション



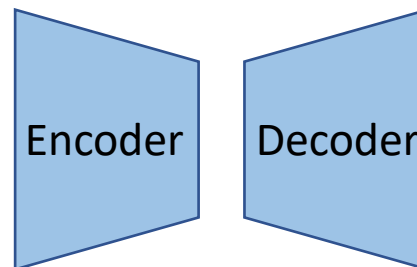
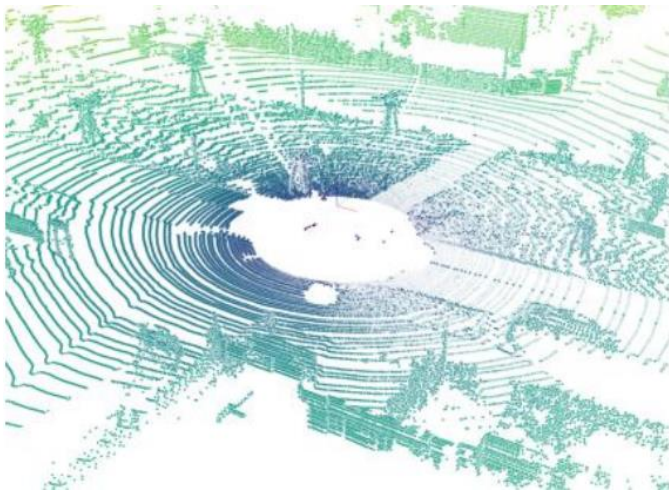
鳥瞰図投射して経路計画

# 研究背景 / 深層学習の適用

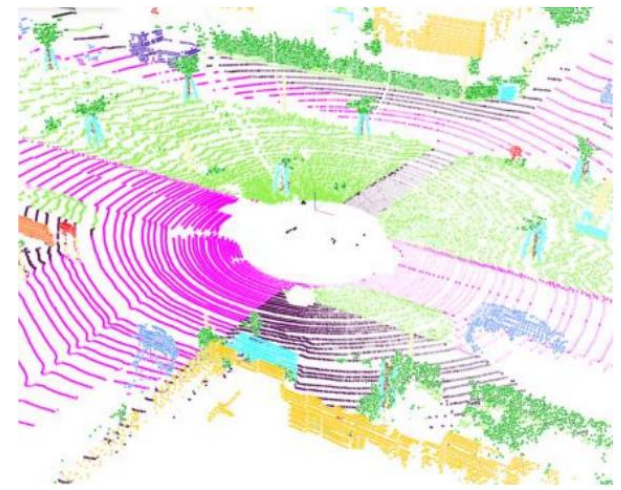
画像引用: [f]

## ▶ 深層学習の活用

- 深層学習の発展に伴い活用が広がっている
- LiDAR点群のSemantic Segmentationの研究多数
  - ➡ 点群中の各点に意味ラベルを付与して環境認識



CNNやMLPを活用



# 研究背景 / 深層学習を用いた点群処理

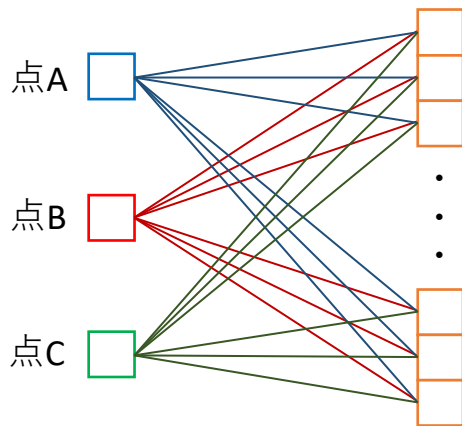
## ▶ 点群処理における条件や制約

- 点群中の各点の $(x, y, z)$ の集合として得られる
  - ➡ 点の並び順に意味を持たない(決まっていない)

単純なMLPを用いて処理する場合

(Multi-Layer Perceptron)

点群(3個の点)

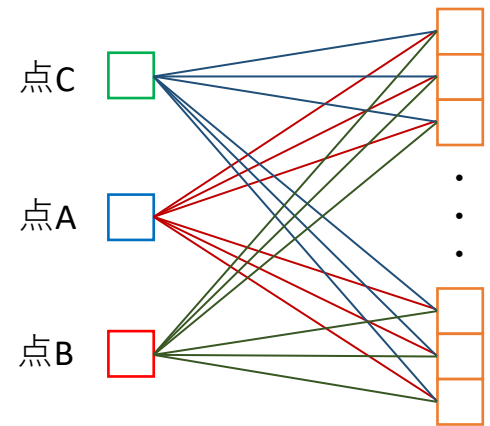


各エッジには別々の重み



点の入力順によって  
結果が変わってしまう

点群(3個の点)



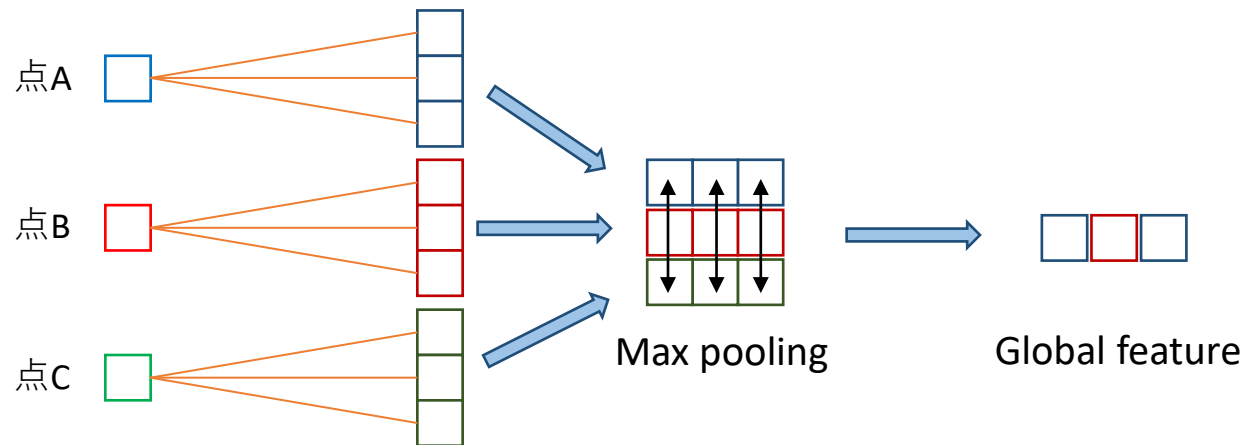
各エッジには別々の重み

# 研究背景 / 深層学習を用いた点群処理

画像引用: [1]

## ➤ Point-wise methods (PointNet based)

- 重みをシェアしたMLPで点ごとに処理
- 点方向にmax poolingをとってGlobal Featureを取得



各点に対して重みを共有

- 物体の形状情報を取得しやすい  
➡ 物体解析などに利用される

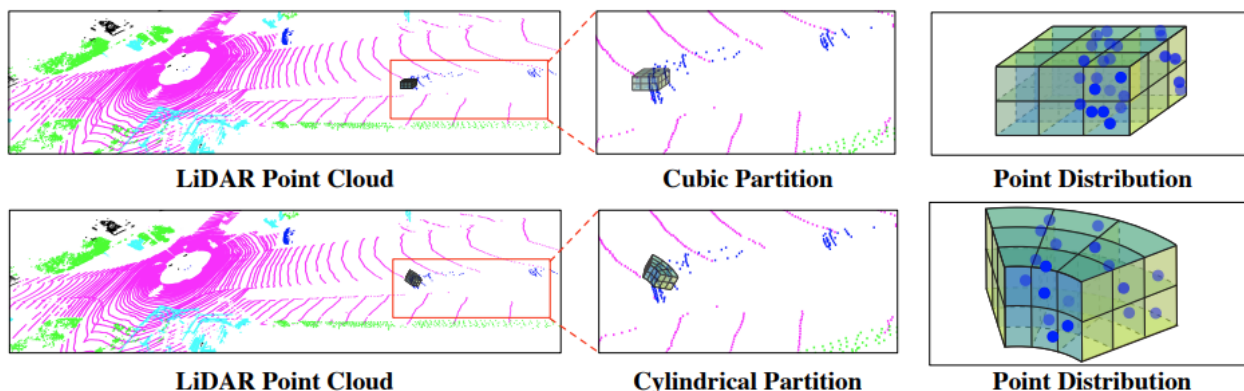


# 研究背景 / 深層学習を用いた点群処理

画像引用 : [m]

## ➤ Voxel-based methods

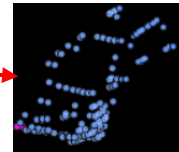
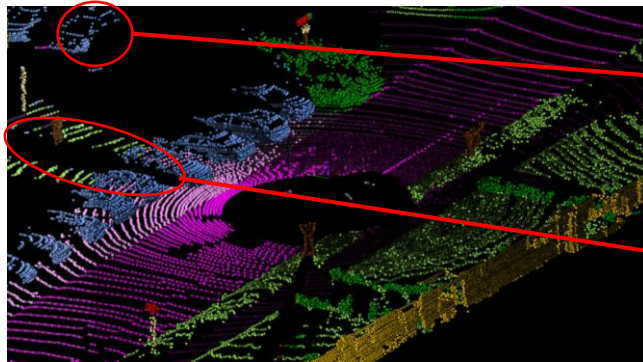
- 3次元空間をボクセルに区切る
- 各ボクセル内の点を集約してボクセル特徴にする
- 3D CNNやTransformerを用いて処理



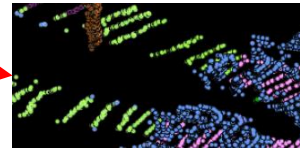
- ボクセルを細かくすればPoint-wiseな処理になる  
➡ 近傍情報と組み合わせて高精度を実現可能



## ➤ LiDARの特性による課題



遠方は点が疎になる



一部が遮られ点がない

- 場所によって点群の密度が異なる
- 3次元空間上で点が存在しない領域が多数生じる



Point-wise methodでは...

- 点の数が多く単純に計算が遅い
- 文脈情報を抽出するのが苦手

Voxel-based methodでは...

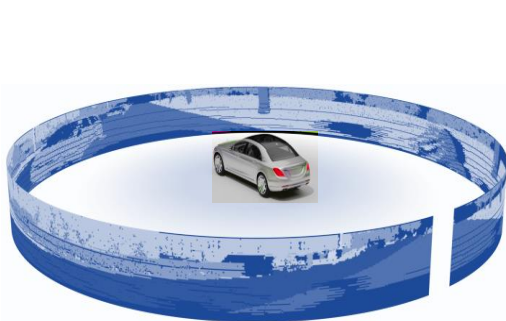
- 点が含まれないボクセルが多数  
⇨ 無駄な計算が増えて計算が遅い

# 先行研究 / RangeNet[1]

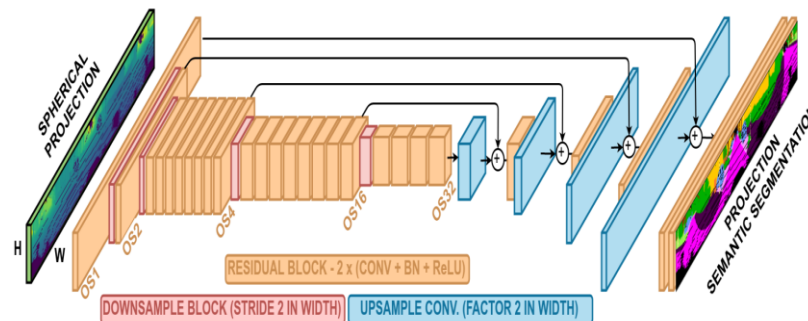
画像引用: [f],[h],[i]

## ➤ Projection-based method (RangeNet[1])

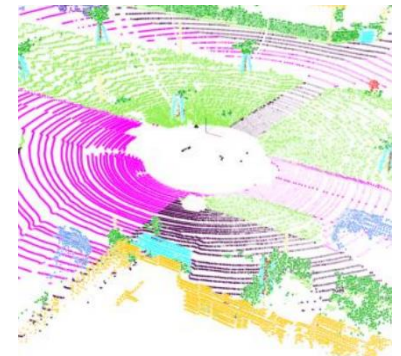
- センサーを中心とした球面に2次元投射する  
➡ 密度が一様な距離画像が得られる
- 2次元のCNN(畳み込みニューラルネット)を用いて処理
- セグメンテーション結果を3次元点群空間に再投射



球面投射



➡ 2次元上でセグメンテーション



➡ 点群区間に再投射



# 先行研究 / 利点と課題

画像引用: [h]

## ➤ 実応用上の利点

- Projection-based methodは計算効率と精度を両立  
➡ 自動運転における制約を満たすことが可能

- 限られた計算資源で高い精度が求められる
- リアルタイム処理する必要がある

## ➤ 2次元投射により生じる課題

- 同一物体間で距離によるスケールの違いが生じる



同じ"車"だがスケールが異なる

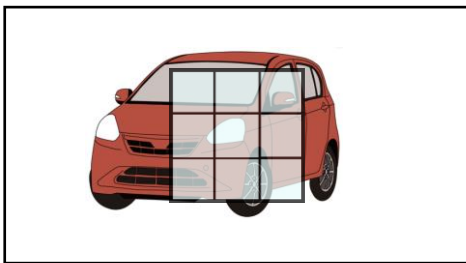
- ➡ 別々に学習されパラメータ効率が低下

# 先行研究 / 課題

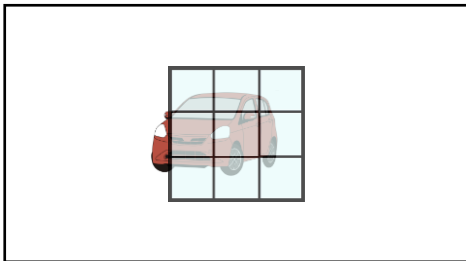
画像引用: [1]

## ➤ CNNのパラメータ効率低下

- 有効な特徴量を抽出できるフィルタを学習したい
- 学習できるフィルタの数には限りがある



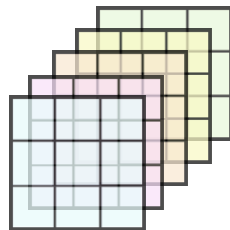
距離(近) / スケール(大)



距離(遠) / スケール(小)

スケールの違いによって、、、

- 1ピクセル当たりの実際の大きさが異なる
- 同じフィルタで得られる値が変化



ある特徴量を抽出するにあたり  
各スケールごとにフィルタを用意

# 提案手法 / モチベーション

画像引用: [1]

## ➤ モチベーション

- スケール同変性を導入してパラメータ効率向上

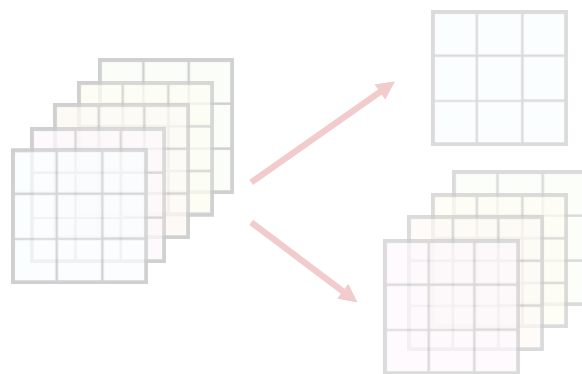
入力のスケールのみが変化したとき、  
出力も同じようにスケールのみが変化する性質



距離(近) / スケール(大)



距離(遠) / スケール(小)



1つで対応

別の特徴量抽出へ



認識精度向上の可能性

スケールに対応  
出力に回せる

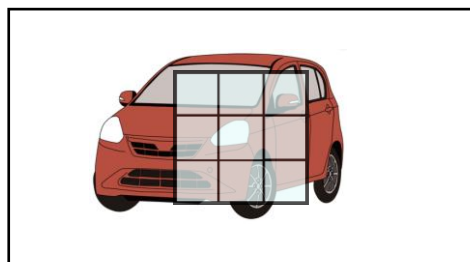
# 提案手法 / モチベーション

画像引用: [1]

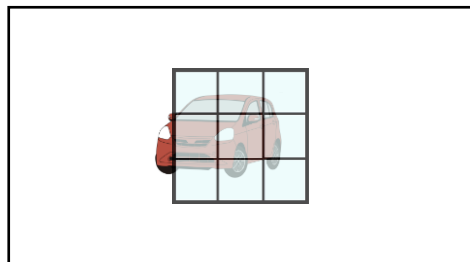
## ➤ モチベーション

- スケール同変性を導入してパラメータ効率向上

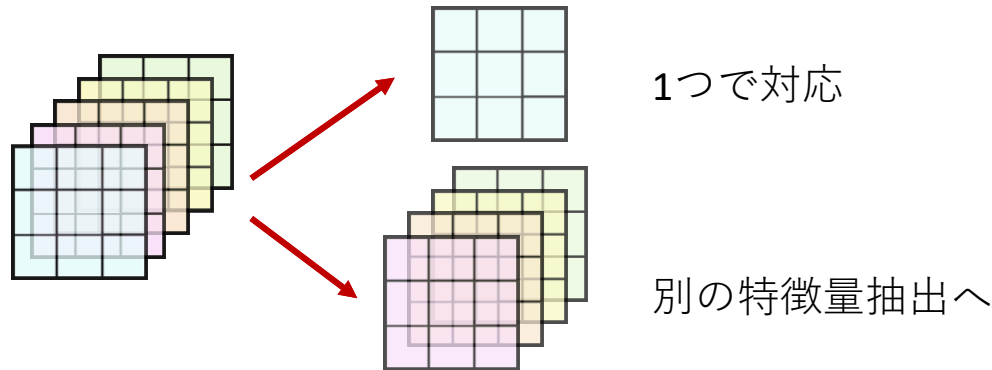
- 1つのフィルタで任意のスケールに対応
- 余った分は他の特徴量の抽出に回せる



距離(近) / スケール(大)



距離(遠) / スケール(小)



認識精度向上の可能性

# 提案手法 / 方針

## ➤ 着眼点

- スケール比 $s$ と距離値 $r$ は $s = 1/r$ の関係が成立
- スケーリングと偏微分作用素 $D_*$ の関係性

$$D_*[\underline{L_s[f]}](s\xi) = \underline{s_*} D_*[\underline{f}](\xi)$$

ただし、 $\underline{L_s[f]}(\xi) = f(s^{-1}\xi)$  (スケーリング関数)  
 $x, y$ 方向それぞれの微分階数 $i_*, j_*$ に対して  $\underline{s_*} = 1/s^{i_*+j_*}$   
 $* \in \{0, x, y, xx, xy, yy\} = \mathcal{D}$



## ➤ 方針

- 偏微分作用素を用いた畳み込みフィルタを定義
- 距離値に応じた重み付けで $s_*$ を打ち消すよう設計



# 提案手法 / 概要

## ➤ 従来の畳み込み

- フィルタをピクセルごとに学習

## ➤ 距離同変畳み込み(REconv)

- 偏微分作用素をフィルタ  $D_*^d$  で近似
- 畳み込みフィルタを  $D_*^d$  の線形結合で定義

$$\begin{array}{c} \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} \\ \text{畳み込みフィルタ} \\ D^d \end{array} = \beta_1 \begin{array}{c} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ D_0^d \end{array} + \beta_2 \frac{1}{\Delta x} \begin{array}{c} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} \\ D_x^d \end{array} + \dots + \beta_9 \frac{1}{(\Delta x)^2(\Delta y)^2} \begin{array}{c} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \\ D_{xxyy}^d \end{array}$$

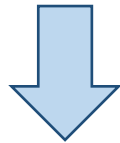
➡ 係数パラメータ  $\boldsymbol{\beta} = \{\beta_1, \beta_2, \dots, \beta_9\}$  を学習する

# 提案手法 / 概要

## ➤ 距離同変畳み込み(REconv)

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \beta_1 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \beta_2 \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} + \dots + \beta_9 \frac{1}{(\Delta x)^2(\Delta y)^2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

畳み込みフィルタ  $D^d$        $D_0^d$        $D_x^d$        $D_{xxyy}^d$



距離  $r$  と微分階数に応じた重みをかける  
 $r_* = \frac{k_x^{i_*} k_y^{j_*}}{r^{i_*+j_*}} \quad (r_* s_* = 1 \text{ が成立})$

$$\begin{bmatrix} w_1' & w_2' & w_3' \\ w_4' & w_5' & w_6' \\ w_7' & w_8' & w_9' \end{bmatrix} = \beta_1 r_0 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \beta_2 r_x \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} + \dots + \beta_9 r_{xxyy} \frac{1}{(\Delta x)^2(\Delta y)^2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

畳み込みフィルタ  $D_r^d$        $D_0^r$        $D_x^r$        $D_{xxyy}^r$

- スケールに依らず同じ特徴量が得られる

$$D_*^r [L_s[f]](s\xi) = \frac{r_* s_*}{=1} D_*^d [f](\xi) = D_*^r [f](\xi)$$

# 提案手法 / 概要

## ➤ 距離同変畳み込み(REconv)

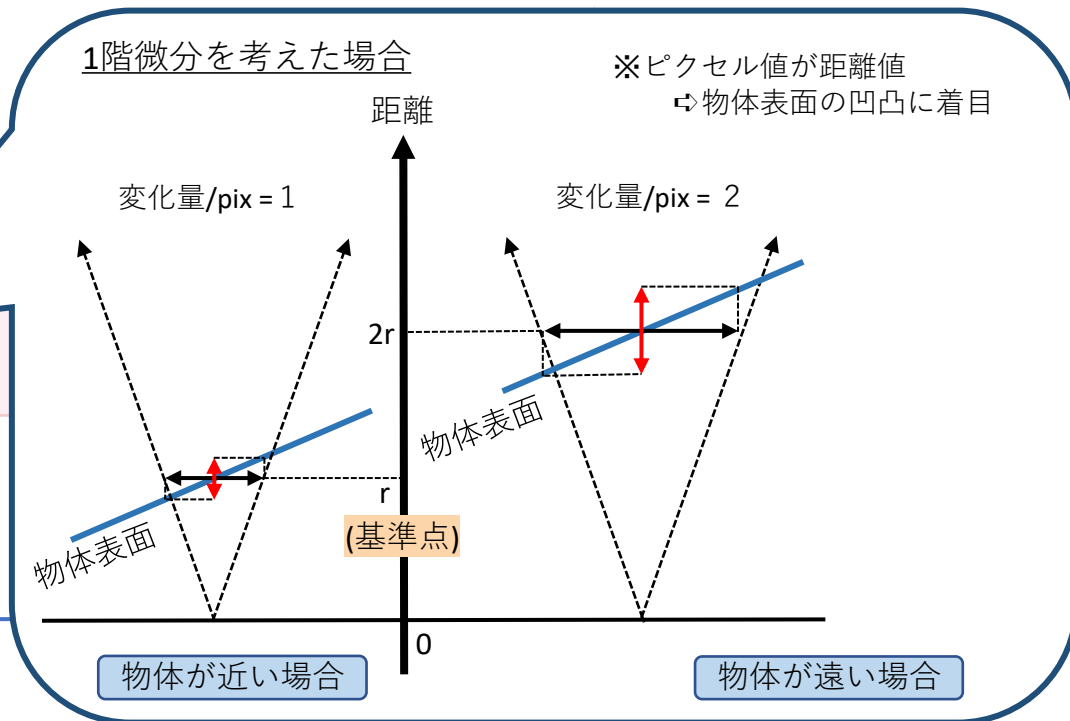
$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \dots + \beta_2 \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix}$$

畳み込みフィルタ  $D^d$

$$\begin{bmatrix} w_1' & w_2' & w_3' \\ w_4' & w_5' & w_6' \\ w_7' & w_8' & w_9' \end{bmatrix} = \dots + \beta_2 r_x \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix}$$

畳み込みフィルタ  $D^d$

$D_x^r$



- スケールに依らず同じ特徴量が得られる

$$D_*^r [L_s[f]](s\xi) = \frac{r_* s_*}{=1} D_*^d [f](\xi) = D_*^r [f](\xi)$$

# 提案手法 / 概要

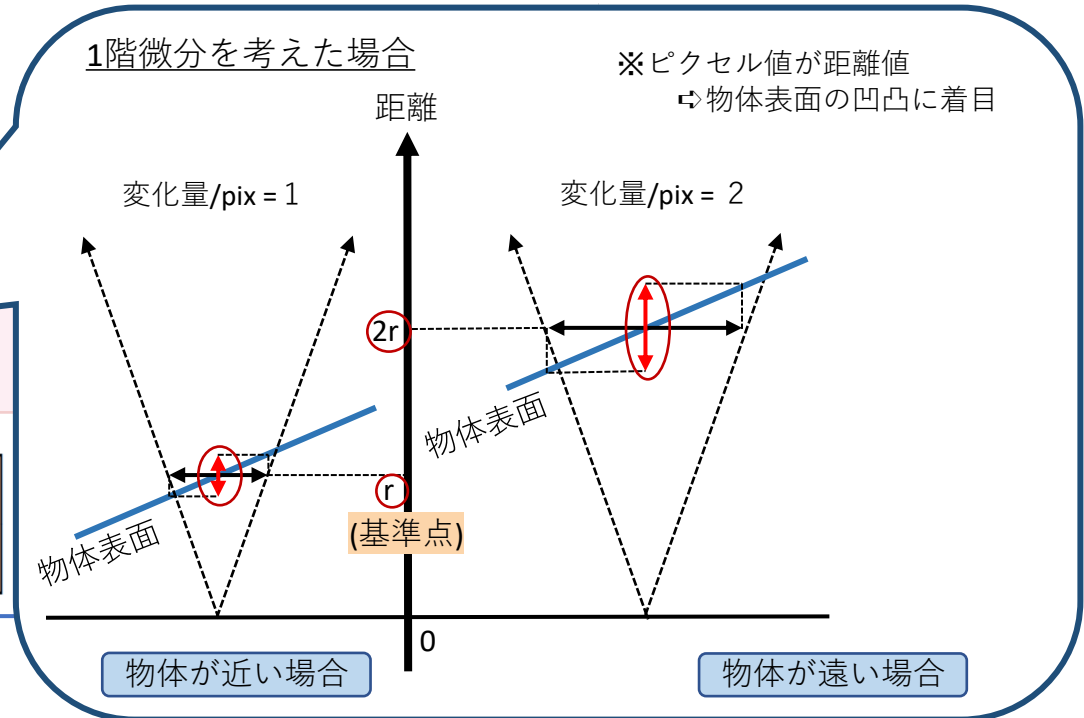
## ➤ 距離同変畳み込み(REconv)

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \dots + \beta_2 \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix}$$

畳み込みフィルタ  $D^d$

$$\begin{bmatrix} w_1' & w_2' & w_3' \\ w_4' & w_5' & w_6' \\ w_7' & w_8' & w_9' \end{bmatrix} = \dots + \beta_2 r_x \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix}$$

畳み込みフィルタ  $D^r$



- スケールに依らず同じ特徴量が得られる

$$D_*^r [L_s[f]](s\xi) = \frac{r_* s_*}{=1} D_*^d [f](\xi) = D_*^r [f](\xi)$$

# 提案手法 / 概要

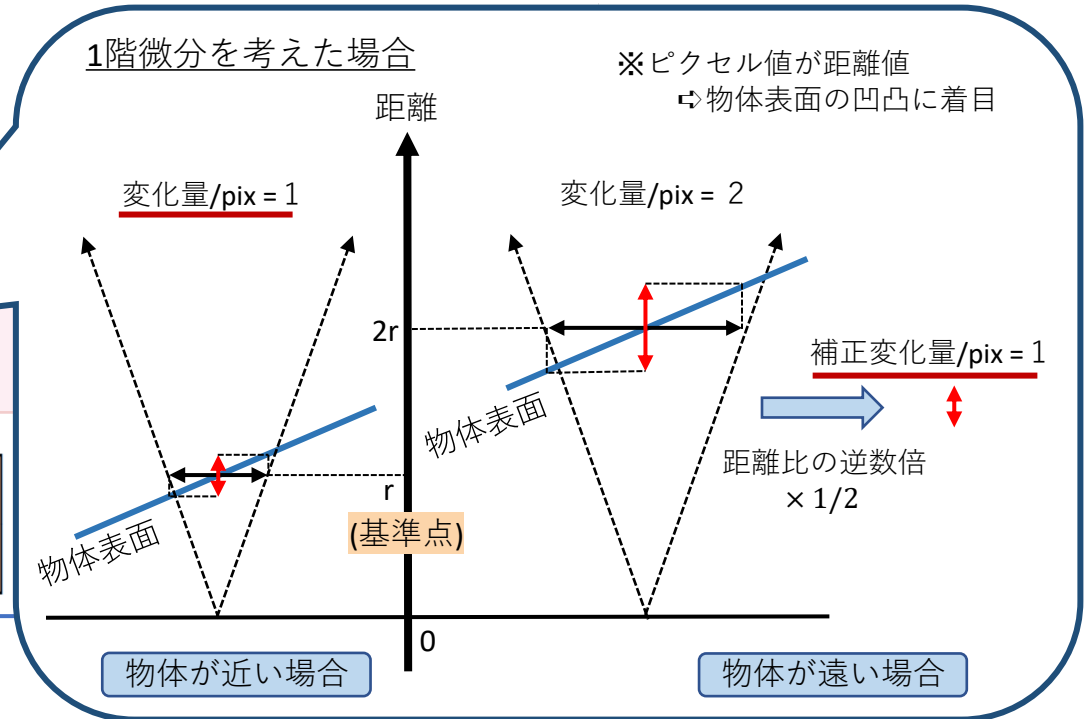
## ➤ 距離同変畳み込み(REconv)

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \dots + \beta_2 \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix}$$

畳み込みフィルタ  $D^d$

$$\begin{bmatrix} w_1' & w_2' & w_3' \\ w_4' & w_5' & w_6' \\ w_7' & w_8' & w_9' \end{bmatrix} = \dots + \beta_2 r_x \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix}$$

畳み込みフィルタ  $D^r$



- スケールに依らず同じ特徴量が得られる

$$D_*^r [L_s[f]](s\xi) = \frac{r_* s_*}{=1} D_*^d [f](\xi) = D_*^r [f](\xi)$$

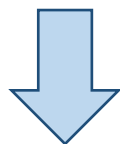


# 提案手法 / 概要

## ➤ 距離同変畳み込み(REconv)

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \beta_1 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \beta_2 \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} + \dots + \beta_9 \frac{1}{(\Delta x)^2(\Delta y)^2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

畳み込みフィルタ  $D^d$        $D_0^d$        $D_x^d$        $D_{xxyy}^d$



距離 $r$ と微分階数に応じた重みをかける  
 $r_* = \frac{k_x^{i_*} k_y^{j_*}}{r^{i_*+j_*}} \quad (r_* s_* = 1 \text{が成立})$

$$\begin{bmatrix} w_1' & w_2' & w_3' \\ w_4' & w_5' & w_6' \\ w_7' & w_8' & w_9' \end{bmatrix} = \beta_1 r_0 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \beta_2 r_x \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} + \dots + \beta_9 r_{xxyy} \frac{1}{(\Delta x)^2(\Delta y)^2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

畳み込みフィルタ  $D_r^d$        $D_0^r$        $D_x^r$        $D_{xxyy}^r$

- スケールに依らず同じ特徴量が得られる

$$D_*^r [L_s[f]](s\xi) = \frac{r_* s_*}{=1} D_*^d [f](\xi) = D_*^r [f](\xi)$$

# 提案手法 / 概要

## ➤ 距離同変畳み込み(REconv)

$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \beta_1 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \beta_2 \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} + \dots + \beta_9 \frac{1}{(\Delta x)^2(\Delta y)^2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

畳み込みフィルタ  $D^d$        $D_0^d$        $D_x^d$        $D_{xxyy}^d$



距離 $r$ と微分階数に応じた重みをかける

$$r_* = \frac{k_x^{i_*} k_y^{j_*}}{r^{i_*+j_*}} \quad (r_* s_* = 1 \text{が成立})$$

$$\begin{bmatrix} w_1' & w_2' & w_3' \\ w_4' & w_5' & w_6' \\ w_7' & w_8' & w_9' \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} \dots \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

畳み込みフィルタ  $D_r^d$        $D_0^r$        $D_x^r$        $D_{xxyy}^r$

スケール同変な畳み込み手法になる

- スケールに依らず同じ特徴量が得られる

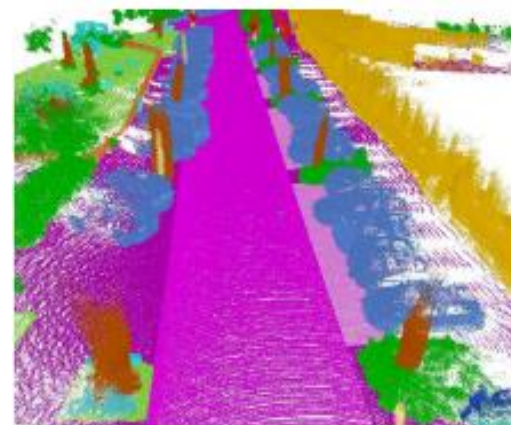
$$D_*^r [L_s[f]](s\xi) = \underbrace{r_* s_*}_{=1} D_*^d [f](\xi) = D_*^r [f](\xi)$$

# 評価実験 / 実験設定

画像引用: [k]

## ➤ データセット

- Semantic KITTI dataset
  - 点ごとに密にラベル付けされた屋外点群データセット



## ➤ 評価指標

- $mIoU$  (mean intersection-over-union)

$$mIoU = \frac{1}{C} \sum_{c=1}^C \frac{\text{予測領域} \cap \text{正解領域}}{\text{予測領域} \cup \text{正解領域}}$$

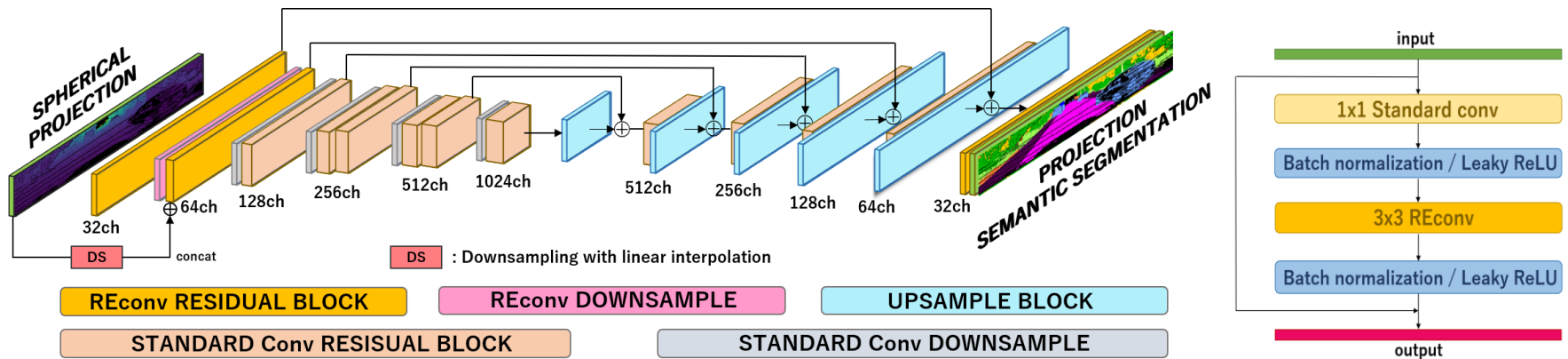
$C$ : クラス数

※この値が大きいほど高精度と言える

# 評価実験 / 実験設定

## ネットワーク構造

- RangeNet[1]をベースラインとする
- 最初の3層をREconvに置き換え



## 入力

- $(range, x, y, z, remission)$  の5チャンネル  
距離値                      反射強度

# 評価実験 / 性能評価

## セグメンテーション性能

method	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean IoU
RangeNet21++ [1]	87.3	<b>19.0</b>	30.1	24.2	23.4	24.3	47.6	0.0	93.0	39.8	78.8	<b>0.3</b>	80.3	40.8	<b>81.3</b>	<b>45.9</b>	69.5	40.2	<b>30.8</b>	45.1
REconvNet21(ours)	<b>89.7</b>	13.2	<b>38.5</b>	<b>29.6</b>	<b>25.2</b>	<b>26.4</b>	<b>48.0</b>	0.0	<b>93.3</b>	<b>42.9</b>	<b>79.9</b>	0.2	<b>81.5</b>	<b>43.0</b>	80.7	45.1	<b>69.7</b>	<b>40.5</b>	30.1	<b>46.2</b>

- ベースラインから  $mIoU$  が 1.1% 向上
- 表面が多い物体クラスで大幅な精度の改善  
Ex) “car” や “building” など
- 表面が少ない物体クラスで性能悪化の傾向  
Ex) “bicycle” や “traffic-sign” など



# 評価実験 / 性能評価

## セグメンテーション性能

method	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean IoU
RangeNet21++ [1]	87.3	<b>19.0</b>	30.1	24.2	23.4	24.3	47.6	0.0	93.0	39.8	78.8	<b>0.3</b>	80.3	40.8	<b>81.3</b>	<b>45.9</b>	69.5	40.2	<b>30.8</b>	45.1
REconvNet21(ours)	<b>89.7</b>	13.2	<b>38.5</b>	<b>29.6</b>	<b>25.2</b>	<b>26.4</b>	<b>48.0</b>	0.0	<b>93.3</b>	<b>42.9</b>	<b>79.9</b>	0.2	<b>81.5</b>	<b>43.0</b>	80.7	45.1	<b>69.7</b>	<b>40.5</b>	30.1	<b>46.2</b>

- ベースラインから  $mIoU$  が 1.1% 向上
- 表面が多い物体クラスで大幅な精度の改善  
Ex) “car” や “building” など
- 表面が少ない物体クラスで性能悪化の傾向  
Ex) “bicycle” や “traffic-sign” など

# 評価実験 / 性能評価

## セグメンテーション性能

method	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean IoU
RangeNet21++ [1]	87.3	19.0	30.1	24.2	23.4	24.3	47.6	0.0	93.0	39.8	78.8	0.3	80.3	40.8	81.3	45.9	69.5	40.2	30.8	45.1
REconvNet21(ours)	89.7	13.2	38.5	29.6	25.2	26.4	48.0	0.0	93.3	42.9	79.9	0.2	81.5	43.0	80.7	45.1	69.7	40.5	30.1	46.2

- ベースラインから  $mIoU$  が 1.1% 向上
- 表面が多い物体クラスで大幅な精度の改善  
Ex) “car” や “building” など
- 表面が少ない物体クラスで性能悪化の傾向  
Ex) “bicycle” や “traffic-sign” など

# 評価実験 / 性能評価

## セグメンテーション性能

method	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean IoU
RangeNet21++ [1]	87.3		30.1	24.2	23.4	24.3	47.6	0.0	93.0	39.8	78.8	0.3	80.3	40.8	81.3	45.9	69.5	40.2		45.1
REconvNet21(ours)	89.7	19.0	38.5	29.6	25.2	26.4	48.0	0.0	93.3	42.9	79.9	0.2	81.5	43.0	80.7	45.1	69.7	40.5	30.8	46.2
		13.2																	30.1	

- ベースラインから  $mIoU$  が 1.1% 向上
- 表面が多い物体クラスで大幅な精度の改善  
Ex) “car” や “building” など
- 表面が少ない物体クラスで性能悪化の傾向  
Ex) “bicycle” や “traffic-sign” など

# 評価実験 / 性能評価

## ▶ 同変性の評価

- 評価方法

- 異スケールの入力に対する特徴マップ類似度

- 評価指標

$$\text{Equivariance Error} = \frac{\text{MSE}}{\text{variance}}$$

※Errorが小さいほど  
同変性があると言える

methods	metrics	1st layer	3rd layer	
RangeNet21 [1]	MSE	0.0329	0.0151	
	variance	64 × 2048	0.0486	0.0317
		32 × 1024	0.0662	0.0506
	Equivariance Error	0.5664	0.3778	
REconvNet21(ours)	MSE	0.0032	0.0039	
	variance	64 × 2048	0.0428	0.0329
		32 × 1024	0.0413	0.0352
	Equivariance Error	<b>0.0823</b>	<b>0.1211</b>	

- 結果

- REconvを用いている部分で同変性誤差が小さい

➡ REconvには確かにスケール同変性がある

# まとめ

---

## ➤ 概要

- 自動運転における深層学習を用いた環境認識
- LiDAR点群を2次元に投射するアプローチ
- 距離によるスケールの違いを補正するような新たな畳み込み手法を提案

## ➤ 結論

- ベースラインから*mIoU*が1.1%向上
- 確かにスケール同変性があることが確認された

➡ REconvの有効性が示された

# 参考文献

---

## ➤ 論文引用

1. Milioto, Andres et al. “ RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. ”Proc. of theIEEE/RSJ IROS (2019)
2. Behley, Jens, et al. ”Semantickitti: A dataset for semantic scene understanding of lidar sequences.” inProc. of the IEEE/CVF ICCV (2019)
3. Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. ”Indoor segmentation and support inference from rgb-d images.” In Proceedings of the ECCV, pages 746–760. Springer, 2012.

# 参考文献

---

## ➤ 画像引用 (1/2)

- a. <https://www.kurume-it.ac.jp/future/selfdriving-attention>
- b. <https://industry.ricoh.com/automotive/stereo-camera>
- c. <https://velodynelidar.com/surround-lidar/>
- d. <https://www.denso.com/jp/ja/news/newsroom/2022/20220114-01/>
- e. <https://automation-insights.blog/2016/03/30/photoelectric-basics-distance-measuring/>
- f. <https://arxiv.org/abs/2303.05367>

# 参考文献

---

## ➤ 画像引用 (2/2)

g. <https://arxiv.org/abs/2207.02605>

h. <https://ieeexplore.ieee.org/document/8967762>

i. <https://arxiv.org/abs/2004.11803>

j. [https://www.biz-newspaper.com/free/light\\_001.html](https://www.biz-newspaper.com/free/light_001.html)

k. <https://arxiv.org/abs/1904.01416>

l. <https://arxiv.org/abs/1808.00671>

m. <https://arxiv.org/abs/2109.05441>

n. <https://arxiv.org/abs/2207.00026>

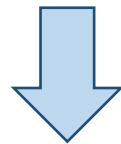
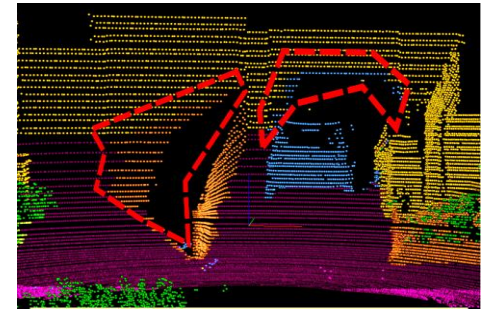


# 付録 / k-NN後処理

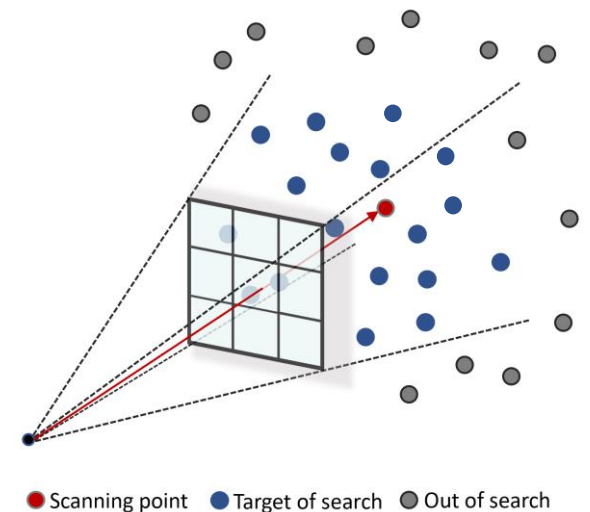
画像引用 : <https://ieeexplore.ieee.org/document/8967762>

## ➤ 点群空間におけるk-NN後処理

- 解像度的に1 pixelに複数の点に対応
- 同じpixelに対応する点は同じラベル
- 背景に"shadow like artifact"が出来る



- K-NN識別によりラベルを精製
  - 以下の近似により計算量削減
    1. 探索範囲を近傍数ピクセルに限定
    2. 点同士の距離を距離値 $r$ の差で近似

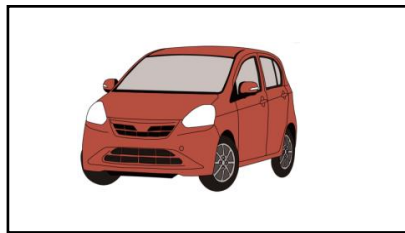


# 付録 / 距離画像の結合 (1/3)

画像引用: [1],[n]

## ➤ 距離値の扱い

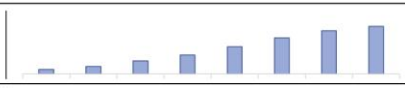
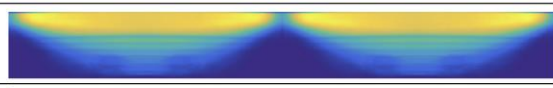

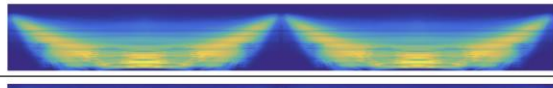
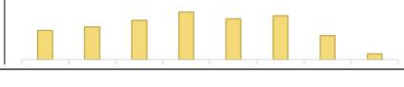
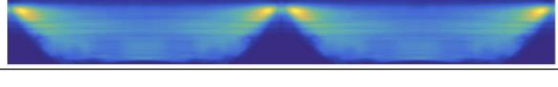
- 微分作用素により値の相対値を見るため  
距離の絶対的な値が消えてしまいやすい



畳み込み  $\ast$   $\frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} D_x^d$  ➔

2ピクセルあたりの変化量を  
計算しているのに等しい

- 距離とクラス分布の相関性が事前分布的に機能  
➔ クラス分類において距離の絶対的な値が重要

Class	Type	Proportion	Distribution	Heatmap
vegetation	static	24.825%		
sidewalk	static	16.353%		
car	dynamic	4.657%		

※この例では正確には  
距離ではなく  
垂直方向の視野角

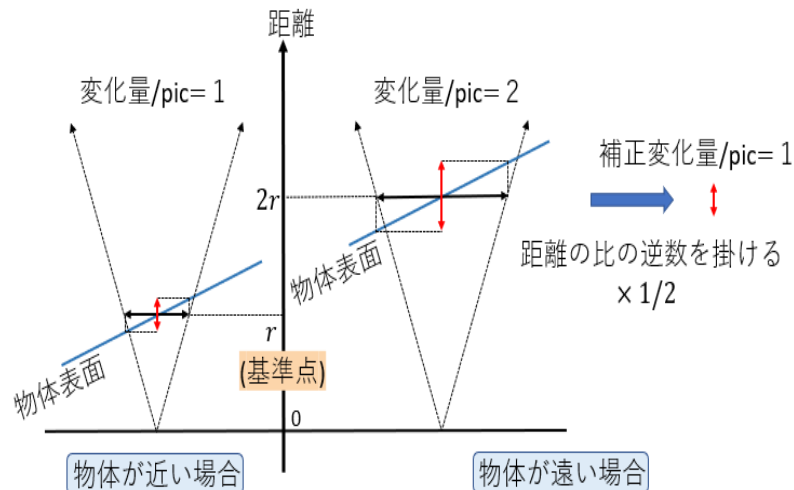
# 付録 / 距離画像の結合 (2/3)

画像引用 : [1],[n]

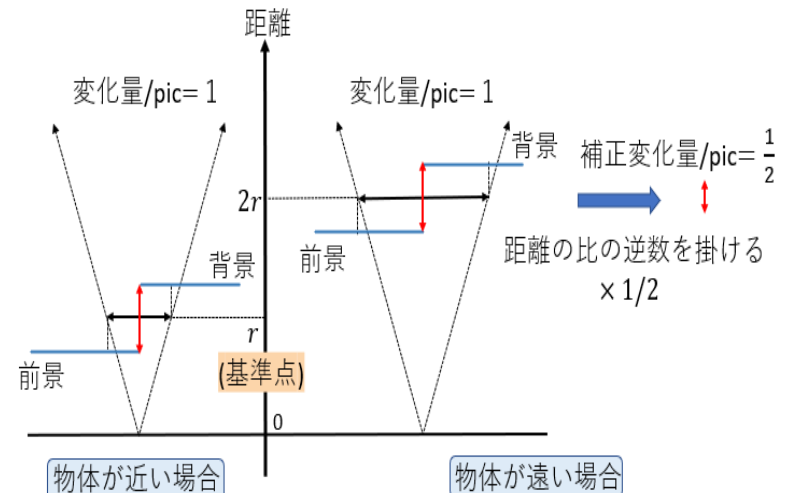
## エッジ抽出能力

- セグメンテーションにおいて物体境界は重要
- エッジの検出能力が精度に大きく影響を与える

同一平面内における1次微分場合



エッジ部分における1次微分の場合



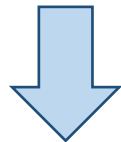
➡ REconvの理論特性的にエッジをぼかしてしまう

# 付録 / 距離画像の結合 (3/3)

画像引用: [1],[n]

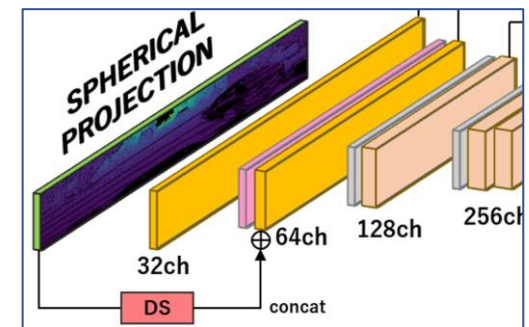
## ➤ 距離画像の結合

- スケール同変性によってパラメータ効率向上 ➤
- 微分処理により絶対的な距離の値が消滅 ➤
- 理論特性的にエッジ抽出が苦手 ➤



REconvによる効果を楽しつつ  
弱点を補うために

- REconvで処理した特徴マップに  
距離画像をチャンネル方向に結合
- 距離情報を付加してから次の層へ



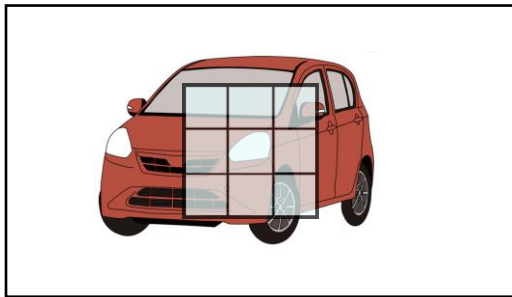
# 付録 / REconvの組み込み位置 (1/2)

## ▶ 最初の3層にしか用いていない理由

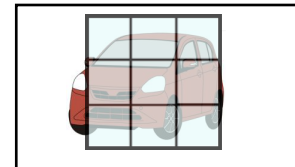
- 離散微分は微小区間での変化量であることが前提

$$D_x[f](\xi) = \lim_{\underline{h_x \rightarrow 0}} \frac{f(\xi + h_x) - f(\xi - h_x)}{\underline{2h_x}} \approx \frac{f(\xi + \Delta x) - f(\xi - \Delta x)}{\underline{2\Delta x}}.$$

- ダウンサンプリングによりスケールダウンすると  
1ピクセル当たりの実際の長さが大きくなる



1ピクセルあたりの長さ (小)



1ピクセルあたりの長さ (大)

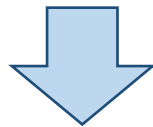
# 付録 / REconvの組み込み位置 (2/2)

## ➤ 最初の3層にしか用いていない理由

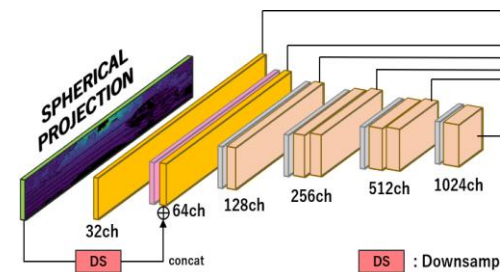
- 1ピクセル当たりの大きさが大きくなると  
離散微分の理論値との解離が大きくなる

$$\lim_{h_x \rightarrow 0} \frac{f(\xi + h_x) - f(\xi - h_x)}{2h_x} \neq \frac{f(\xi + \Delta x) - f(\xi - \Delta x)}{2\Delta x}$$

- エンコーダの後方に行くほど  
ダウンスケールされる



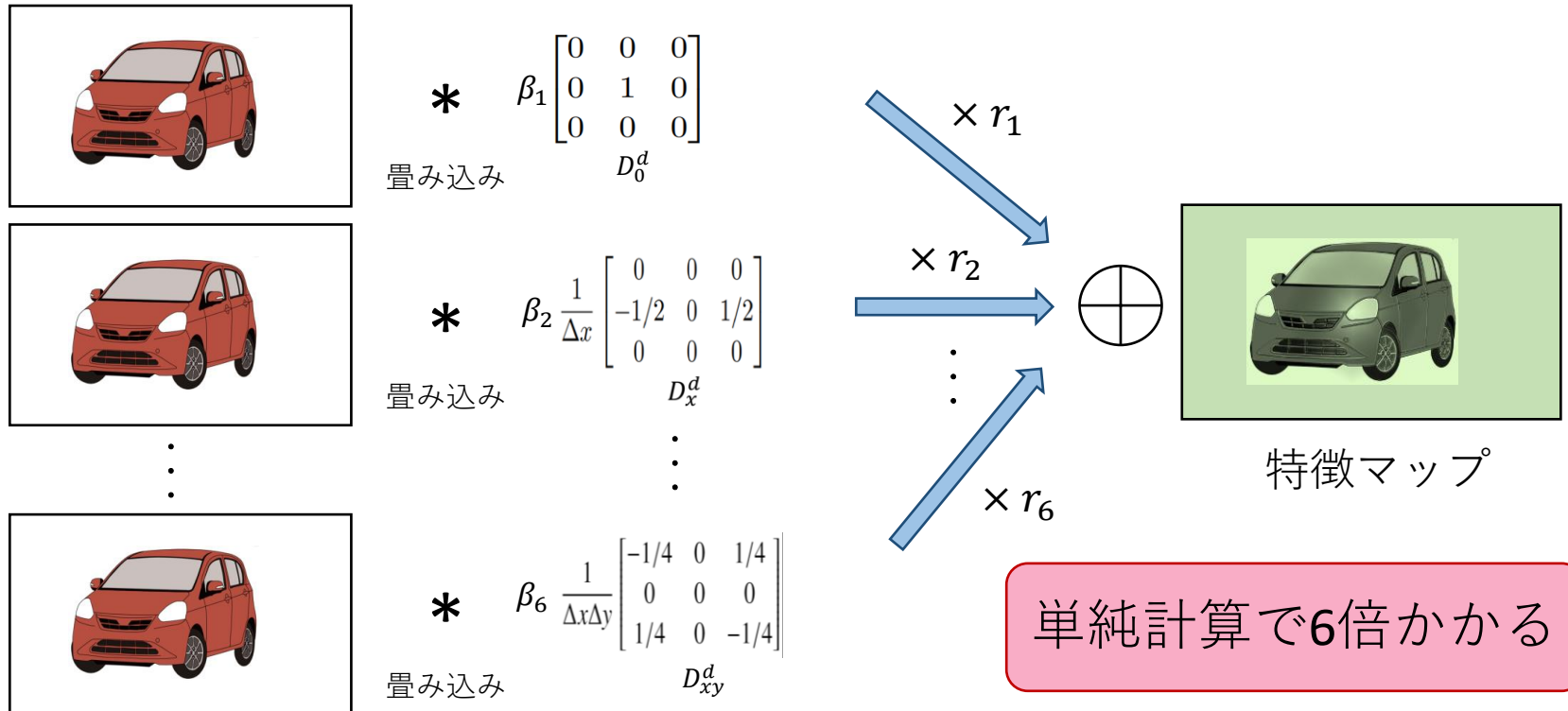
- ネットワーク中の前方数層にのみ使用する



# 付録 / 計算速度 (1/2)

## ➤ 基本コンセプト

- 6個の偏微分フィルタそれぞれによる畳み込み
- 適切な距離重みをそれぞれかけて線形結合



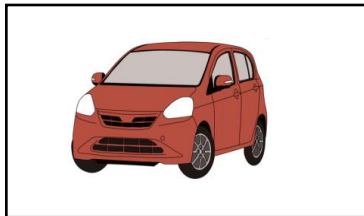
# 付録 / 計算速度 (1/2)

## 基本コンセプト

- 6個の偏微分フィル
- 適切な距離重みをそ

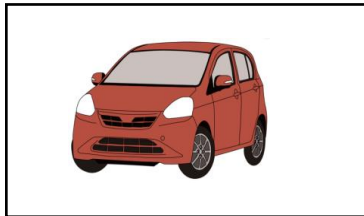
$$\text{距離重み } r_* = \frac{k_x^{i_*} k_y^{j_*}}{r^{i_*+j_*}}$$

- 偏微階数ごと
  - ピクセルごと
- 両方によって値が変わる
- 別々に畳み込んで重み付けする必要がある



$$* \beta_1 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

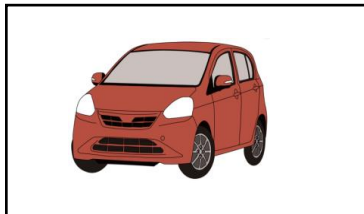
畳み込み  $D_0^d$



$$* \beta_2 \frac{1}{\Delta x} \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 0 & 1/2 \\ 0 & 0 & 0 \end{bmatrix}$$

畳み込み  $D_x^d$

⋮



$$* \beta_6 \frac{1}{\Delta x \Delta y} \begin{bmatrix} -1/4 & 0 & 1/4 \\ 0 & 0 & 0 \\ 1/4 & 0 & -1/4 \end{bmatrix}$$

畳み込み  $D_{xy}^d$

$\times r_1$

$\times r_2$

⋮

$\times r_6$



特徴マップ

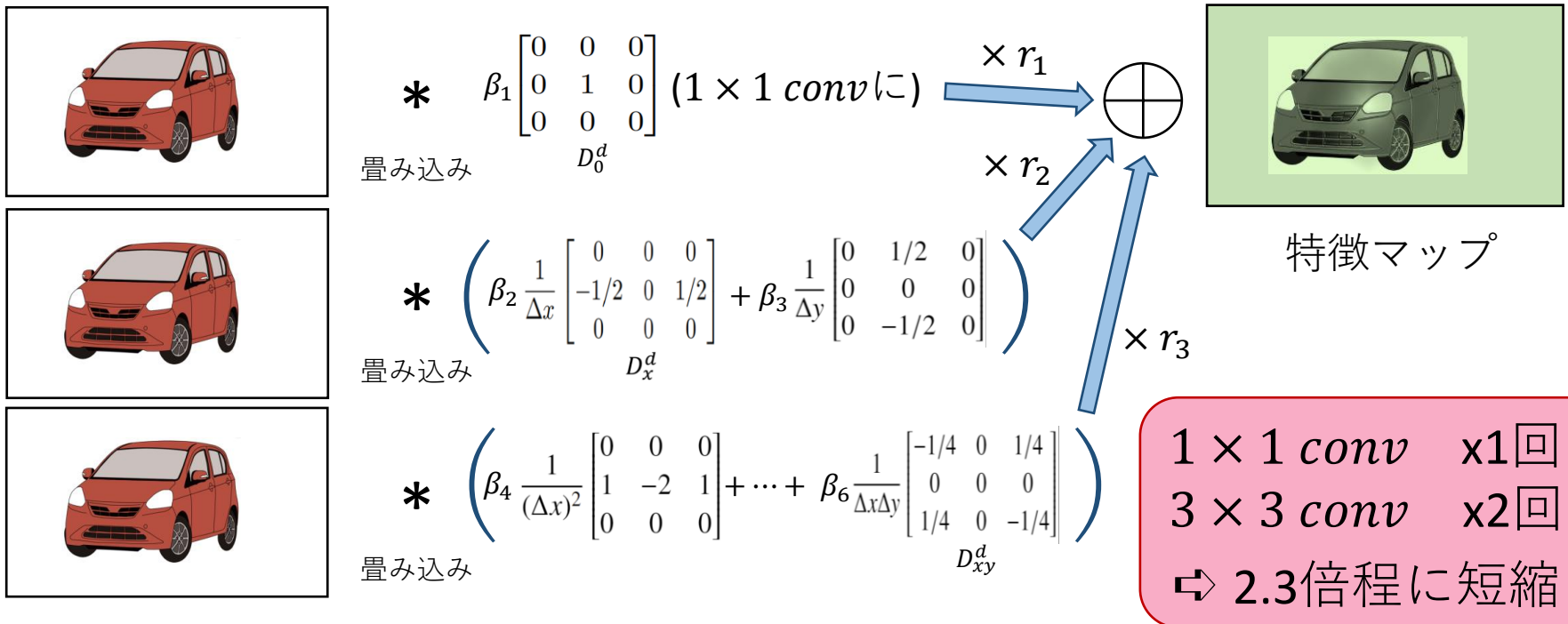
単純計算で6倍かかる



# 付録 / 計算速度 (2/2)

## ▶ 速度アップスキーム

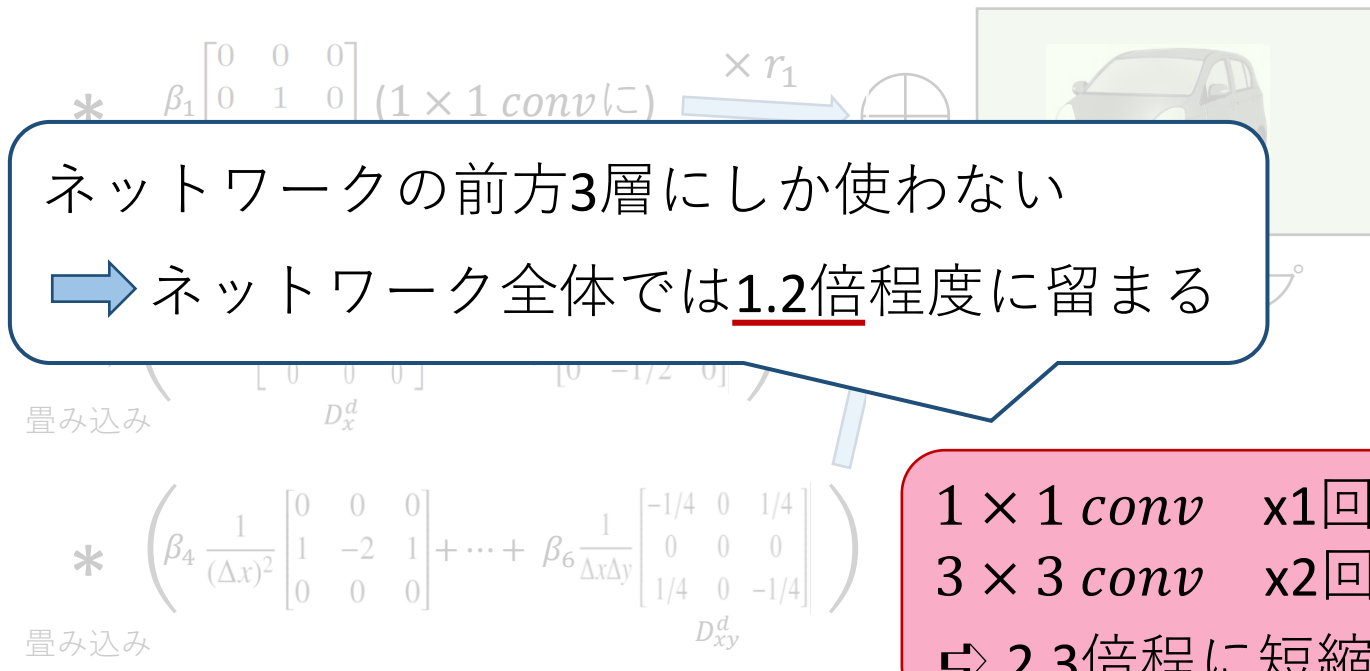
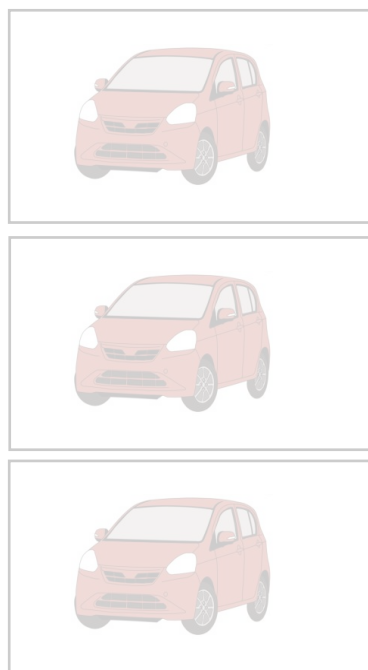
- 距離重みの $x, y$ 方向により異なる部分を学習重みに吸収  
→ 同じ階数の微分作用素はまとめて重み付け可能に



# 付録 / 計算速度 (2/2)

## ▶ 速度アップスキーム

- 距離重みの $x, y$ 方向により異なる部分を学習重みに吸収  
→ 同じ階数の微分作用素はまとめて重み付け可能に



1 × 1 conv x1回  
3 × 3 conv x2回  
⇔ 2.3倍程に短縮

# 追加実験 / 屋内環境への適用

## ➤ 屋内データセットへの適用

- 条件の異なる屋内距離画像データNYUDv2[3]に適用

※実験設定や評価指標は屋外点群データと同じ

method	Bed	Books	Ceiling	Chair	Floor	Furniture	Object	Picture	Sofa	Table	TV	Wall	Window	mean IoU
RangeNet21++ [1]	39.3	<b>23.7</b>	59.8	36.3	74.2	44.0	36.0	<b>51.3</b>	36.6	26.0	38.0	70.3	<b>53.0</b>	45.3
REconvNet21(ours)	<b>55.5</b>	17.5	<b>67.4</b>	<b>47.8</b>	<b>87.2</b>	48.1	<b>38.2</b>	49.6	45.4	<b>33.7</b>	<b>38.3</b>	71.4	50.1	<b>50.0</b>

- ベースラインから*mIoU*が4.7%向上
  - ➡ 屋内環境に対しても有効に機能する
  - ➡ 他の幅広いアプリケーションへの適用可能性