

Prostorová indexace s využitím gridu

Návod na cvičení, Geoinformatika.

Na vstupu je mračno bodů $P = \{p_i\}_{i=1}^n$, kde $p_i = [x_i, y_i, z_i]$. Nad mračnem zkonstruujeme pomocnou indexační strukturu reprezentovanou gridem. Grid je tvořen jednotlivými buňkami (voxely), celkový počet buněk je funkcí velikosti datasetu n a prostorové dimenze

$$n_b = n^{1/3}.$$

Počet buněk v řádku/sloupci gridu je roven

$$n_r = n_c = n_b^{1/3}.$$

Velikost buňky. Velikost buňky určíme ze vztahu

$$b_x = \frac{\bar{x} - \underline{x}}{n_r}, \quad b_y = \frac{\bar{y} - \underline{y}}{n_r}, \quad b_z = \frac{\bar{z} - \underline{z}}{n_r},$$

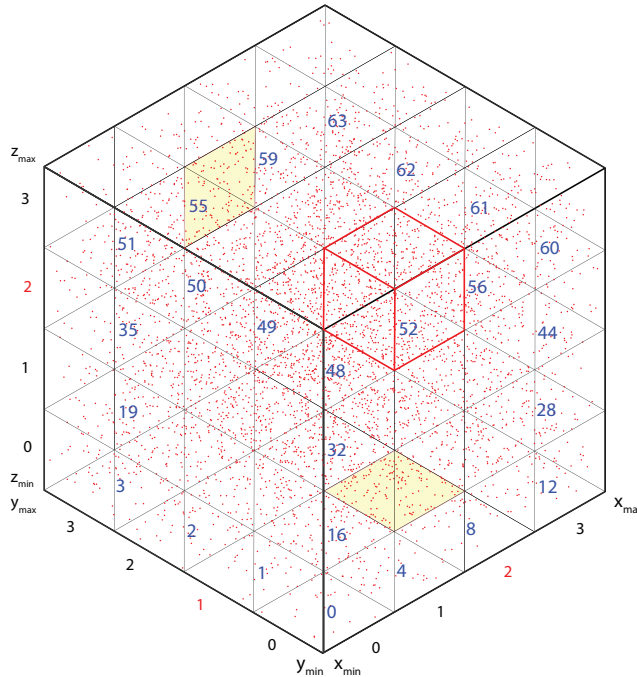
kde

$$\underline{x} = \min_{1 \leq i \leq n} \{x_i\}, \quad \bar{x} = \max_{1 \leq i \leq n} \{x_i\}, \quad \underline{y} = \min_{1 \leq i \leq n} \{y_i\}, \quad \bar{y} = \max_{1 \leq i \leq n} \{y_i\}, \quad \underline{z} = \min_{1 \leq i \leq n} \{z_i\}, \quad \bar{z} = \max_{1 \leq i \leq n} \{z_i\}.$$

Redukce souřadnic. Redukované souřadnice bodu $p_i = [x'_i, y'_i, z'_i]$ mají tvar

$$x'_i = \frac{x_i - \underline{x}}{\bar{x} - \underline{x}}, \quad y'_i = \frac{y_i - \underline{y}}{\bar{y} - \underline{y}}, \quad z'_i = \frac{z_i - \underline{z}}{\bar{z} - \underline{z}}.$$

Následně budou použity k výpočtu hashovací funkce.

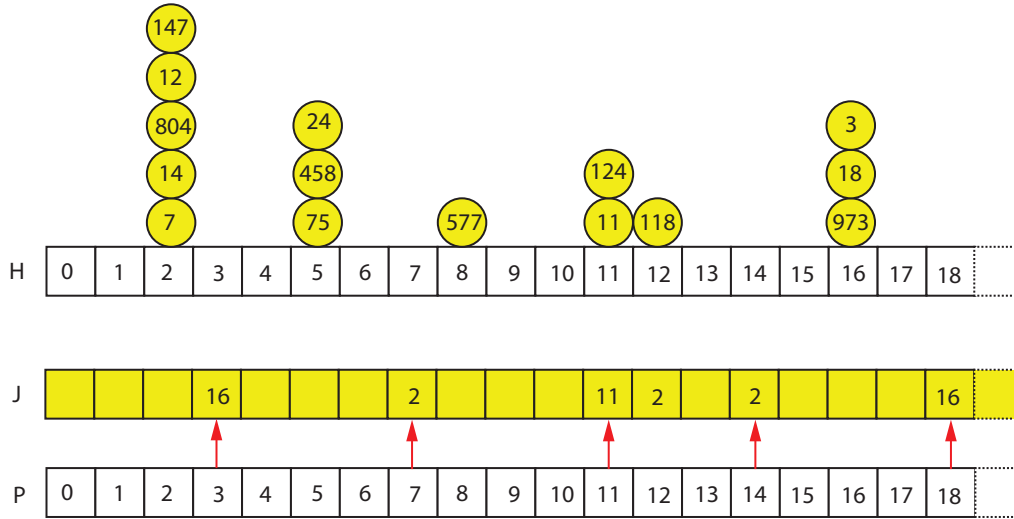


Obrázek 1: Ukázka 3D indexačního gridu, z3D a 1D indexy.

Výpočet indexů bodu. Poloha každé z buněk v gridu je určena trojicí indexů $\langle j_x, j_y, j_z \rangle$. Bod p_i leží v buňce gridu s indexy

$$j_x = \lfloor cn_r x'_i \rfloor, \quad j_y = \lfloor cn_r y'_i \rfloor, \quad j_z = \lfloor cn_r z'_i \rfloor,$$

kde $c = 0.99$ je “zaokrouhlovací” konstanta.



Obrázek 2: Ukázka prostorové indexace bodů.

Hashovací funkce. Pro konverzi $\langle j_x, j_y, j_z \rangle$ na jednodimenzionální index j použijeme jednoduchou hashovací funkci

$$a = h(k),$$

kde

$$a \equiv j, \quad k \equiv p, \quad h(k) = j_x + j_y n_r + j_z n_r^2.$$

Číslování buněk probíhá po řádcích a jednotlivých vrstvách, každá z buněk má unikátní hash reprezentovaný indexem j . Protože prostor klíčů je “větší” než prostor adres, dochází ke kolizím, kdy uvnitř jedné buňky bude více bodů.

Datové struktury. Pro vlastní indexaci budou vytvořeny dvě pomocné datové struktury. První představuje *hashovací* tabulku, která pro každou buňku ukládá indexy j_i všech bodů, které jsou v ní obsaženy. Pro reprezentaci v programovacím jazyce Python bude použit Dictionary

$$H = \{j_1 : [i_{11}, i_{12}, \dots, i_{1k}], j_2 : [i_{21}, i_{22}, \dots, i_{2k}], \dots, j_m : [i_{m1}, i_{m2}, \dots, i_{mk}]\}.$$

Druhou strukturu bude tvořit 1D pole, které každému bodu p_i přiřadí jednodimenzionální index j_i

$$J = [j_1, j_2, \dots, j_n].$$

Dojde tak k obousměrnému prolínání a budeme vědět, které body p_i se nachází v konkrétní buňce, a ve které buňce se nachází konkrétní bod p_i .

Aplikace hashovací funkce. Pro libovolný “query point: $q = [x, y, z]$ postupujeme následujícím způsobem. S využitím hashovací funkce spočteme 1D adresu buňky j , která tento bod obsahuje. Dotazem do hashovací struktury

$$Q = H[j]$$

získáme list bodů, který je v buňce obsažen. Tyto body můžeme použít např. k rychlému nalezení nejbližšího souseda. Protože se v každém koši nachází průměrně 1/10 původní množiny, dojde tak zhruba k desetinásobnému urychlení této operace.