

DEEP CLOSEST POINT

{ NIKOLAY TSOY, VOJTĚCH NYDRLE }

ZADÁNÍ

Tema OJ1:

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

ITERATIVE CLOSEST POINT

ICP je iterační algoritmus, který určuje rotaci \mathbf{R}_{xy} a translaci \vec{t}_{xy} mezi množinami bodů X a Y splňujícími:

- $X = \{\vec{x}_1, \dots, \vec{x}_N\} \subset \mathbb{R}^3$
- $Y = \{\vec{y}_1, \dots, \vec{y}_N\} \subset \mathbb{R}^3$
- Y vznikne otočením a posunutím X

tak že minimalizuje chybu $E(\mathbf{R}_{xy}, \vec{t}_{xy})$.

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

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

Problémy IPC

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

DATASET

Jako real-world data byl vybrán "KITTI Vision" dataset, obsahující 22 nahrávek jízdy autem s lidarem GPS a spoustou dalších senzorů. K prvním 11 jízdám jsou poskytnuta jak lidarová data, ta skutečná pozice a natočení auta.

DCP bylo trénováno na jedné jízdě a testováno na jiné.

DCP - DEEP CLOSEST POINT

Na rozdíl od ICP nejprve stanoví přiřazení odpovídajících si bodů $y = m(\vec{x})$, $\vec{x} \in Y$, $\vec{x} \in Y$.

Pro stanovení odpovídajících si bodů používá PointNet nebo DGCNN skácený o poslední vrstvu, tím ke každému bodu \vec{x}_i dostane mnohorozměrný vektor \vec{x}_i^L , popisující význam tohoto bodu. Tedy z množin X a Y vypočítá množiny $F_X = \{\vec{x}_1^L, \dots, \vec{x}_N^L\} \subset \mathbb{R}^P$ a $F_Y = \{\vec{y}_1^L, \dots, \vec{y}_N^L\} \subset \mathbb{R}^P$, použitím naučených parametrů θ použité neuronové sítě.

Pak vypočte přiřazení odpovídajících si bodů:

$$m(x_i) = \operatorname{softmax}(\Phi_Y(Y)^T \Phi_X(x_i)),$$

kde Φ funkce reagující na společné znaky množin.

$$\Phi_Y(Y) = F_Y + \phi(F_Y, F_X)$$

$$\Phi_X(x_i) = x_i^L + \phi(x_i^L, F_Y)$$

ϕ je asymetrická funkce zvaná "Transformer", používaná při zarovnání sekvencí, jejíž parametry se také učí při učení celého DCP modelu.

Poté z odpovídajících si dvojic \vec{x}_i a $\hat{\vec{y}}_i = \mathbf{Y}^T m(\vec{x}_i)$, kde \mathbf{Y} je matice z prvků Y , vypočte rotaci \mathbf{R}_{xy} a translaci \vec{t}_{xy} mezi množinami X a Y . Těžiště množin je:

$$\vec{x}_c = \frac{1}{N} \sum_{i=1}^N \vec{x}_i, \vec{y}_c = \frac{1}{N} \sum_{i=1}^N \hat{\vec{y}}_i$$

a jejich cross-covarianční matice:

$$\mathbf{H} = \sum_{i=1}^N (\vec{x}_i - \vec{x}_c)(\hat{\vec{y}}_i - \vec{y}_c)^T = \mathbf{U} \mathbf{S} \mathbf{V}^T$$

ze které se nechá vypočíst posun a rotace:

$$\mathbf{R}_{xy} = \mathbf{V} \mathbf{U}^T, \vec{t}_{xy} = \vec{y}_c - \mathbf{R}_{xy} \vec{x}_c.$$

Při učení se používá ztrátová funkce

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

kde \mathbf{R}_{xy}^g a \vec{t}_{xy}^g popisují skutečnou transformaci.

DALŠÍ MOŽNÉ ZLEPŠENÍ

- Nahradit $m = \operatorname{softmax}(\Phi_Y(y)^T \Phi_X(x_i))$ 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 lidaru.
- Po vypočtení pozice pomocí DCP použít na tento výsledek ještě ICP pro zpřesnění výsledku.

DOSAŽENÉ VÝSLEDKY

Podařilo se natrénovat DCP na dvou po sobě jdoucích lidarových záběrech, tak, že z každého záběru bylo použito jen N náhodných bodů. Výsledek testu sítě je na obr. 1. Dále byla síť trénována na všech dvojicích obsahujících společnou část, tím se zvětšil dataset a trénování jsem nestihly dokončit. Domníváme se že DCP nefungovalo, protože bylo vyvinuto a testováno pro určování vzájemné polohy dvou stejných objektu otočených, posunutých a zašuměných, ale ne deformovaných. To by mohlo jít napravit odstraněním bodů které nejsou v obou záběrech. Tato hypotéza by šla potvrdit

VÝSLEDKY AUTORŮ

Autoři původního článku prováděli experimenty na datasetu ModelNet40, který obsahuje 12 311 CAD modelů rozdělených do 40 kategorií. Použili 9 843 modelů na trénování a 2 468 na testování, modely velikostně přizpůsobily aby byly vepsány jednotkové koule, náhodně otáčely v rozsahu 45° a posouvaly až o 0.5. Porovnání DCP a ostatních algoritmu je v tabulce.

Model	RMSE(R)	RMSE(t)
ICP	29.9148	0.2909
Go-ICP	11.8523	0.0257
FGR	9.3628	0.0139
PointNetLK	15.0954	0.0221
DCP	1.143385	0.001786

Table 1: Porovnání s jinými metodami

Dále ukázaly že s DGCNN dosahuje DCP lepších výsledků než s PointNet, a je možné zpřesnit výsledky následným použitím IPC.

testem na datech z interiéru.

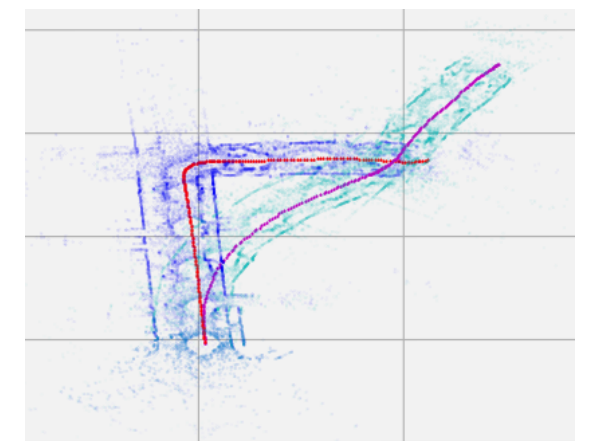


Figure 1: Test na testovací sekvenci