

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Detekcia jednoduchých 3D objektov v RGB-D zázname

Bakalárska práca

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Detekcia jednoduchých 3D objektov v RGB-D zázname

Bakalárska práca

Študijný program: Aplikovaná informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: Ing. Viktor Kocur

Bratislava, 2021

Andrej Paluch



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Andrej Paluch
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Detekcia jednoduchých 3D objektov v RGB-D zázname
Detection of simple 3D objects in RGB-D footage

Anotácia: Toto zadanie je súčasťou projektu interakcie ľudského učiteľa s robotom. Robot pri tejto interakcii manipuluje jednoduchými objektmi na základe pokynov od ľudského učiteľa. Pre tento účel je tak vhodné aby robot dokázal správne detegovať pozíciu daných objektov pomocou svojej RGB-D kamery.

Cieľ: Cieľom tejto práce je navrhnúť, implementovať a otestovať algoritmus na detekciu jednoduchých 3D objektov v RGB-D zázname. Algoritmus bude navrhnutý a testovaný v kontexte interakcie ľudského učiteľa s robotom.

Vedúci: Ing. Viktor Kocur
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 30.09.2020

Dátum schválenia: 06.10.2020

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Abstrakt

Cieľom tejto bakalárskej práce je navrhnúť, implementovať a otestovať algoritmus na detekciu jednoduchých 3D objektov v RGB-D zázname. Práca je rozdelená na úvodnú časť popisujúcu prehľad problematiky a možné prístupy rozpoznávania objektov z 3D dát, druhá časť práce je venovaná návrhu riešenia. V tretej časti práce sa nachádza samotný návrh a implementácia riešenia použitím metódy RANSAC.

Kľúčové slová: detekcia jednoduchých 3D objektov, RANSAC

Abstract

The aim of this bachelor thesis is to design, implement and test an algorithm for detecting simple 3D objects in RGB-D recording. The work is divided into an introductory part describing an overview of the issue and possible approaches of recognizing objects from 3D data, the second part of the work is devoted to the design of the solution. In the third part of the work is the design and implementation of the solution using the RANSAC method.

Keywords: simple 3D geometric primitives detection, RANSAC

Obsah

1	Prehľad problematiky	8
1.1	Detegované objekty	8
1.2	Híbková kamera	8
1.3	Vstupne dáta	9
1.4	Výpočet point cloudu	9
1.5	Farebné priestory.....	10
1.6	Možné prístupy	12
1.7	Random sample consensus.....	13
1.7.1	Vzorkovanie vstupných údajov	13
1.7.2	Ohodnotenie potenciálneho modelu.....	13
1.7.3	vylepšenia	14
1.8	Ostatné prístupy	15
1.8.1	Houghova transformácia.....	15
1.8.2	Zhlukovanie v parametrickom priestore.....	15
1.8.3	Zhlukovanie	16
	Literatúra	17

Úvod

Kapitola 1

1 Prehľad problematiky

V tejto kapitole si uvedieme základné teoretické východiská, ktoré sú nevyhnutné pre pochopenie a neskoršie riešenie danej problematiky. Na úvod si vysvetlíme čo sú to jednoduché 3D objekty. Ďalej si popíšeme princípy na základe ktorých dokážu hĺbkové kamery zachytiť 3D dáta, základné reprezentácie 3D dát a možné konverzie medzi nimi. Tiež si uvedieme niečo o farebných modeloch a priestoroch. Hlavnou časťou tejto kapitoly bude vysvetlenie algoritmu RANSAC a náhľad aj do ostatných metód detekcie jednoduchých 3D objektov.

1.1 Detegované objekty

V mojej práci sa zaoberám detekciou jednoduchých 3D objektov. To sú také, pre ktoré platí nasledujúca charakteristika:

- majú fixne daný a konečný počet globálnych vlastných parametrov, t.j. takých, ktoré definujú iba veľkosť, orientáciu a polohu daného objektu.
- sú konvexné (okrem kruhového prstenca).
- sú symetrické.
- majú jednoduchý tvar, a ich spájaním navzájom môžu vznikajú iné komplexné objekty.

Jednoduché objekty spĺňajúce túto definíciu delíme v oblasti detekcie objektov z 3D dát na štyri kategórie:

- rovinné útvary
- kvádre a kocky
- gule, valce a kužele
- ostatné

Väčšina prístupov na detekciu 3D objektov sa snaží nájsť dané objekty skôr ako sieť povrchov, ktoré daný objekt ohraničujú, ako objem, ktorý predstavujú [1].

1.2 Hĺbková kamera

Na snímanie objektov v 3D je použitá kamera Intel RealSense Depth Camera D435i, ktorá na výpočet hĺbky využíva stereoskopické videnie [2]. Princíp stereoskopie spočíva v tom, že z dvoch bodov videnia (kamier), ktorých vzdialenosť poznáme získame dva mierne odlišné obrázky. Ak vieme identifikovať jeden konkrétny bod v našej scéne na oboch obrázkoch, a teda poznať posun, potom vieme vypočítať hĺbku daného bodu.

Implementácia stereoskopického videnia pozostáva z ľavej a pravej kamery, a voliteľného infračerveného projektora, vďaka ktorému je presnosť hĺbky v snímke lepšia, a to hlavne v scénach s nevýraznou textúrou. Ľavá a pravá kamera zachytávajú scénu a posielajú údaje do procesora na spracovanie hĺbky (depth vision processor), ktorý vypočíta hĺbkové hodnoty pre každý pixel v obrázku, pomocou porovnania bodov na obrázku z ľavej kamery a obrázku z pravej kamery, a ich vzájomným posunom. Následne sa vytvorí hĺbková mapa, ktorá sa pridá k RGB obrázku ako ďalší kanál.

Kamera dokáže nasnímať 30 RGB snímok za sekundu v maximálnom rozlíšení 1920x1080 a až 90 hĺbkových máp za sekundu s rozlíšením 1280x720. Pri najvyššom rozlíšení je minimálna hĺbka, ktorú dokáže s veľkou presnosťou určiť 28cm.

1.3 Vstupne dáta

Ako výstup z kamery dostaneme RGB-D záznam. RGB-D obraz je kombináciou normálneho RGB obrázku a jemu prislúchajúcej hĺbkovej mapy. Hĺbková mapa je obraz alebo obrazový kanál, ktorý obsahuje informácie týkajúce sa vzdialenosti povrchov objektov scény od kamery. Spojením s RGB obrázkom tak okrem farebnej informácie o scéne dostávame aj informáciu o tvare. Takýto RGB-D obraz vieme premietnuť do 3D svetových súradníc a získať tak farebný point cloud.

Point cloud, po slovensky mračno bodov, je množina bodov, ktorá reprezentuje scénu. Ideou point cloudu je navzorkovanie skutočného sveta, ktorý je spojitý, do jednotlivých bodov, čím dostaneme diskkrétne dáta. Každý bod je reprezentovaný ako trojica XYZ súradníc s možnými dodatočnými informáciami ako napr. farba alebo normála. Point cloud môže byť štruktúrovaný, to znamená, že jeho body sú organizované v jednej, alebo viacerých 2D mriežkach, čo umožňuje rýchle prehľadávanie.

RGB-D obrazy a farebné point cloudy sú dve najčastejšie používané reprezentácie 3D dát z informáciou o farbe [3]. RGB-D obrazy sú používané hlavne v oblasti počítačového videnia, keďže majú rovnakú topológiu ako obrázky, zatiaľ čo v oblasti počítačovej grafiky sú využívané hlavne point cloudy [3]. Čo je dôležité je, že obe reprezentácie poskytujú rovnakú informáciu o scéne. So správnymi parametrami kamery je teda možná konverzia medzi oboma reprezentáciami. Point cloudy získane konverziou z RGB-D obraz sú navyše organizované, keďže je korešpondencia jedna k jednej medzi bodom v point cloude a pixelom v RGB-D obraze.

1.4 Výpočet point cloudu

Ako sme si uviedli vyššie, medzi RGB-D obrazom a point cloudom je možná konverzia, a práve konverzia z RGB-D obrazu do point cloudu je potrebná, keďže väčšina algoritmov na detekciu objektov pracuje práve s údajmi vo forme point cloudu.

Model dierkovej komory je matematický model, ktorý popisuje vzťah medzi súradnicami bodu v trojrozmernom priestore, a jeho premietnutím na rovinu obrazu. Je založený na princípe fungovania ideálnej dierkovej komory, čo je jednoduchá kamera bez objektívu.

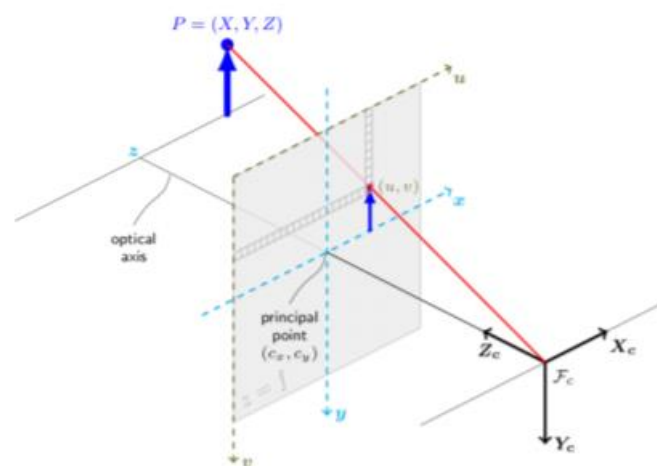
Hlavnou ideou tohto modelu je získanie 2D súradníc premietacej roviny z 3D súradníc sveta, keďže ale ku každému bodu z premietacej roviny (obrázku) poznáme aj jeho hĺbku, dokážeme tak naopak zo súradníc premietacej roviny získať súradnice sveta, A to na základe nasledujúcich pomerne jednoduchých rovníc.

XYZ sú súradnice sveta, xy sú súradnice premietacej roviny, z informácia o hĺbke a f je ohnisková vzdialenosť.

$$Z = z, \quad X = \frac{Z}{f}x, \quad Y = \frac{Z}{f}y$$

Štandardne je však nutné tieto rovnice ešte upraviť, keďže bežne je začiatok súradnicového systému obrazu v jeho ľavom hornom rohu a nie v bode prieniku optickej osi.

$$Z = z, \quad X = \frac{Z}{f}(u - c_x), \quad Y = \frac{Z}{f}(v - c_y)$$



Obrázok 1: pinhole camera model

1.5 Farebné priestory

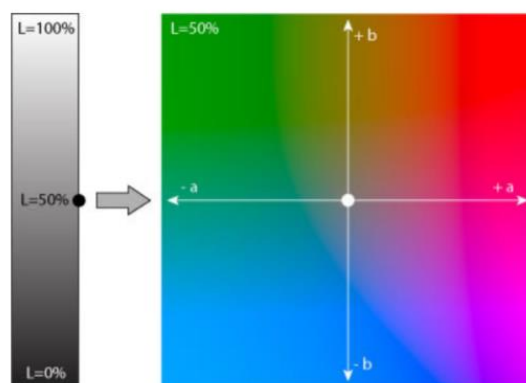
Farebný priestor je špecifická organizácia farieb, väčšinou založený na farebnom modeli, čo je matematické vyjadrenie farby na základe podielu základných zložiek, ktorými je definovaný. Farebné priestory vieme rozdeliť podľa toho, či sú strojovo závislé alebo strojovo nezávislé, a podľa toho z akého farebného modelu vychádzajú.

Medzi základné farebné modely patrí RGB. Jeho základnými zložkami sú červená (Red), zelená (Green) a modrá (Blue). Pre tieto farby je príznačné, že ľudské oko má najlepšiu citlivosť pre ich vlnové dĺžky [4]. Model vieme reprezentovať ako jednotkovú kocku, kde každá z troch osí predstavuje jednu zložku. RGB je z pomedzi ostatných modelov asi najviac technicky orientovaný, no pri práci s obrazmi, predovšetkým segmentácií, nie je veľmi efektívny. V hardvérovo orientovaných modeloch môže byť ťažké vyjadriť niektoré farby a preto vznikli používateľsky orientované modely, ktoré sú navrhnuté pre vnímanie farieb bližšie vnímaniu človeka.

Medzi takéto modely patrí napríklad Lab (často tiež L^*a^*b), ktorý bol navrhnutý tak, aby bol úplne nezávislý na zariadení. To ho umožňuje používať ako referenčný farebný model, čo znamená že pri prevode medzi dvoma modelmi ho môžeme použiť ako pomocný model. Je definovaný troma zložkami:

- svetlosť (L) – popisuje svetlosť bodu v rozsahu 0 až 100.
- zložka a – popisuje farbu na zeleno(-a) – červenej(+a) osi
- zložka b – popisuje farbu na modro(-b) – žltej(+b) osi

Najväčšími výhodami modelu Lab okrem samotnej strojovej nezávislosti je aj to, že má najširší rozsah zaznamenateľných farieb z pomedzi všetkých farebných modelov. Ďalšou veľkou výhodou je úplne oddelenie jasovej zložky od farebných zložiek, čo viac korešponduje s tým ako vníma farby ľudské oko, ktoré je na zmeny jasú oveľa citlivejšie, ako na zmeny farieb.

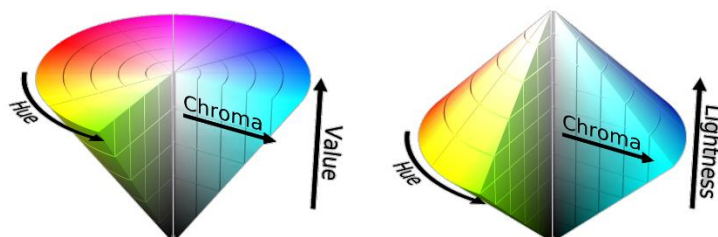


Obrázok 2: vizualizácia modelu Lab [4]

Ďalším používateľsky orientovaným modelom je HSV, ktorý asi najlepšie odpovedá ľudskému vnímaniu farieb [4]. Tiež je definovaný troma základnými zložkami:

- odtieň farby (Hue) – udáva sa v stupňoch od 0 po 360 a popisuje vlastnú čistú farbu.
- sýtosť (Saturation) – udáva sa v percentách a popisuje, ako veľmi je farba čistá, teda bez prímеси bielej.
- jas (Value) – popisuje jas farby v percentách

Podobným modelom ako je HSV je HSL, ale odstraňuje niektoré nedostatky HSV. Je definovaný odtieňom farby, sýtosťou a svetlosťou (Lightness). Jednou z jeho výhod je, že kontrastné farby čierna a biela tu majú svoju vlastnú zložku.



Obrázok 3: vľavo vizualizácia modelu HSV, vpravo vizualizácia modelu HSL

1.6 Možné prístupy

Keď už máme dáta v potrebnom tvare prichádza hlavná časť, ktorou je samotná metóda na detekciu objektov. Existuje viacero možných prístupov, z ktorých každý má svoje výhody aj nevýhody. Dané prístupy rozdeľujeme do troch skupín:

- využívajúce náhodu (resp. stochastické) – RANSAC, local statistic
- využívajúce parametrický priestor – Hough transform, Clustering parameter space
- zhlukovacie metódy – region growing, automatic clustering, segmentation then fitting

V praxi platí že mnoho konkrétnych aplikácií na detekciu jednoduchých 3D objektov nepoužíva čisto iba jednu z týchto metód, ale nejako vhodne ich kombinuje [1]. Dané metódy môžeme popísať na základe určitých charakteristík, na základe ktorých sa môžeme inšpirovať pri výbere algoritmu pre našu vlastnú aplikáciu. Niektoré z nich sú napr.:

- detected objects – prvá základná charakteristika podľa ktorej môžeme jednotlivé metódy rozlíšiť, a to podľa objektov, na ktorých detekciu sa zameriavajú.
- application context – kontext, v akom sú poskytnuté vstupne dáta. Teda či ide o interiéry, exteriéry alebo samostatné objekty.
- timing – podľa veľkosti vstupných dát môžeme rozlišovať metódy na tie, ktoré sú real-time, a tie ktoré nie.
- scalability – to ako sa ktorá metóda zachová pri zmene objemu dát
- user assistance – či metóda vyžaduje, alebo ponúka možnosť používateľovi zapojiť sa do procesu.
- needs extra information – či sa vyžaduje nejaká informácia navyše, aby sme mohli úspešne detegovať objekt.
- robustness to noise – či dokáže metóda správne detegovať objekt aj zo zašumených dát.
- Robustness to incomplete data – či dokáže metóda správne detegovať objekt aj z neúplných dát

1.7 Random sample consensus

Random sample consensus, skrátene RANSAC, je veľmi populárny iteratívny algoritmus určený na odhadovanie parametrov matematického modelu, prvý krát ho predstavili Fisher a Bolles v roku 1981 [5]. Je známy predovšetkým tým, že je obzvlášť odolný voči odľahlým hodnotám (tzv. outliers), to sú hodnoty ktoré buď nepatria modelu ktorý chceme detegovať, alebo sú to hodnoty, ktoré vďaka chybe pri meraní, alebo nesprávnou interpretáciou vstupných dát nadobudli nesprávnu hodnotu, a preto tiež nezapadajú do modelu.

Základnou myšlienkou RANSACu je vyskúšať veľa možných náhodných modelov, ktoré by mohli zodpovedať pravým nezašumeným dátam. Je to nedeterministický algoritmus, v zmysle, že poskytuje zmysluplný výsledok iba s určitou pravdepodobnosťou, tá však môže byť vysoká, ak zvolíme dostatočný počet iterácií. Algoritmus pozostáva z dvoch hlavných krokov:

1.7.1 Vzorkovanie vstupných údajov

Štruktúra RANSACu je pomerne jednoduchá. V prvom kroku je zo vstupnej množiny dát vybraná náhodná vzorka. Každý bod má rovnakú šancu, že bude vybraný do vzorky. Na základe tejto vzorky sa vypočíta pasujúci model a jemu zodpovedajúce parametre. Veľkosť vzorky závisí od objektu, ktorý detegujeme. V zásade ale platí, že veľkosť vzorky je najmenšia možná, ale taká aby bolo z nej možné určiť parametre modelu. Napríklad, ak by sme chceli detegovať kružnicu, potrebujeme do vzorky vybrať tri body, keďže kružnica je jednoznačne určená tromi bodmi.

1.7.2 Ohodnotenie potenciálneho modelu

V druhom kroku sa ohodnotí kvalita potenciálneho modelu na celej vstupnej množine. Bežná funkcia na ohodnotenie jednoducho spočíta, ktoré body z celej množiny sú z potencionálnym modelom konzistentné. Tieto dva kroky sa stále opakujú, pričom si pamätáme najlepšie doposiaľ ohodnotený model. Opakujú sa dovtedy, kým buď nevyčerpáme počet iterácií, alebo natrafíme na taký model, ktorý dostatočne dobre vyhovuje vstupným dátam. Čo znamená dostatočne dobre je na konkrétnej aplikácii algoritmu. Existuje viacero variantov RANSACu, ktoré zachovávajú prvú časť kde sa náhodne vzorkuje, ale používajú zložitejšie metódy na ohodnotenie potenciálneho modelu.

1.7.3 Premenné

RANSAC používa na riadenie procesu detekcie tri premenné. Prvá určuje, s akou toleranciou môžeme o nejakom bode prehlásiť, že patrí potenciálnemu modelu. Druhá premenná je maximálny počet iterácií, ktoré musíme vykonať. Ten závisí od toho, koľko odľahlých hodnôt nepatriacich modelu sa nachádza vo vstupnej množine a od veľkosti vzoriek. Viac odľahlých hodnôt vo vstupných dátach a väčšia veľkosť vzoriek znamená viac iterácií ktoré musí

algoritmus vykonať. Tretia premenná sa tiež týka tolerancie, a určuje ako dobre musí byť ohodnotený potencionálny model, aby sme ho mohli prehlásiť za detegovaný objekt.

```
while iterations < k do
  maybeInliers := n randomly selected values from data
  maybeModel := model parameters fitted to maybeInliers
  alsoInliers := empty set
  for every point in data not in maybeInliers do
    if point fits maybeModel with an error smaller than t
      add point to alsoInliers
    end for
  if the number of elements in alsoInliers is > d then
    // This implies that we may have found a good model
    // now test how good it is.
    betterModel := model parameters fitted to all points in maybeInliers and alsoInliers
    thisErr := a measure of how well betterModel fits these points
    if thisErr < bestErr then
      bestFit := betterModel
      bestErr := thisErr
    end if
  end if
  increment iterations
end while

return bestFit
```

Obrázok 4: algoritmus RANSAC - pseudokód [6]

1.7.4 vylepšenia

Výpočtová zložitosť RANSACu sa dá vylepšiť niekoľkými spôsobmi. Rýchlosť algoritmu závisí od dvoch faktorov. Prvým je počet iterácií, ktoré je potrebné vykonať, aby sme mali zaručené, že dostaneme dobrý odhad. Druhým je čas, ktorý nám zaberie ohodnotenie každého potencionálneho modelu, ten je úmerný veľkosti vstupnej množiny.

Platí, že veľké množstvo potencionálnych modelov je vytvorených z kontaminovaných vzoriek (t.j. vzoriek obsahujúcich body nepatriace modelu). Takéto modely sú konzistentné iba s malým množstvom údajov. Ohodnotenie modelov možno zrýchliť tým že . Každý potencionálny model najprv otestujeme iba náhodných bodov zo vstupnej množiny. Ak daný model nedostane dostatočnú podporu od tejto malej množiny náhodných bodov, potom môžeme z vysokou pravdepodobnosťou usúdiť, že nie je dobrým odhadom a ďalej nepokračovať s jeho ohodnocovaním. Modely, ktoré prejdú cez náhodné ohodnotenie sa potom vyhodnotia aj na celej množine. Takto vylepšený algoritmus sa volá Randomized RANSAC [7].

Výkonnosť algoritmu sa zhoršuje z narastajúcou veľkosťou vzorky, keďže je menšia šanca že nebude obsahovať žiadne hodnoty nepatriace modelu (outliers). Bežné pozorovanie je také, že hodnoty nepatriace modelu majú difúzne rozdelenie, na rozdiel od hodnôt patriacich modelu, ktoré majú tendenciu byť umiestnené blízko seba [8]. Preto, pôvodné vzorkovanie bodov môže byť nahradené vzorkovaním kde berieme do úvahy priestorové vzťahy bodov [9]. Počiatočný bod vzorky je vybraný náhodne. Ostatné body sú náhodne zvolené body, ktoré ležia v hypersfére so stredom v prvom bode. Výber vzorky, ktorá obsahuje susedné body môže významne zvýšiť pravdepodobnosť, že všetky body danej vzorky patria modelu,

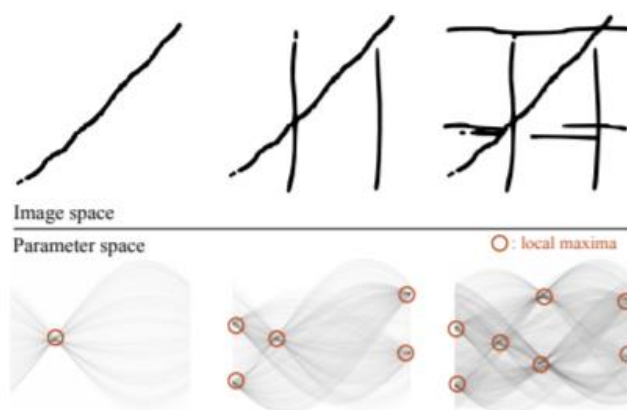
a tým pádom znížime počet potrebných iterácií algoritmu. Takto vylepšený algoritmus poznáme ako NAPSAC [9].

1.8 Ostatné prístupy

1.8.1 Houghova transformácia

Princíp fungovania Houghovej transformácie je založený na vytvorení parametrického priestoru. Parametrický priestor je toľko rozmerný, koľko parametrov definuje detegovaný objekt. Každý bod v ňom tak predstavuje jeden objekt. Po nakvantovaní parametrického priestoru dostaneme tzv. akumulčný priestor. Všetky body zo vstupných dát zahlasujú za všetky tie modely, ktorých sú súčasťou. Ako rozpoznaný objekt bude vybraný z akumulčného priestoru ten, ktorý ma najviac hlasov. Pôvodne sa tento algoritmus používal na detekciu hrán v obraze, no neskôr bol zovšeobecnený na detekciu 2D aj 3D objektov.

Veľkou výhodou tohto algoritmu je odolnosť voči neúplným dátam, zato jeho hlavnou nevýhodou sú nedostatočne definované hranice rozmerov parametrického priestoru, čo môže byť problém z hľadiska pamäti a času.



Obrázok 5: príklad Houghovej transformácie na detekciu hrán [1]

1.8.2 Zhlukovanie v parametrickom priestore

Metódy založené na tomto princípe priamo využívajú a analyzujú parametrický priestor aby detegovali jednoduché objekty. Predovšetkým sú využívané na detekciu rovín, keďže parametrický priestor pre roviny vieme rozdeliť do dvoch disjunktných priestorov. Najprv rozdelíme parametrický priestor podľa normál. Tým ešte nedostaneme jednotlivé objekty ako skôr skupiny bodov s rovnakou orientáciou. Keď však použijeme aj zhlukovanie na základe vzdialenosti, zhluky bodov ktoré dostaneme budú predstavovať detegované objekty.

1.8.3 Zhlukovanie

Existuje niekoľko metód ktoré využívajú segmentáciu a zhlukovanie na detekciu jednoduchých objektov. V zásade rozdeľujeme tri hlavné typy zhlukovania [1]:

- primitive growing
- automatic clustering
- segmentatin then fitting

Algoritmus jednoduchého narastania regiónov sa používa na detekciu poprepájaných objektov buď v RGB-D obrazoch, alebo v štruktúrovaných point cloudoch, kde máme informáciu o susednosti. Na začiatku jedno alebo viac tzv. semienok, to sú body ktoré patria objektom. Potom iteratívne prechádzame ich susedné body, a tie môžeme k ním priradiť ak sú ich vlastnosti dostatočne podobné. Tieto vlastnosti ako farba, hĺbka alebo orientácia normály sú vyhodnocované ako podobné s určitou toleranciou, ktorá zvyčajne závisí od konkrétnej aplikácie algoritmu.

Algoritmy automatického zhlukovania postupujú iteratívne za cieľom rozdelenia bodov do rovnomerne distribuovaných oblastí. Dva najbežnejšie algoritmy automatického zhlukovania sú: K-Means a Mean-Shift. Algoritmus K-Means berie na vstupe fixné číslo k počtu zhlukov, ktoré sa majú detegovať. Najprv sa výbere k náhodných bodov, ktoré budeme považovať za ťažiská zhlukov. Zvyšné body potom priradíme k tým zhlukom, ktorých ťažiská sú najbližšie. Potom vypočítame nové ťažiská jednotlivých zhlukov a celý proces opakujeme až kým neskonvergujeme k výsledku. Na rozdiel od toho, algoritmus Mean-Shift nepotrebuje na vstupe počet zhlukov a dokáže sám zistiť koľko ich je.

Pri metóde segmentation then fitting najprv segmentujeme vstupne dáta, aby sme sa čo najviac zbavili bodov nepatriacich modelu. Segmentovať môžeme napr. metódami ako region growing. V tomto prípade však dané segmentačné algoritmy nevyhľadávajú objekty, namiesto toho používajú heuristiky ako farba, vzdialenosť alebo nejaké iné špecifické pre konkrétnu aplikáciu.

Literatúra

- [1] Adrien Kaiser, Jose Alonso Ybanez Zepeda, Tamy Boubekeur, A Survey of Simple Geometric Primitives Detection Methods for Captured 3D Data, LTCI, Telecom ParisTech, Paris-Saclay University.
- [2] „Intel® RealSense™ Product Family D400 Series datasheet,“ 2020. [Online]. Available: URL: <<https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf>>.
- [3] Paul L. Rosin, Yu-Kun Lai, Ling Shao, Yonghuai Liu, RGB-D Image Analysis and Processing, Springer Nature, 2019.
- [4] J. Chudý, Farebné modely a práca s nimi, Univerzita Tomáše Bati ve Zlíně, 2007.
- [5] M.A. Fischler and R.C. Bolles, Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography, Communications of the ACM, 1981.
- [6] „Random sample consensus,“ [Online]. Available: https://en.wikipedia.org/wiki/Random_sample_consensus.
- [7] O. Chum and J. Matas, Randomized ransac with $t(d,d)$ test, British Machine Vision Conference, Cardiff, UK, 2002.
- [8] H. Cantzler, Random Sample Consensus (RANSAC), Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh.
- [9] D.R. Myatt, P.H.S. Torr, S.J. Nasuto, J.M. Bishop, and R. Craddock., Napsac: High noise, high dimensional robust estimation - it's in the bag..
- [10] Dibya Jyoti Bora, Anil Kumar Gupta, Fayaz Ahmad Khan, Comparing the Performance of $L^*A^*B^*$ and HSV Color Spaces with Respect to Color Image Segmentation, Department of Computer Science & Applications, Barkatullah University, Bhopal, India, 2015.