Rekonštrukcia pletív z mračien bodov

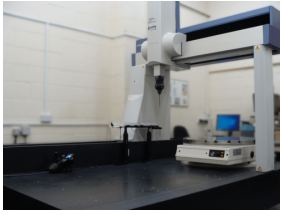
Modelovanie kriviek a plôch (3)

Marcel Makovník

Department of Algebra and Geometry
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava, Slovakia

Motivácia

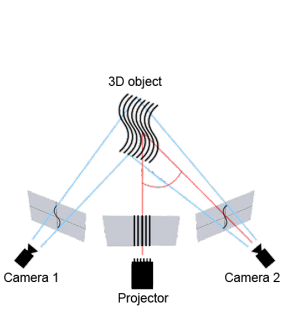
- Spracovanie reálnych 3D objektov pre reprezentáciu na počítači (mračno bodov, mnohouholníkové pletivo) je možné realizovať pomocou rôznych technológií.



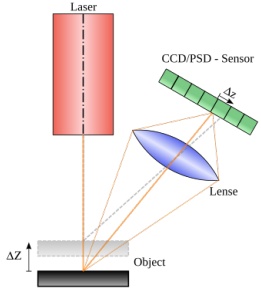
(a) Kontaktný skener



(b) LiDAR



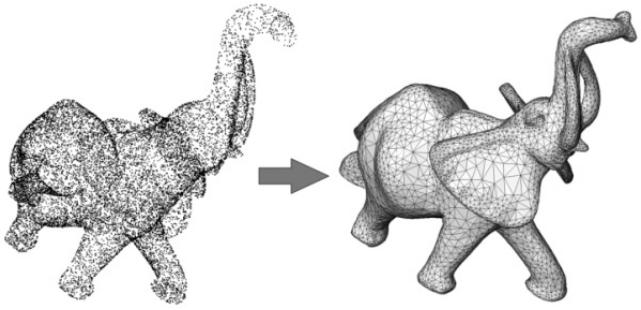
(c) Štruktúrované svetlo



(d) Triangulačné skenovanie

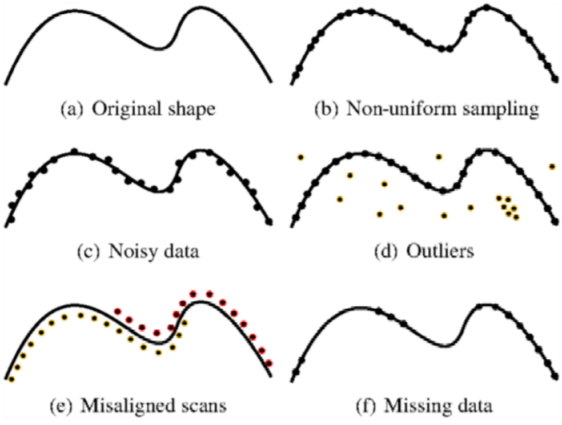
Motivácia

- Väčšina prístupov 3D skenovania bez ďalšieho spracovania ponúka len mračno bodov (neštruktúrovaná množina bodov), nanajvýš príslušné normálové vektory (ak je známy smer skenovania).
- Cieľom je teda z mračna bodov zostrojiť plochu, ktorá je najštandardnejšie charakterizovaná ako mnohouholníkové pletivo.



Motivácia

- Na výstupe môže vzniknúť mračno bodov, ktoré môže obsahovať viacero problémov:



Motivácia

- Existuje množstvo prístupov na rekonštrukciu plôch, v závislosti na predpokladoch pre vstup a požiadaviek pre výstup.
- Ak na vstupe uvažujeme mračno bodov neobsahuje normálové vektory a použitá metóda ich vyžaduje, potrebujeme ich nejakým spôsobom odhadnúť.
- Na výstupe nemusí dostať len mnohouholníkové pletivo (vtedy hovoríme o explicitnej rekonštrukcii), môžeme získať aj implicitný opis zoskenovanej plochy.
- Budeme sa baviť (zatiaľ) len explicitných metódach, napr.:
  - Ball pivoting algorithm (BPA) – algoritmus valiacej sa gule
  - Poisson disk reconstruction (a jeho modifikácie)
  - Rekonštrukcia pomocu Voronoiových oblastí

Ball pivoting algorithm (BPA) - myšlienka

- VSTUP: mračno bodov, kde každý bod má asociovaný normálový vektor
- VÝSTUP: trojuholníkové pletivo, ktoré mračno bodov interpoluje.

- Predstavme si sféru s polomerom ρ, ktorá sa „valí” po mračne bodov
- Dodržiavame pravidlo: **Tri body sformujú trojuholník, ak sa ich sféra dotýka tak, že neobsahuje žiadne ďalšie body (ani v jej vnútri)**
- Algoritmus pokračuje v konštrukcii pletiva tak, že pivotuje sféru po už existujúcich hranách.

Ball pivoting algorithm (BPA) - 2D

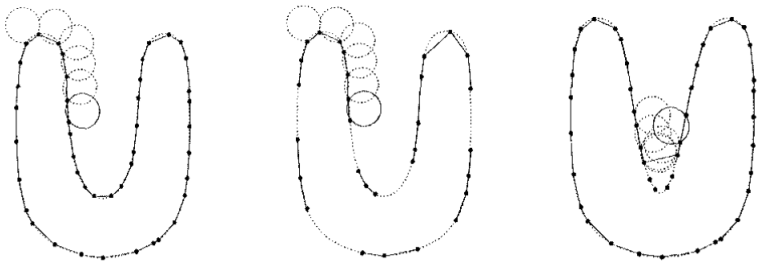
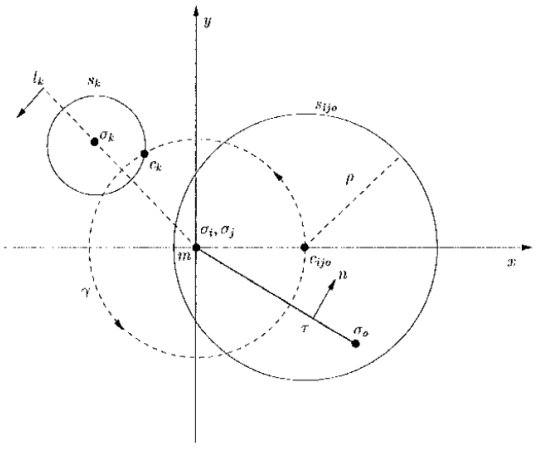


Fig. 3. The Ball Pivoting Algorithm in 2D. (a) A circle of radius ρ pivots from sample point to sample point, connecting them with edges. (b) When the sampling density is too low, some of the edges will not be created, leaving holes. (c) When the curvature of the manifold is larger than $1/\rho$, some of the sample points will not be reached by the pivoting ball, and features will be missed.

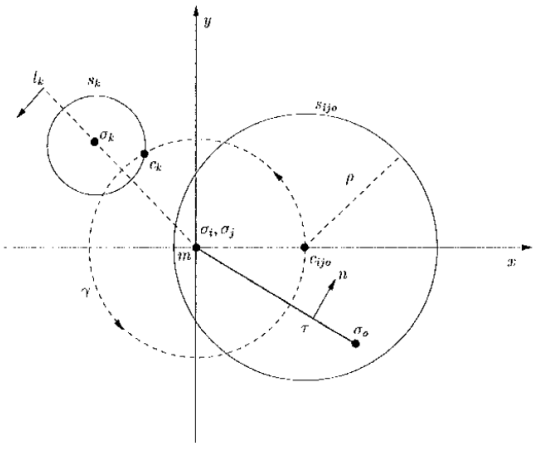
BPA - pridanie nového vrchola k pivotovanej hrane

- Uvažujme, že pivotovaná hrana je $\sigma_i\sigma_j$ (je stred je m) a sféra s polomerom ρ a stredom c_{ijo} vytvorila $\triangle\sigma_i\sigma_j\sigma_o$ s normálovým vektrom n.
- Rotáciou sféry (tak aby zachovala kontakt s hranou $\sigma_i\sigma_j$), jej stred c_{ijo} opíše kružnicu γ.
- Predpokladajme, že počas tohto pohybu sféra narazí na bod σ_k.
- Potom stred sféry po dotknutí bodu σ_k dostaneme ako priesečník $c_k = s_k \cap \gamma$, kde s_k je kružnica, ktorú sme dostali ako prienik sféry s polomerom ρ a stredom v σ_k a roviny, obsahujúcej kružnicu γ.



BPA - pridanie nového vrchola k pivotovanej hrane

- σ_k je hľadaný tretí vrchol nového trojuholníka.
- V praxi ho nájdeme nasledovne:
  - Uvažujeme všetky body mračna σ_x v 2ρ okolí bodu m.
  - Bod c_x určíme ako stred sféry, dotýkajúcej sa bodov σ_i, σ_j a σ_x. (Sú tieto body určené jednoznačne?)
  - Každý bod c_x leží na kružnicovej trajektórii γ so stredom m, a teda vieme ho vypočítať ako prienik ρ-sféry so stredom σ_x a vyberáme ten priesečník, ktorý sa vyskytne ako prvý, rešpektujúc smer trajektórie γ.



BPA - ako zvoliť začiatočný trojuholník (seed)?

- Vyberieme ľubovoľný bod σ, ktorý sa ešte nevyskytol v rekonštruovanej triangulácii.
- Zoradíme všetky dvojice bodov σ_a, σ_b podľa vzdialenosti od σ (treba uvažovať všetky?).
- Zostrojíme potenciálny $\triangle\sigma\sigma_a\sigma_b$:
  - Najprv zistíme či všetky vrcholové normálové vektory sú orientované konzistentne.
  - Potom zistíme, či ρ-sféra so stredom vo vonkajšom polpriestore sa dotýka všetkých troch vrcholov a či neobsahuje žiadny ďalší bod z mračna.
- Ak tento test neprejde, pokračujeme prehľadávaním zoznamu dvojíc vrcholov.
- Ak prejde, postupujeme pivotovaním po hrane podľa predošlého kroku.

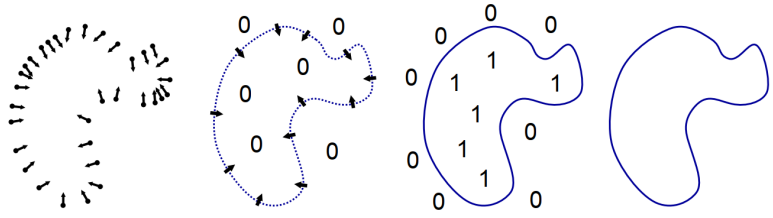
BPA - modifikácie, rozšírenia, závery

- Pri mračnách s obrovským množstvom vrcholov ho môžeme nakrájať na menšie ”kúsky”.
- Nakoľko BPA je citlivé na hustotu vzorkovania, diery vzniknuté v pletive môžeme zaplátať pustením opätovného behu avšak s väčším polomerom ρ.
- Hlavnou výhodou je robustnosť algoritmu a iba jeden vstupný parameter.
- Dá sa odôvodniť, že vzniknuté pletivo je homeomorfné k pôvodnej variete a leží v ohraničenej vzdialenosti.

Poisson surface reconstruction – Poissonova rekonštrukcia plochy – myšlienka

- VSTUP: Mračno bodov s asociovanými normálovými vektormi
- VÝSTUP: Vodotesné pletivo, aproximujúce pôvodné mračno

- **Gradient indikátorovej funkcie telesa je nulový všade, okrem jeho hranice, kde ukazuje normálovým smerom.**



Poisson surface reconstruction – postup

- Definujme indikátorovú funkciu telesa

$$\chi(\mathbf{x}) = \begin{cases} 1 & \text{vnútri telesa} \\ 0 & \text{von z telesa} \end{cases}$$

- Jej gradientom je vektorové pole, ktoré má nenulové hodnoty na hranici, t. j. tieto vektory ukazujú v normálovom smere. Preto

$$\nabla\chi \approx \mathbf{V},$$

kde $\mathbf{V}$ je vektorové pole získané zo vstupných normálových vektorov.

Poisson surface reconstruction – postup

- Preto, cieľom Poissonovej rekonštrukcie je nájsť takú indikátorovú funkciu χ, ktorá najlepšie aproximuje naše vstupné vektorové pole $\mathbf{V}$. Čiže potrebujeme v zmysle metódy najmenších štvorcov riešiť úlohu

$$\text{minimize} \|\nabla\chi - \mathbf{V}\|^2.$$

- To sa vďaka identite $\nabla \cdot \nabla\chi = \Delta\chi$ zredukuje na riešenie **Poissonovej rovnice** (podľa toho je názov)

$$\Delta\chi = \nabla \cdot \mathbf{V}.$$

Poisson surface reconstruction – postup

- Riešenie Poissonovej rovnice (detaily možno nájsť v pôvodnom článku) sa vykoná ”diskrétne”, t. j. pre vstupné mračno bodov sa zostrojí **adaptívny octree** (väčšia hĺbka znamená lepší detail výslednej plochy). Následne sa rieši riedky systém rovníc v uzloch octree
- Následne sa extrahuje izoplocha (zvyčajne pre $\chi = 0.5$ pomocou algoritmu **pochodujúcich kociek**.

Poisson surface reconstruction – výhody a limitácie

- Výhodou tohto prístupu je, že produkuje vodotesné plochy, dokonca i pre mračná, ktoré obsahujú diery.
- Je robustný aj vzhľadom na šum vo vstupnom mračne.
- Vďaka využitiu diskretizácie pomocou octree dokáže spracovať aj veľké mračná.

- Nevýhodou je, že potrebujeme mať k dispozícii aj asociované normálové vektory.
- Môžu sa strácať niektoré detaily (napr. ostré hrany), keďže má tendenciu k vyhladzovaniu (oversmoothing).

Screened Poisson surface reconstruction – myšlienka

- Problém s oversmoothingom rieši modifikovaná verzia tohto algoritmu, ktorá sa v súčasnosti považuje za SoTA, presnejšie Screened Poisson surface reconstruction.
- Jej myšlienkou je, že namiesto minimalizácie

$$\|\nabla\chi - \mathbf{V}\|^2.$$

sa minimalizuje (v zjednodušenej forme)

$$\int \|\nabla\chi - \mathbf{V}\|^2 + \gamma \sum_i (\chi(p_i) - 1)^2.$$

To znamená, že sa pridá korekčný (screen) parameter γ, ktorý zohľadňuje pozíciu vstupných bodov mračna.

A ako dopočítať normálové vektory?

- Predpokladajme, že hľadáme asociovaný normálový vektor $\mathbf{n}_0$ pre bod P_0.
- Pre tento bod zostavíme množinu $\{P_1, ..., P_k\}$ ako k najbližších susedov (napr. prehľadávaním k-D stromu.
- Následne zostrojíme ťažisko C, ktoré bude bodom najlepšie aproximujúcej roviny (v zmysle MNŠ) ako

$$C = \frac{1}{k+1} \sum_{i=0}^{k} P_i.$$

A ako dopočítať normálové vektory?

- Potom ťažisko C odčítame od jednotlivých bodov P_i:

$$Q_i = P_i - C.$$

- Ďalej zostavíme kovariačnú maticu

$$\mathbf{C} = \frac{1}{k+1} \sum_{i=0}^{k} Q_i Q_i^\top.$$

- Matica $\mathbf{C}$ nám zakóduje rozloženie bodov v každom smere

A ako dopočítať normálové vektory?

- Vypočítame vlastné čísla a vlastné vektory kovariačnej matice C

$$\mathbf{C}\mathbf{v}_j = \lambda_j \mathbf{v}_j, \quad j = 1, 2, 3.$$

- Vlastné vektory $\mathbf{v}_j$ nám určujú **hlavné smery** pre naše dáta.
- Vlastné čísla λ_j nám indikujú varianciu pozdĺž každého hlavného smeru.

- Vlastné vektory prislúchajúce prvým **dvom najväčším** vlastným číslam nám určujú najlepšie aproximujúcu rovinu (keďže tam je variancia dát najväčšia)
- Normálový vektor $\mathbf{n}_0$ preto dostaneme ako vlastný vektor prislúchajúci **najmenšiemu** vlastnému číslu.