

# Deep Closest Point

Nikolay Tsoy, Vojtěch Nydrle

5. ledna 2020

# Zadání

# Zadání

## Tema OJ1:

- Vyzkoušet DCP (<https://arxiv.org/abs/1905.03304>) na real-world datech jako náhrada standardního SLAM algoritmu.

# Zadání

## Tema OJ1:

- Vyzkoušet DCP (<https://arxiv.org/abs/1905.03304>) na real-world datech jako náhrada standardního SLAM algoritmu.

## DCP

- Deep Closest Point
- Deep Learning náhrada ICP

# ICP

# ICP

**Iterační algoritmus**

# ICP

## Iterační algoritmus

- určuje  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$  mezi množinami bodů  $X$  a  $Y$ 
  - $X = \{\vec{x}_1, \dots, \vec{x}_N\} \subset \mathbb{R}^3$
  - $Y = \{\vec{y}_1, \dots, \vec{y}_N\} \subset \mathbb{R}^3$
  - $Y$  vznikne otočením a posunutím  $X$

# ICP

## Iterační algoritmus

- určuje  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$  mezi množinami bodů  $X$  a  $Y$

- $X = \{\vec{x}_1, \dots, \vec{x}_N\} \subset \mathbb{R}^3$

- $Y = \{\vec{y}_1, \dots, \vec{y}_N\} \subset \mathbb{R}^3$

- $Y$  vznikne otočením a posunutím  $X$

- tak aby  $E(\mathbf{R}_{xy}, \vec{t}_{xy})$  byla minimální

- $$E(\mathbf{R}_{xy}, \vec{t}_{xy}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{R}_{xy} \vec{x}_i + \vec{t}_{xy} - \vec{y}_{m(x_i)}\|^2$$

- $$m(x_i) = \operatorname{argmin}_j \|\mathbf{R}_{xy} \vec{x}_i + \vec{t}_{xy} - \vec{y}_j\|^2$$



# Problémy ICP

# Problémy ICP

- 1 nelze optimalizovat  $\mathbf{R}_{xy}$ ,  $\vec{t}_{xy}$  i  $m$  najednou

# Problémy ICP

- 1 nelze optimalizovat  $\mathbf{R}_{xy}$ ,  $\vec{t}_{xy}$  i  $m$  najednou
- 2 v jednom kroku optimalizuje  $m$  a v dalším  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$

# Problémy ICP

- 1 nelze optimalizovat  $\mathbf{R}_{xy}$ ,  $\vec{t}_{xy}$  i  $m$  najednou
- 2 v jednom kroku optimalizuje  $m$  a v dalším  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$
- 3 velmi často najde jen lokální optimum

# Problémy ICP

- 1 nelze optimalizovat  $\mathbf{R}_{xy}$ ,  $\vec{t}_{xy}$  i  $m$  najednou
- 2 v jednom kroku optimalizuje  $m$  a v dalším  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$
- 3 velmi často najde jen lokální optimum
- 4 neuvažuje zajímavost některých bodů

# Problémy ICP

- 1 nelze optimalizovat  $\mathbf{R}_{xy}$ ,  $\vec{t}_{xy}$  i  $m$  najednou
- 2 v jednom kroku optimalizuje  $m$  a v dalším  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$
- 3 velmi často najde jen lokální optimum
- 4 neuvažuje zajímavost některých bodů
- 5 neporadí si se šumem a řídkostí měření

# DCP

# DCP

- 1 nejprve stanoví  $m(x_i)$



# DCP

- 1 nejprve stanoví  $m(x_i)$
- 2 z odpovídajících si  $\vec{x}_i$  a  $\vec{y}_i \in \mathbb{R}^3$  vypočte  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$

# DCP

- 1 nejprve stanoví  $m(x_i)$
- 2 z odpovídajících si  $\vec{x}_i$  a  $\vec{y}_i \in \mathbb{R}^3$  vypočte  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$
- 3  $\vec{x}_c = \frac{1}{N} \sum_{i=1}^N \vec{x}_i$ ,  $\vec{y}_c = \frac{1}{N} \sum_{i=1}^N \vec{y}_i$

# DCP

- 1 nejprve stanoví  $m(x_i)$
- 2 z odpovídajících si  $\vec{x}_i$  a  $\vec{y}_i \in \mathbb{R}^3$  vypočte  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$
- 3  $\vec{x}_c = \frac{1}{N} \sum_{i=1}^N \vec{x}_i$ ,  $\vec{y}_c = \frac{1}{N} \sum_{i=1}^N \vec{y}_i$
- 4  $\mathbf{H} = \sum_{i=1}^N (\vec{x}_i - \vec{x}_c)(\vec{y}_{m(x_i)} - \vec{y}_c)^T = \mathbf{USV}^T$

# DCP

- 1 nejprve stanoví  $m(x_i)$
- 2 z odpovídajících si  $\vec{x}_i$  a  $\vec{y}_i \in \mathbb{R}^3$  vypočte  $\mathbf{R}_{xy}$  a  $\vec{t}_{xy}$
- 3  $\vec{x}_c = \frac{1}{N} \sum_{i=1}^N \vec{x}_i$ ,  $\vec{y}_c = \frac{1}{N} \sum_{i=1}^N \vec{y}_i$
- 4  $\mathbf{H} = \sum_{i=1}^N (\vec{x}_i - \vec{x}_c)(\vec{y}_{m(x_i)} - \vec{y}_c)^T = \mathbf{USV}^T$
- 5  $\mathbf{R}_{xy} = \mathbf{VU}^T$ ,  $\vec{t}_{xy} = \vec{y}_c - \mathbf{R}_{xy}\vec{x}_c$

# DCP - nalezení $m(x_i)$

# DCP - nalezení $m(x_i)$

- 1 PointNet nebo DGCNN před poslední vrstvou

# DCP - nalezení $m(x_i)$

- 1 PointNet nebo DGCNN před poslední vrstvou
- 2 generuje  $F_X = \{\vec{x}_1^L, \dots, \vec{x}_N^L\} \subset \mathbb{R}^P$  a  $F_Y = \{\vec{y}_1^L, \dots, \vec{y}_N^L\} \subset \mathbb{R}^P$

# DCP - nalezení $m(x_i)$

- 1 PointNet nebo DGCNN před poslední vrstvou
- 2 generuje  $F_X = \{\vec{x}_1^L, \dots, \vec{x}_N^L\} \subset \mathbb{R}^P$  a  $F_Y = \{\vec{y}_1^L, \dots, \vec{y}_N^L\} \subset \mathbb{R}^P$
- 3  $\vec{x}_i^L$  a  $\vec{y}_i^L$  "sémanticky" popisují bod  $\vec{x}_i$  a  $\vec{y}_i$



# DCP - nalezení $m(x_i)$

- 1 PointNet nebo DGCNN před poslední vrstvou
- 2 generuje  $F_X = \{\vec{x}_1^L, \dots, \vec{x}_N^L\} \subset \mathbb{R}^P$  a  $F_Y = \{\vec{y}_1^L, \dots, \vec{y}_N^L\} \subset \mathbb{R}^P$
- 3  $\vec{x}_i^L$  a  $\vec{y}_i^L$  "sémanticky" popisují bod  $\vec{x}_i$  a  $\vec{y}_i$
- 4  $m(x_i) = \text{softmax}(\Phi(Y)^T \Phi(x_i))$ ,  $\Phi$  je "Transformer"

# DCP - nalezení $m(x_i)$

- 1 PointNet nebo DGCNN před poslední vrstvou
- 2 generuje  $F_X = \{\vec{x}_1^L, \dots, \vec{x}_N^L\} \subset \mathbb{R}^P$  a  $F_Y = \{\vec{y}_1^L, \dots, \vec{y}_N^L\} \subset \mathbb{R}^P$
- 3  $\vec{x}_i^L$  a  $\vec{y}_i^L$  "sémanticky" popisují bod  $\vec{x}_i$  a  $\vec{y}_i$
- 4  $m(x_i) = \text{softmax}(\Phi(Y)^T \Phi(x_i))$ ,  $\Phi$  je "Transformer"
- 5 taky poznáváme podobné body ne podle barvy a polohy ale podle významu

# DCP

# DCP

## Učení:

- 1  $L = \|\mathbf{R}_{xy}^T \mathbf{R}_{xy}^g - I\|^2 + \|\vec{t}_{xy} - \vec{t}_{xy}^g\|^2 + \lambda \|\theta\|$
- 2  $\mathbf{R}_{xy}^g$  a  $\vec{t}_{xy}^g$  popisují skutečnou transformaci,  $\theta$  jsou parametry CNN

# DCP

## Učení:

1  $L = \|\mathbf{R}_{xy}^T \mathbf{R}_{xy}^g - I\|^2 + \|\vec{t}_{xy} - \vec{t}_{xy}^g\|^2 + \lambda \|\theta\|$

2  $\mathbf{R}_{xy}^g$  a  $\vec{t}_{xy}^g$  popisují skutečnou transformaci,  $\theta$  jsou parametry CNN

## Závěr autorů:

# DCP

## Učení:

- 1  $L = \|\mathbf{R}_{xy}^T \mathbf{R}_{xy}^g - I\|^2 + \|\vec{t}_{xy} - \vec{t}_{xy}^g\|^2 + \lambda \|\theta\|$
- 2  $\mathbf{R}_{xy}^g$  a  $\vec{t}_{xy}^g$  popisují skutečnou transformaci,  $\theta$  jsou parametry CNN

## Závěr autorů:

- 1 DCP je dost spolehlivý na určení kvalitního výstupu v jednom běhu, ale nechá se ještě doplnit ICP.

# DCP

## Učení:

- 1  $L = \|\mathbf{R}_{xy}^T \mathbf{R}_{xy}^g - I\|^2 + \|\vec{t}_{xy} - \vec{t}_{xy}^g\|^2 + \lambda \|\theta\|$
- 2  $\mathbf{R}_{xy}^g$  a  $\vec{t}_{xy}^g$  popisují skutečnou transformaci,  $\theta$  jsou parametry CNN

## Závěr autorů:

- 1 DCP je dost spolehlivý na určení kvalitního výstupu v jednom běhu, ale nechá se ještě doplnit ICP.
- 2 Může být snadno použit ve složitějších například SALM

# Naše práce



# Naše práce

## Dataset:

- "KITTI Vision" 22 sekvencí k 11 se skutečnou pozici

# Naše práce

## **Dataset:**

- "KITTI Vision" 22 sekvencí k 11 se skutečnou pozici
- na jedné sekvenci jsme učily a druhé testovaly

## **Problémy začlenění KITTY do DCP:**

# Naše práce

## Dataset:

- "KITTI Vision" 22 sekvencí k 11 se skutečnou pozici
- na jedné sekvenci jsme učily a druhé testovaly

## Problémy začlenění KITTY do DCP:

- 1 celý lidarový záběr je natolik velký při forwardpass spotřebuje víc jak 16 GB RAM proto jsme použili jen prvních  $N$  bodů

# Naše práce

## Dataset:

- "KITTI Vision" 22 sekvencí k 11 se skutečnou pozici
- na jedné sekvenci jsme učily a druhé testovaly

## Problémy začlenění KITTY do DCP:

- 1 celý lidarový záběr je natolik velký při forwardpass spotřebuje víc jak 16 GB RAM proto jsme použili jen prvních  $N$  bodů
- 2 špatně jsme pochopili popis datasetu a trénovaly na špatně spočítaných rotačních maticích

# Dosažené výsledky

# Dosažené výsledky

- 1 načíst dataset do formátu, který vyžaduje DCP

# Dosažené výsledky

- 1 načíst dataset do formátu, který vyžaduje DCP
- 2 natrénovat na prvních  $N$  bodech pointkloudů jedné sekvence

# Dosažené výsledky

- 1 načíst dataset do formátu, který vyžaduje DCP
- 2 natrénovat na prvních  $N$  bodech pointkloudů jedné sekvence



# Dosažené výsledky

- 1 načíst dataset do formátu, který vyžaduje DCP
- 2 natrénovat na prvních  $N$  bodech pointkloudů jedné sekvence  
na testovacích datech veliká chyba

# Dosažené výsledky

- 1 načíst dataset do formátu, který vyžaduje DCP
- 2 natrénovat na prvních  $N$  bodech pointkloudů jedné sekvence  
na testovacích datech veliká chyba
- 3 natrénovat na náhodných  $N$  bodech pointkloudů jedné  
sekvence

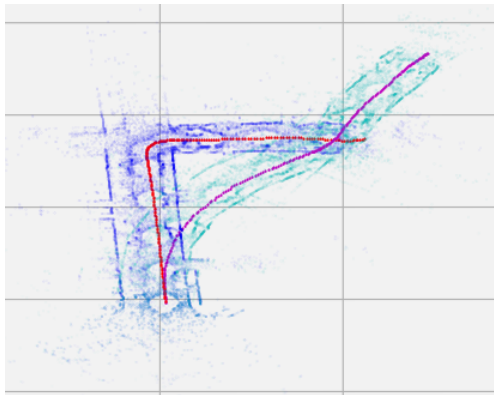
# Dosažené výsledky

- 1 načíst dataset do formátu, který vyžaduje DCP
- 2 natrénovat na prvních  $N$  bodech pointkloudů jedné sekvence  
na testovacích datech veliká chyba
- 3 natrénovat na náhodných  $N$  bodech pointkloudů jedné  
sekvence zlepšení výsledků
- 4 předělat dataset na trénování všech kombinací záběrů se  
společnou částí scény

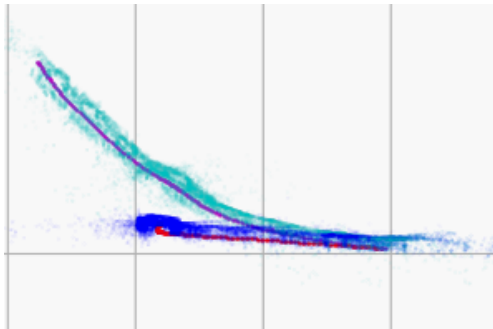
# Dosažené výsledky

- 1 načíst dataset do formátu, který vyžaduje DCP
- 2 natrénovat na prvních  $N$  bodech pointkloudů jedné sekvence  
na testovacích datech veliká chyba
- 3 natrénovat na náhodných  $N$  bodech pointkloudů jedné  
sekvence zlepšení výsledků
- 4 předělat dataset na trénování všech kombinací záběrů se  
společnou částí scény jedna epocha se trénuje přes dvě hodiny  
→ nestihly jsme natrénovat

# Dosažené výsledky



# Dosažené výsledky



# Proč to nefunguje?

# Proč to nefunguje?

- 1 testováno na otočený, posunutých a zašuměných objektech



# Proč to nefunguje?

- 1 testováno na otočený, posunutých a zašuměných objektech
- 2 ne na deformovaných objektech

# Proč to nefunguje?

- 1 testováno na otočený, posunutých a zašuměných objektech
- 2 ne na deformovaných objektech

# Proč to nefunguje?

- 1 testováno na otočený, posunutých a zašuměných objektech
- 2 ne na deformovaných objektech

# Proč to nefunguje?

- 1 testováno na otočený, posunutých a zašuměných objektech
- 2 ne na deformovaných objektech
- 3 přebývající body jsou špatně přiřazeny

# Proč to nefunguje?

- 1 testováno na otočený, posunutých a zašuměných objektech
- 2 ne na deformovaných objektech
- 3 přebývající body jsou špatně přiřazeny

**Lze to zpravit?**

# Proč to nefunguje?

- 1 testováno na otočený, posunutých a zašuměných objektech
- 2 ne na deformovaných objektech
- 3 přebývající body jsou špatně přiřazeny

## Lze to zpravit?

- 1 záběru odstranit část scény (určenou směrem a rychlostí pohybu)

# Proč to nefunguje?

- 1 testováno na otočený, posunutých a zašuměných objektech
- 2 ne na deformovaných objektech
- 3 přebývající body jsou špatně přiřazeny

## Lze to zpravit?

- 1 záběru odstranit část scény (určenou směrem a rychlostí pohybu)
- 2 hypotézu lze ověřit otestováním na datech z interiéru

# Možné zlepšení



# Možné zlepšení

- Nahradit  $m = \text{softmax}(\Phi_Y(y)^T \Phi_X(x_i))$  složitější funkcí, která by se nesnažila přiřadit body co nejsou v druhé množině.

# Možné zlepšení

- Nahradit  $m = \text{softmax}(\Phi_Y(y)^T \Phi_X(x_i))$  složitější funkcí, která by se nesnažila přiřadit body co nejsou v druhé množině.
- Trénovat neuronovou síť na více jízdách než jen na jedné.

# Možné zlepšení

- Nahradit  $m = \text{softmax}(\Phi_Y(y)^T \Phi_X(x_i))$  složitější funkcí, která by se nesnažila přiřadit body co nejsou v druhé množině.
- Trénovat neuronovou síť na více jízdách než jen na jedné.
- Při výpočtu aktuální pozice nevycházet jen z předešlého záběru scény ale všech v dosahu lidarů.

# Možné zlepšení

- Nahradit  $m = \text{softmax}(\Phi_Y(y)^T \Phi_X(x_i))$  složitější funkcí, která by se nesnažila přiřadit body co nejsou v druhé množině.
- Trénovat neuronovou síť na více jízdách než jen na jedné.
- Při výpočtu aktuální pozice nevycházet jen z předešlého záběru scény ale všech v dosahu lidarů.
- Po vypočtení pozice pomocí DCP použít na tento výsledek ještě ICP pro zpřesnění výsledku.

# Děkujeme za pozornost