

Pontfelhő alapú hossz tengely keresés OpenGL GLSL nyelven

Largest principal axis search based on point clouds, utilizing OpenGL GLSL

HORVÁTH Gergely¹
Dr. ERDŐS Ferenc Gábor^{1,2}

- 1) MTA SZTAKI 1111 Budapest, Kende utca 13-17., Telefon:: +36-1-279-6000, Fax: +36-1-466-7503
- 2) BME Gyártástudomány és –technológia Tanszék., 1111 Budapest, Műegyetem rkp. 3. T ép. 4. em. Telefon: +36-1-463-2515, Fax: +36-1-463-3176

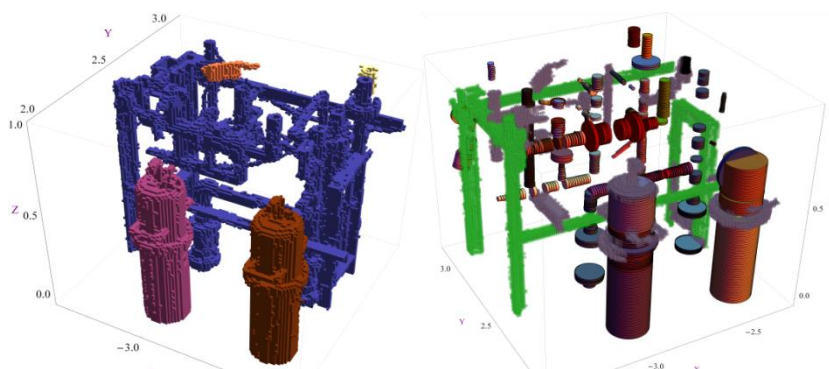
The objective of our research is to determine the largest principal axis of a long, extruded objects, based on its measured point cloud. Determining the principal axis is a common task, solved by calculating the eigenvectors of the inertia matrix in most cases. Although it is the mathematically correct way, there are cases---because of measurement flaws---when it cannot be used. In this paper a method to address these problems is discussed, and an effective implementation is shown.

Kutatásunk célja hosszú objektumok hossz tengelyének meghatározása, mért pontfelhők segítségével. Ez a feladat gyakran felmerül és a megoldása többnyire a másodrendű nyomatéki mátrix sajátértékének meghatározására vezet vissza. Bár ez a megközelítés matematikai értelemben helyes, mérési hibából eredő problémák kezelésére alkalmatlan. Ebben a cikkben egy olyan megközelítést tárgyalunk, amely képes ezen hibák kikerülésére, illetve bemutatunk egy működő implementációt.

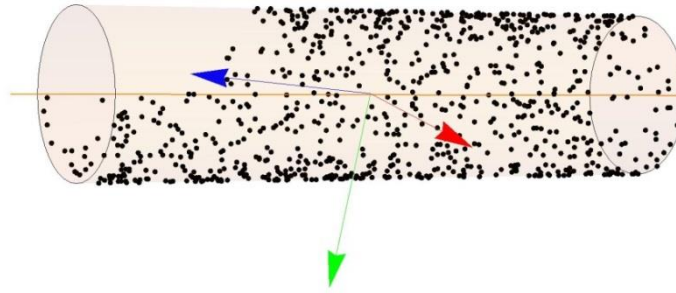
Axis, point cloud, GPU, OpenGL, extruded object

1. Bevezetés

Napjaink gyártási folyamataiban a gyártási pontosság és a gyártási idő két fontos szempont. A precíz naplózás, minőségellenőrzés és dokumentáció pontos és friss adatokat kíván a folyamatról és a befejezett termékről. A kutatás motivációját egy partner vállalat adta, amely – többek között – üzemek építésével, áttervezésével, karbantartásával is foglalkozik. Amikor ekkora méretű és ilyen komplexitású objektumokról van szó, elkerülhetetlen, hogy a tervezett és megvalósult állapotok eltérnek egymástól. A valódi és modell állapotokat egymáshoz kell illeszteni, majd a modellt ennek megfelelően átalakítani. Ennek az illesztési, felismerési feladatnak egyik lépése, hogy csövek, tartószerkezeti profilok keresztmetszete alapján azonosítjuk a típusukat. Az 1. ábra egy víztisztító pontfelhőjének részleges feldolgozását mutatja, illetve a már illesztett hengereket (egyéb okokból itt a hengerek egy-egy szelete látható).



1. ábra Víztisztító pontfelhőjének részleges feldolgozása



2. ábra Pontfelhő aminél a PCA hibás eredményt ad

A keresztmetszet meghatározásának előfeltétele, hogy a szerkezeti elem hossz tengelyét meghatározzuk. A hossz tengely ismeretében a merőleges metszet segítségével az objektum típusának felismerése már 2D-ben történik. A hossz tengely keresésére korábban kidolgozásra került egy új megközelítés [1], amelynek itt egy implementációját mutatjuk be. Az utolsó fejezet valós pontfelhőkön végzett tesztek eredményét tartalmazza.

1.1 Jelenleg használt eljárások

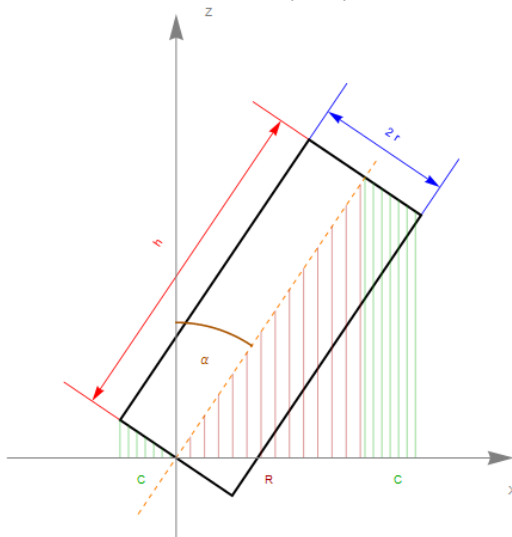
A főkomponens-analízis (angolul Principal Component Analysis, röviden PCA, [5]) egy gyakori eljárás pontfelhők hossz tengelyének meghatározására ([2][3][6]). Ez egy statisztikai módszer, amely nagyméretű, több dimenziós adatpontok főkomponenseinek meghatározására szolgál. Egy pontfelhő esetében a legnagyobb főkomponens éppen az általunk keresett fő tengely lesz. Az eljárás alapját, [4] szerint, a mechanikai tehetetlenségi nyomaték számítás adja. A tehetetlenségi nyomaték, a szilárd testek egy tulajdonsága, a tömeggel analóg forgó mozgás esetén. A tehetetlenségi nyomaték számítás és a PCA nagyon hasonló eredményt hoz pontfelhők esetében. Az eljárások hibája akkor mutatkozik meg, amikor a pontfelhő technikai okokból sérült. A 2. ábra egy ilyen pontfelhőt és a PCA által számolt fő irányokat mutatja.

1.2 Új megközelítés

Az új módszer egy tisztán geometriai megközelítés. Az eljárás lényege, hogy egy testet (esetünkben pontfelhőt), úgy orientálunk, hogy a fő tengelye párhuzamos legyen a Z tengellyel. Ehhez egy ellenőrzést dolgoztunk ki. A testet levetítjük az X-Y síkra, majd mérjük a vetület területét. Ezt mutatja a 4. ábra. A legkisebb területhez tartozó orientációban a test fő tengelye párhuzamos a Z tengellyel. Pontfelhők esetében a terület nem értelmezhető, ilyenkor egy rácsot definiálunk az X-Y síkon és azon négyzetek területének összegét számoljuk, amelyek tartalmaznak vetített pontot.

2. Algoritmus elméleti helyességének vizsgálata

Ebben a fejezetben bemutatjuk, hogy az általunk vázolt eljárás valóban megfelelő eredményt ad teljes henger pontfelhőjére, ha a henger „hosszának” tekinthető. Az állítás a következő: tekintjük egy henger sík vetületét. Ha $\frac{r}{h} \leq \frac{2}{\pi}$, ahol r a henger sugara és h a henger hossza, abban az esetben a vetület



3. ábra Henger metszete az X-Z síkban

területe akkor lesz minimális, ha a vetítés irány párhuzamos a henger tengelyével.

Legyen X-Y sík a vetítés síkja és Z a vetítés tengelye, a henger alapkörének középpontja pedig essen az origóba. A henger orientációja két szöggel leírható, a Z és az Y tengelyek körüli forgatással. A Z tengely körüli forgatásnak nincs hatása a terület nagyságára, ezért ettől eltekintünk. Nagyságát jelöljük α -val. A 3. ábra egy X-Z síkbeli metszetet mutat. A C-vel jelölt terület nagysága

$$C = \frac{|\cos \alpha| r^2 \pi}{2},$$

míg az R-el jelölt terület nagysága

$$R = |\sin \alpha| h 2r.$$

Ezek alapján a vetület területe

$$A = 2C + R = |\cos \alpha| r^2 \pi + |\sin \alpha| h 2r.$$

Ennek a függvénynek keressük a minimum helyét. Ehhez képezzük az első deriváltját, majd az így kapott egyenletnek vesszük a zérus helyeit. Egyelőre eltekintünk az abszolút értékektől, azt később fogjuk figyelembe venni.

$$\frac{\partial A}{\partial \alpha} = \frac{\partial(2hr \sin \alpha + \pi r^2 \cos \alpha)}{\partial \alpha} = 2hr \cos \alpha - \pi r^2 \sin \alpha = 0,$$

rendezés után ezt kapjuk

$$\frac{\pi r}{2h} = \frac{\cos \alpha}{\sin \alpha}.$$

Az osztások elvégzése miatt az alábbi eseteket meg kell vizsgálni:

- $r = 0, 2h = 0$, ellentmond a fizikai képnek, amivel dolgozunk,
- $\sin \alpha = 0$, nem elégíti ki a kezdeti egyenletet.

Ezek alapján a lehetséges szélsőértékek

$$\alpha = \cot^{-1} \frac{\pi}{2} q,$$

ahol $q = \frac{r}{h}$. Ezt az értéket visszahelyettesítjük a második deriváltba és megvizsgáljuk a kapott eredmény előjelét. Mivel a $\cot^{-1} \alpha$ függvény π szerint periodikus, a terület függvényünk pedig 2π szerint, ezért két α értéket kell vizsgálni (α_1 és α_2) úgy, hogy $\alpha_2 = \alpha_1 + \pi$. Az értékek

$$\left. \frac{d^2 A}{d\alpha^2} \right|_{\alpha=\alpha_1+2k\pi} = -\frac{4hr}{\pi q \sqrt{\frac{4}{\pi^2 q^2} + 1}} - \frac{\pi r^2}{\sqrt{\frac{4}{\pi^2 q^2} + 1}} \leq 0$$

$$\left. \frac{d^2 A}{d\alpha^2} \right|_{\alpha=\alpha_2+2k\pi} = \frac{4hr}{\pi q \sqrt{\frac{4}{\pi^2 q^2} + 1}} + \frac{\pi r^2}{\sqrt{\frac{4}{\pi^2 q^2} + 1}} \geq 0$$

A becült előjelek az ismert fizikai képből erednek, hiszen minden változó értéke pozitív. Legyen $\alpha_1 \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. Figyelembe véve a probléma fizikai képéből eredő korlátokat, a keresett szög érték $\alpha = \alpha_1$, ami egy lokális maximum.

Most tudjuk figyelembe venni az abszolút érték jelentőségét is az eredeti egyenletben. Ha egy függvény bizonyos alfüggvénye abszolút értékben van (az alfüggvény legyen $f(x)$), akkor az $f(x) = 0$ helyeken a függvényben törés lehet, ami lokális szélsőértékek kialakulásához vezet. Ez azt jelenti, hogy ezeket a pontokat is vizsgálni kell. Az abszolút értékben álló függvények gyökei az alábbiak:

$$\sin \alpha = 0 \rightarrow \alpha = k\pi,$$

$$\cos \alpha = 0 \rightarrow \alpha = \frac{\pi}{2} + k\pi.$$

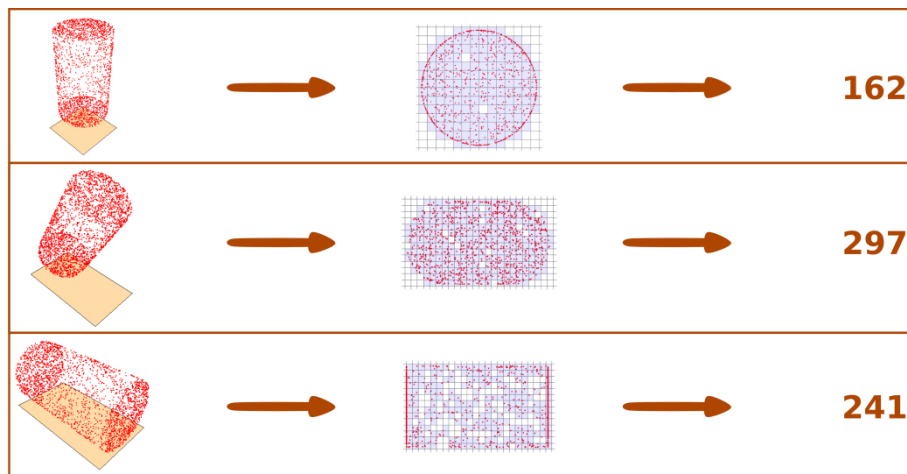
Mivel csak a $[0, \pi]$ tartomány érdekes, ezért a következő egyenlőtlenséget kell megoldani: $A(0) < A\left(\frac{\pi}{2}\right)$, ami rendezés után az alábbira adódik:

$$\frac{r}{h} < \frac{2}{\pi},$$

ahol $r > 0$ és $h > 0$. Vagyis, ha az egyenlőtlenség igaz, akkor lesz minimális a vetület területe ha a henger tengelyével párhuzamosan vetítünk.

3. Implementáció

Az algoritmus implementációja OpenGL GLSL nyelven történt. Mivel a síkba vetítés egy renderelési feladat, adta magát a programozási környezet választása. Feladatunk a különböző orientációjú pontfelhők textúrába fotózása, a textúrában lévő alakzat területének számítása, majd a számolt eredmények közül a legkisebb terület kiválasztása. Ezt mutatja a 4. ábra. Az összes orientáció vizsgálatához az egyik tengely körül 360° , míg a másik tengely körül 180° -al forgatjuk körbe a testet. A keresési teret először mind a két irányba 64-szer mintavételeztük, majd a legkisebb területhez tartozó orientáció környezetében ismét felosztottuk a tartományt 64 részre.



4. ábra Objektum hossz tengelyének keresése

A teszteléshez különböző nagyságú ipari pontfelhőket használtunk. A tesztesetek nagyságát és a hozzájuk tartozó futási időket mutatja az 1. táblázat. A tesztelésre használt számítógép egy Intel Core i5-2500K@3300MHz processzorral, 16GB memóriával, NVIDIA GeForce GTS-450 videokártyával szerelt számítógép. A teszteket Windows 7 Professional operációs rendszeren végeztük el, az NVIDIA driver verziója 347.62.

1. táblázat Teszt esetek futási eredményei

Azonosító	Alakzat típusa	Pontok száma	Számítási idő [s]
1	Henger	1000	0,66
2	U-szelvény	110422	2,252
3	U-szelvény	102262	2.072
4	U-szelvény	76676	1.697
5	U-szelvény	123960	2.603
6	U-szelvény	120316	2.366

4. Köszönetnyilvánítás

A kutatást az Országos Tudományos Kutatási Alapprogram (OTKA) 113038 támogatása tette lehetővé.

5. Irodalomjegyzék

- [1] G. Erdős, G. Horváth, L. Kovács, T. Szirányi, and J. Váncza. „Progress Recognition for Plant Construction”, Technical report, Hitachi YRL, MTA SZTAKI, 2013. Március 29.
- [2] Luis Malagón-Borja, Olac Fuentes, Object detection using image reconstruction with PCA, Image and Vision Computing, Volume 27, Issues 1–2, 1 January 2009, Pages 2-9
- [3] Nagabhushan, P., D. S. Guru, and B. H. Shekar. "Visual learning and recognition of 3D objects using two-dimensional principal component analysis: A robust and an efficient approach." Pattern Recognition 39(4) (2006): 721-725.
- [4] Pearson, K. „On Lines and Planes of Closest Fit to Systems of Points in Space”, Philosophical Magazine, 1901/11, 559-572
- [5] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, 2nd Edition, Wiley, 2000
- [6] Yun-Seok Lee, Han-Suh Koo, Chang-Sung Jeong, A straight line detection using principal component analysis, Pattern Recognition Letters, Volume 27, Issue 14, 15 October 2006, Pages 1744-1754