VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

ALGORITMUS ICP (ITERATIVE CLOSEST POINT) PRO SESAZOVÁNÍ MRAKU BODŮ

THE ALGORITHM ICP (ITERATIVE CLOSEST POINT) FOR POINT CLOUD REGISTRATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — **JIŘÍ WRÓBEL**

AUTHOR

VEDOUCÍ PRÁCE — **Mgr. JANA PROCHÁZKOVÁ, Ph.D.**

SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav matematiky
Student:	**Jiří Wróbel**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	**Mgr. Jana Procházková, Ph.D.**
Akademický rok:	2017/18

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Algoritmus ICP (Iterative Closest Point) pro sesazování mraku bodů

Stručná charakteristika problematiky úkolu:

Pokud máme k dispozici 3D data z různých pohledů, lze vypočítat optimální rotaci a posunutí, které tyto dva marky bodů sesadí, aby vzdálenost mezi nimi byla nejmenší ve smyslu nejmenších čtverců. K tomu slouží například algoritmus ICP, který bude hlavní náplní práce.

Cíle bakalářské práce:

1. Popis algoritmu ICP.
2. Shrnutí možných zlepšení algoritmu.
3. Programové zpracování a otestování s různými parametry na reálných datech ze 3D skeneru. Popis získaných výsledků.

Seznam doporučené literatury:

SZELISKI, Richard. Computer vision: algorithms and applications. London: Springer, c2011. Texts in computer science. ISBN 978-1-84882-934-3.

WEINMANN, Martin. Reconstruction and Analysis of 3D Scenes. Switzerland: Springer, 2016.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

Abstrakt
Tato bakalářská práce se zabývá algoritmem ICP (Iterative Closest Point) pro sesazování mračna bodů. Zaměřuje se zejména na popis klasického algoritmu ICP. Tento algoritmus je modifikován použitím KD-stromu. Práce dále obsahuje popis jednoho možného vylepšení pomocí minimalizace vzdálenosti bodu od roviny. Výsledky byly testovány na reálných datech.

Summary
This bachelor's thesis deals with the ICP (Iterative Closest Point) algorithm for point cloud registration. It focuses mainly on the classic ICP algoritm. This algorithm is modified using KD-tree. The thesis also describes one the possible improvements using the point-to-plane minimatization. The results were tested on real-world data set.

Klíčová slova
ICP, mračno bodů, KD-strom

Keywords
ICP, point cloud, KD-tree

WRÓBEL, J.*Algoritmus ICP (Iterative Closest Point) pro sesazování mraku bodů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 39 s. Vedoucí Mgr. Jana Procházková, Ph.D.

Prohlašuji, že jsem bakalářskou práci Algoritmus ICP (Iterative Closest Point) vypracoval samostatně pod vedením Mgr. Jany Procházkové, Ph.D. a s použitím materiálů uvedených v seznamu literatury.

Jiří Wróbel

Děkuji své vedoucí Mgr. Janě Procházkové, Ph.D. za odborné vedení, cenné rady a připomínky při zpracování této bakalářské práce.

Jiří Wróbel

Obsah

1 Úvod

Iterative closest point (ICP), někdy též nazývaný iterative corresponding point, je algoritmus umožňující skládání dvou množin křivek, ploch nebo dvou mračen bodů. Tento algoritmus byl poprvé popsán Yangem Chenem a Gérardem Medionini v roce 1991 [1] a Paulem J. Beslem a Neilem D. McKayem v roce 1992 [2].

Algoritmus ICP je jedna z nejrozšířenějších metod sesazování mračen bodů. Začíná s dvěma modely a počátečním odhadem jejich vzájemné polohy. Poté tuto polohu opakovaně upřesňuje hledáním korespondujících bodů a minimalizací chyby, dokud tato chyba není dostatečně malá.

Algoritmus ICP je široce využíván pro sesazování dat ze 3D skeneru, kdy je výsledný objekt složen z několika dílčích skenů. Pro tento účel je použit i v této bakalářské práci.

Během posledních let byla navržena řada variant a vylepšení algoritmu ICP pro sesazování mračna bodů. Práce je zaměřena zejména na klasický ICP alogritmus, který při určování korespondence bodů předpokládá, že odpovídající si body jsou ty nejbližší, a následně minimalizuje součet čtverců vzdáleností mezi těmito body. Je zde popsáno i jedno z možných zlepšení algoritmu při minimalizaci chyby. V této variantě je místo vzdálenosti bodu od bodu minimalizována vzdálenost bodu od roviny.

Práce je rozdělena do pěti kapitol. Kapitola 2 obsahuje základní poznatky teorie matic nutné k odvození algoritmu ICP pro sesazování mračna bodů.

Kapitola 3 obsahuje popis samotného algoritmu ICP pro sesazování mračna bodů. Nejdříve je zde popsán způsob získávání dat a jejich ukládání do datové struktury KD--strom, který slouží k urychlení vyhledávání nejbližších bodů. Dále je pozornost věnována dvěma důležitým bodům algoritmu, nalezení korespondence bodů a zejména minimalizaci vzdáleností mezi těmito body. Nakonec následuje popis dvou maticových rozkladů, singulárního a Choleského. Oba jsou použity při minimalizaci vzdálenstí, každý však v jiné variantě algoritmu.

V kapitole 4 je popsán výsledný program, který je určen pro sesazování mračna bodů pomocí algoritmu ICP.

Kapitola 5 shrnuje celou práci a na reálných datech hodnotí funkčnost programu.

2 Matematický aparát

K odvození algortimu ICP je potřeba některých poznatků lineární algebry, konkrétně maticového počtu. Proto zde uvedeme některé základní definice teorie matic. Tyto definice lze nalézt například v [3] nebo [4].

Definice 2.1 *Nechť X je neprázdná množina reálných čísel a m, n přirozená čísla. Matice $\mathbf{A}$ typu (m, n) nad X je zobrazení množiny $1, \ldots, m \times 1, \ldots, n$ do množiny X. Matici A typu (m, n) budeme zapisovat ve tvaru*

$$\mathbf{A} = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{pmatrix},$$

nebo jen krátce $\mathbf{A} = (a_{ij})$, kde a_{ij} je prvek na i-tém řádku a j-tém sloupci. Prvky $a_{11}, a_{22}, \ldots$ nazveme hlavní diagonálou. Matici typu (n, n) nazveme čtvercovou maticí řádu n.

Definice 2.2 *Nechť $\mathbf{A} = (a_{ij})$ je matice typu (m, n) s prvky $a_{ij} \in X$. Matice $\mathbf{A}$ se skládá z m řádků $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m$, délky n: $\mathbf{r}_i = (a_{i1}, a_{i2}, \ldots, a_{in})$. Pro $c_1, c_2, \ldots, c_m \in X$ lineární kombinací řádků nazveme vektor*

$$c_1\mathbf{r}_1 + c_2\mathbf{r}_2 + \cdots + c_m\mathbf{r}_m := (c_1a_{11} + \cdots + c_ma_{m1}, \ldots, c_1a_{1n} + \cdots + c_ma_{mn}).$$

Definice 2.3 *Řádky $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m$ matice $\mathbf{A}$ nazveme lineárně nezávislé, pokud jediná jejich lineární kombinace, která dává nulový vektor $\mathbf{0} = (0, \ldots, 0)$, je kombinace nulová, tj.*

$$c_1\mathbf{r}_1 + c_2\mathbf{r}_2 + \cdots + c_m\mathbf{r}_m = \mathbf{0} \implies c_1 = c_2 = \cdots = c_m = 0.$$

Definice 2.4 *Hodnost matice $\mathbf{A}$ je počet jejích lineárně nezávislých řádků. Značíme ji $h(\mathbf{A})$. Hodnost nulové matice položíme $h(\mathbf{0}) = 0$.*

Definice 2.5 *Nechť $\mathbf{A} = (a_{ij})$ je matice typu (m, n). Maticí k ní transponovaná je matice $\mathbf{A}^T$ typu (n, m) s prvky (a_{ji}), tj. $\mathbf{A}^T = (a_{ji})$.*

Definice 2.6 *Čtvercovou matici $\mathbf{A}$ řádu n nazveme diagonální, pokud má na hlavní diagonále libovolné nenulové prvky a mimo hlavní diagonálu nuly. Diagonální matici budeme psát ve tvaru $\mathbf{A} = diag(a_{11}, a_{22}, \ldots, a_{nn})$.*

Definice 2.7 *Ortogonální maticí nazveme čtvercovou matici $\mathbf{A}$, pro kterou platí*

$$\mathbf{A}^T\mathbf{A} = I,$$

kde I je jednotková matice, tj. $I = diag(1, \ldots, 1)$.

Definice 2.8 *Nechť $\mathbf{A}$ je čtvercová matice řádu n. Pak číslo $\sum_{i=1}^{n} a_{ii}$ nazveme stopou matice. Stopu matice A značíme $tr(\mathbf{A})$.*

Definice 2.9 *Matici $\mathbf{A}$ nazveme horní trojúhelníkovou maticí, pokud platí $a_{ij} = 0$ pro všechna $i > j$.*

Definice 2.10 *Matici* $\mathbf{A}$ *nazveme dolní trojúhelníkovou maticí, pokud platí* $a_{ij} = 0$ *pro všechna* $i < j$.

Definice 2.11 *Nechť* $\mathbf{A}$ *je matice řádu* n. *Číslo* $\lambda \in \mathbb{C}$ *se nazývá vlastí čisla matice* $\mathbf{A}$, *jestliže existuje nenulový vektor* $\mathbf{x} \in \mathbb{C}^n$ *tak, že* $\mathbf{Ax} = \lambda\mathbf{x}$. *Vektor* $\mathbf{x}$ *se nazývá vlastní vektor příslušný k* λ.

Definice 2.12 *Symetrickou matici* $\mathbf{A}$ *řádu* n *nazveme pozitivně definitní, jestliže pro každý nenulový vektor* $\mathbf{x} \in \mathbb{R}^n$ *platí*

$$\mathbf{x}^T\mathbf{Ax} > 0.$$

Pro ověření pozitivní definitnoti slouží následující věta, známa jako Sylvestrovo kritérium.

Věta 2.1 *Symetrická matice* $\mathbf{A}$ *řádu* n *je pozitivně definitní, právě když jsou kladné determinanty všech hlavních rohových submatic* $a_{ij}{}_{i,j=1}^{k}$, *pro* $k = 1, 2, \ldots, n$, *tj. když platí*

$$a_{11} > 0, \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \quad \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} > 0, \ldots, \begin{vmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nn} \end{vmatrix} > 0.$$

Důkaz této věty je uveden např v [5].

3 ICP

Iterative closest point (ICP) je metoda široce využívána pro sesazování mračen bodů v případech, kdy je znám počáteční odhad relativní polohy. Tato metoda byla poprvé popsána Yang Chenem a Gérardem Medioni [1] a Paulem J. Beslem a Neilem D. McKayem [2].

Mračno bodů (v angličtině point cloud), je neuspořádaná množina bodů ve dvou nebo třídimenzionálním prostoru, přičemž každý z nich je dán svými souřadnicemi v daném systému souřadnic. Poloha každého bodu P v prostoru je jednoznačně určena souřadnicemi (x, y), případně (x, y, z). Navíc musí být definována jednotka systému a musí být definována metrika, pomocí níž lze měřit vzdálenost mezi body.

Algoritmus ICP pracuje na vstupu se dvěma mračny bodů M, D o velikostech N_m, N_d a počátečním odhadem jejich vzájemné polohy. Mračno M reprezentuje známý model a mračno D reprezentuje data, která chceme přidat. Počáteční odhad lze získat zaznamenáváním pozice skeneru, výpočtem hlavních os [6] nebo uživatelským vstupem. Každá iterace algoritmu má několik kroků:

1. Určení vzájemné korespondence bodů z mračen M a D.
2. Výpočet rotace $\mathbf{R}$ a translace $\mathbf{t}$ minimalizací vzdáleností mezi odpovídajícími si body.
3. Aplikace vypočítané transformace na body z mračna D.
4. Pokud je dosažená vzdálenost mezi odpovídajícími si body dostatečně malá, algorismus končí.

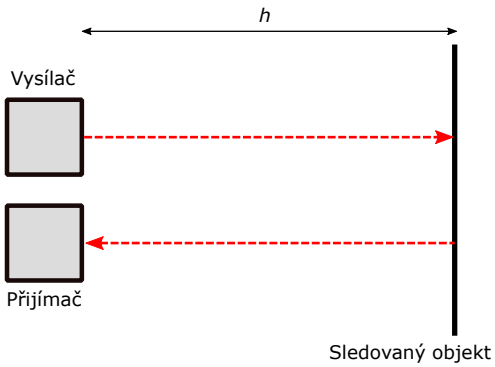
3.1 Získávání dat

Mračno bodů je výsledkem skenování trojrozměného objektu. Toto skenování může být prováděno několika způsoby. Mračna bodů použita v této práci byla získána za použití přístroje LIDAR, který zobrazen na obr. 3.1. Tato metoda je založena na měření doby šíření laserové paprsku. Přístroj nejprve paprsek vyšle, ten se následně odrazí od sledovaného objektu a poté se vrátí zpět. Tento princip je zobrazen na obr. 3.2. Na základě známé rychlosti šíření tohoto paprsku c a času t, který uběhl od vyslání, potom vypočítá vzdálenost h od sledováneho objektu jako

$$h = \frac{1}{2}ct.$$

Podrobně je tato metoda popsána v [7].

Obrázek 3.1: LIDAR



Obrázek 3.2: Princip LIDARU

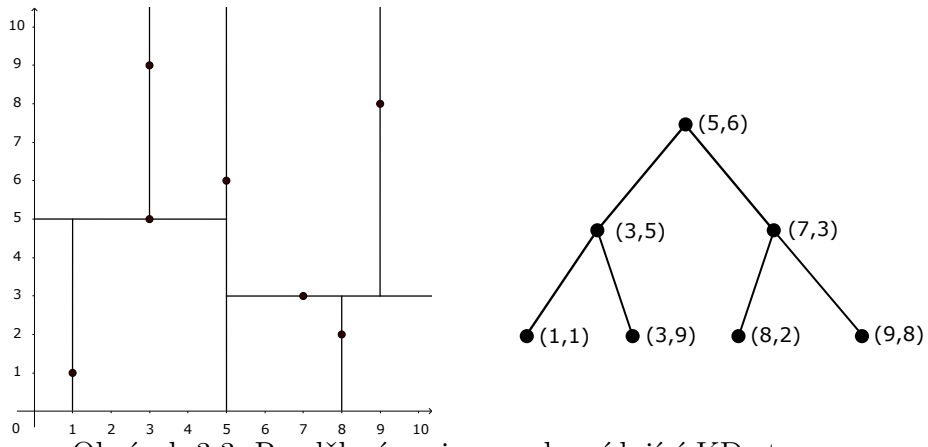
3.2 KD-strom

Pro urychlení hledání nejbližších bodů jsou data nejdříve uloženy do datové struktury KD-strom.

KD-strom patří mezi prostorové datové struktury sloužící pro ukládání dat v k-rozměrném prostoru. Při kostrukci KD-stromu je prostor postupně rozdělen nadrovinami kolmými k jednotlivým osám, přičemž za bod dělení se volí medián souřadnic bodů v příslušném podintervalu vzniklém v předchozích děleních. Prostor takto rozdělený nadrovinami a odpovídající KD-strom je možno vidět na obr. 3.3.

3.2.1 Vytváření KD-stromu

Vytváření KD-stromu probíhá rekurzivně. V každém kroku se vstupní body seřadí podle jedné z dimenzí a z nich se vybere medián jako bod dělení. Za bod dělení lze zvolit i jiný bod, medián je však nejpoužívanější. Tento postup se opakuje na obou vzniklých podprostorech. Dimenze, podle kterých se prostor dělí, se většinou cyklicky střídají, lze však použít i jiný systém. Můžeme osy volit například náhodně, při tomto postupu je však nutné tuto volbu ukládat.



Obrázek 3.3: Rozdělení roviny a odpovídající KD-strom

Následující algoritmus popisuje konstrukci KD-stromu.

Algoritmus 1 Konstrukce KD-stromu

```
1: function KDTREE(points, depth)
2:     axis := depth%k
3:     SORT(points, axis)
4:     median := LENGTH(points)/2
5:     node.location := points[median]
6:     node.leftChild := KDTREE(points[:median], depth + 1)
7:     node.rightChild := KDTREE(points[median + 1:], depth + 1)
8:     return node
9: end function
```

3.2.2 Vyhledání nejbližšího bodu v KD-stromu

KD-strom se velmi často používá při hledání nejbližšího bodu k danému bodu. Vyhledávání je efektivní, protože použití KD-stromu umožňuje eliminovat velké části prostoru. Průměrná časová náročnost hledání nejbližšího bodu v množině o n prvcích je $O(\log n)$, v nejhorším případě potom $O(n)$.

Algoritmus začíná v nejvyšším uzlu a postupuje rekurzivně dolů po stromu na levý nebo pravý uzel podle toho, zda je daný bod v dimenzi dělení menší nebo větší než bod dělení tohoto uzlu. Jakmile dojde algoritmus na konec stromu, označí bod dělení v posledním uzlu jako aktuální nejbližší bod a postupuje zpět nahoru po stromu. V každém uzlu potom kontroluje, zda nemůže být na druhé straně od bodu dělení bližší bod. Pokud může, hledání provede i tam. Algoritmus končí, když se vrátí zpět do nejvyššího uzlu.

3.3 Nalezení odpovídajicích dvojic bodů

Prvním krokem každé iterace algoritmu je nalezení odpovídajících si bodů.

Pro funkci algoritmu není nutné není nutné hledat korespondenci všech bodů z mračna D, je možné použít jen jejich část. Použité body můžeme vybírat rovnoměrně [8] nebo náhodně, a to s různým vzorkem v každé iteraci [9].

V další fázi je nutné k vybraným bodům z mračna D najít ty, které jim odpovídají v mračnu M. V klasickém algoritmu ICP jsou za odpovídající si body považovány ty nejbližší [2]. Tento typ algoritmu může být urychlen použitím KD-stromu [10]. Další možností je například hledání průsečíku paprsku vycházejícího z daného bodu ve směru normály v tomto bodě s mrakem M [1].

Než přistoupíme k dalšímu kroku algoritmu, můžeme některé páry bodů vyřadit. Můžeme vyřadit n procent párů s největší vzdáleností [11] nebo páry, jejichž vzdálenost je větší než daný násobek směrodatné odchylky všech vzdáleností [9]. Dále můžeme vyřadit páry, jejichž body leží na okraji mračen [8].

3.4 Minimalizace vzdáleností

Existuje několik způsobů, jak minimalizovat vzdálenosti mezi korespondujícími body a tím vypočítat ideální transformaci. Mezi nejpoužívanější patří minimalizace vzdáleností bodu od bodu a to pomocí SVD rozkladu [12] nebo pomocí kvaternionů [13]. Tento způsob popisuje část 3.4.1. Možným vylepšením algoritmu je minimalizace vzdáleností bodů od roviny, kterou popisuje část 3.4.2. Pro tento postup je však nutné znát normály mračna bodů. V případě testovacích dat by tak musely být vypočítány, a proto byla v programu použita minimlizace vzdáleností bodu od bodu s použitím singulárního rozkladu.

3.4.1 Vzdálenost bodu od bodu

V následující části popíšeme teoretický základ pro výpočet optimální matice rotace a vektoru translace mezi dvěma mračny bodů. Tento postup vychází z popisu v [14] a [15].

Pokud máme mračna bodů M a D a známe korespondenci bodů z mračna D k bodům v mračnu M, můžeme optimální rotaci $\mathbf{R}$ a translaci $\mathbf{t}$ vypočítat minimalizací chyby

$$E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_d} w_{i,j}\|\mathbf{m}_i - \mathbf{R}\mathbf{d}_j + \mathbf{t}\|^2,$$

kde $\mathbf{m}_i$ jsou body mračna M, $\mathbf{d}_j$ body mračna D a $w_{i,j}$ označuje váhy odpovídajících si bodů. Pro korespondující body je $w_{ij} = 1$, jinak $w_{ij} = 0$. Pokud tuto matici nahradíme sumou pouze přes korespondující body, můžeme tento výraz přepsat na

$$E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N} \|\mathbf{m}_i - \mathbf{R}\mathbf{d}_i + \mathbf{t}\|^2,$$

kde $N = \sum_{i=1}^{N_m}\sum_{j=1}^{N_d} w_{i,j}$. Tento výraz můžeme dále upravit. Označíme $\mathbf{c}_m$ a $\mathbf{c}_d$ těžiště jednotlivých mračen:

$$\mathbf{c}_m = \frac{1}{N}\sum_{i=1}^{N}\mathbf{m}_i,$$

$$\mathbf{c}_d = \frac{1}{N}\sum_{i=1}^{N}\mathbf{d}_i$$

a obě mračna posuneme do těžišť:

$$M' = \{\mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_m\}_{1,\dots,N},$$

$$D' = \{\mathbf{m}'_i = \mathbf{d}_i - \mathbf{c}_d\}_{1,\dots,N}.$$

Po dosazení dostáváme

$$E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N} \|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i - \underbrace{(\mathbf{t} - \mathbf{c}_m + \mathbf{R}\mathbf{c}_d)}_{\tilde{t}}\|^2$$

$$= \sum_{i=1}^{N}\|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\|^2 - 2\tilde{t}\sum_{i=1}^{N}(\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i) + \sum_{i=1}^{N}\|\tilde{\mathbf{t}}\|^2.$$

Suma bodů centrovaných do těžiště je nulová, proto je druhý člen v tomto výrazu nulový. Třetí člen minimalizujeme volbou $\tilde{\mathbf{t}} = 0$, platí tedy

$$\mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d.$$

K minimalizacizaci chyby tak stačí minimalizovat pouze první člen a tento výraz můžeme přepsat do tvaru

$$E(\mathbf{R}, \mathbf{t}) \propto \sum_{i=1}^{N} \|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\|^2.$$

Tento člen lze dále rozepsat jako

$$\begin{aligned}\|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\|^2 = &= (\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i)^T(\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i) = \\ &= (\mathbf{m}'^T_i - \mathbf{d}'^T_i\mathbf{R}^T)(\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i) = \\ &= \mathbf{m}'^T_i\mathbf{m}'_i - \mathbf{m}'^T_i\mathbf{R}\mathbf{d}'_i - \mathbf{d}'^T_i\mathbf{R}^T\mathbf{m}'_i + \mathbf{d}'^T\mathbf{R}^T\mathbf{R}\mathbf{d}'_i = \\ &= \mathbf{m}'^T_i\mathbf{m}'_i - \mathbf{m}'^T_i\mathbf{R}\mathbf{d}'_i - \mathbf{d}'^T_i\mathbf{R}\mathbf{m}'_i + \mathbf{d}'^T\mathbf{d}'_i.\end{aligned}$$

Protože $\mathbf{d}'^T_i$ má rozměr 1×3, $\mathbf{R}$ je matice řádu 3 a $\mathbf{m}_i$ má rozměr 3×1, člen $\mathbf{d}'^T_i\mathbf{R}\mathbf{m}'_i$ je skalár. Pro každý skalár a platí $a^T = a$ a tedy

$$\mathbf{d}'^T_i\mathbf{R}\mathbf{m}'_i = (\mathbf{d}'^T_i\mathbf{R}\mathbf{m}'_i)^T = \mathbf{m}'^T_i\mathbf{R}\mathbf{d}'_i.$$

Dostáváme tedy

$$\|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\|^2 = \mathbf{m}'^T_i\mathbf{m}'_i - 2\mathbf{m}'^T_i\mathbf{R}\mathbf{d}'_i + \mathbf{d}'^T\mathbf{d}'_i.$$

Odtud můžeme chybovou funkci vyjádřit následovně:

$$\begin{aligned}E(\mathbf{R}, \mathbf{t}) &\propto \sum_{i=1}^{N} (\mathbf{m}'^T_i\mathbf{m}'_i - 2\mathbf{m}'^T_i\mathbf{R}\mathbf{d}'_i + \mathbf{d}'^T\mathbf{d}'_i) = \\ &= \sum_{i=1}^{N} \mathbf{m}'^T_i\mathbf{m}'_i - 2\sum_{i=1}^{N} \mathbf{m}'^T_i\mathbf{R}\mathbf{d}'_i + \sum_{i=1}^{N} \mathbf{d}'^T\mathbf{d}'_i.\end{aligned}$$

První ani třetí člen nezávisí na $\mathbf{R}$, chybu tak můžeme přepsat do tvaru:

$$E(\mathbf{R}, \mathbf{t}) \propto -2\sum_{i=1}^{N} (\mathbf{m}'^T_i\mathbf{R}\mathbf{d}'_i).$$

Pro minimalizaci chyby nám nyní stačí maximalizovat výraz $\sum_{i=1}^{N_d} (\mathbf{m}'^T_i\mathbf{R}\mathbf{d}'_i)$. Protože platí

$$\begin{pmatrix}\mathbf{m}'^T_1 \\ \mathbf{m}'^T_2 \\ \vdots \\ \mathbf{m}'^T_N\end{pmatrix} \mathbf{R} \begin{pmatrix}\mathbf{d}'_1 & \mathbf{d}'_2 & \dots & \mathbf{d}'_N\end{pmatrix} = \begin{pmatrix}\mathbf{m}'^T_1 \\ \mathbf{m}'^T_2 \\ \vdots \\ \mathbf{m}'^T_N\end{pmatrix} \begin{pmatrix}\mathbf{R}\mathbf{d}'_1 & \mathbf{R}\mathbf{d}'_2 & \dots & R\mathbf{R}\mathbf{d}'_N\end{pmatrix} =$$

$$= \begin{pmatrix}\mathbf{m}'^T_1\mathbf{R}\mathbf{d}'_1 & & & \star \\ & \mathbf{m}'^T_2\mathbf{R}\mathbf{d}'_2 & & \\ & & \ddots & \\ \star & & & \mathbf{m}'^T_N\mathbf{R}\mathbf{d}'_N\end{pmatrix},$$

můžeme tento člen přepsat do tvaru

$$\sum_{i=1}^{N} (\mathbf{m}_i'^T \mathbf{R} \mathbf{d}_i') = tr(\mathbf{X}^T \mathbf{R} \mathbf{Y}),$$

kde $\mathbf{X}$ je matice $3 \times N$, která je tvořena sloupečky $\mathbf{m}_i'$, a $\mathbf{Y}$ je matice $3 \times N$ tvořená sloupečky $\mathbf{d}_i'$. Protože platí

$$tr(\mathbf{AB}) = tr(\mathbf{BA})$$

pro všechny matice $\mathbf{A}$, $\mathbf{B}$, pro které má tento výraz smysl, můžeme psát

$$tr(\mathbf{X}^T \mathbf{R} \mathbf{Y}) = tr(\mathbf{R} \mathbf{Y} \mathbf{X}^T).$$

Označíme $\mathbf{S} = \mathbf{Y}\mathbf{X}^T$ a provedeme singulární rozklad

$$\mathbf{S} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T.$$

Matice $\mathbf{S}$ má tvar

$$\mathbf{S} = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zz} & S_{zy} & S_{zz} \end{pmatrix},$$

kde $S_{xy} = \sum_{i=1}^{N} \mathbf{m}_{ix}' \mathbf{d}_{iy}'$, přičemž $\mathbf{m}_{ix}$ je x-ová souřadnice vektoru $\mathbf{m}_i$ a $\mathbf{d}_{iy}$ je y-ová souřadnice vektoru $\mathbf{d}_i$ a Po dosazení do členu, který maximalizujeme dostáváme

$$tr(\mathbf{R}\mathbf{Y}\mathbf{X}^T) = tr(\mathbf{R}\mathbf{S}) = tr(\mathbf{R}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T) = tr(\mathbf{\Sigma}\mathbf{V}^T\mathbf{R}\mathbf{U}).$$

Matice $\mathbf{V}$, $\mathbf{R}$ a $\mathbf{U}$ jsou ortogonální, takže i matice $\mathbf{Z} = \mathbf{V}^T\mathbf{R}\mathbf{U}$ je ortogonální. Sloupečky matice $\mathbf{Z}$ jsou tedy ortonormální vektory a platí $\mathbf{z}_j^T \mathbf{z}_j = 1$ pro každý sloupeček $\mathbf{z}_j$ matice $\mathbf{Z}$. Dále platí

$$1 = \mathbf{z}_j^T \mathbf{z}_j = \sum_{i=1}^{3} z_{ij}^2 \implies z_{ij}^2 \leq 1 \implies |z_{ij}| \leq 1.$$

Velikost všech prvků z_{ij} matice $\mathbf{Z}$ je tedy menší nebo rovna jedné. Protože $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \sigma_3)$, platí

$$tr(\mathbf{\Sigma}\mathbf{Z}) = \begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{pmatrix} \begin{pmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{pmatrix} = \sum_{i=1}^{3} \sigma_i z_{ii} \leq \sum_{i=1}^{3} \sigma_i.$$

Stopa je tedy největší, pokud $z_{ii} = 1$. Protože Z je ortonormální matice, musí být jednotková. Dále platí

$$\mathbf{I} = \mathbf{Z} = \mathbf{V}^T\mathbf{R}\mathbf{U} \implies \mathbf{V} = \mathbf{R}\mathbf{U} \implies \mathbf{R} = \mathbf{V}\mathbf{U}^T.$$

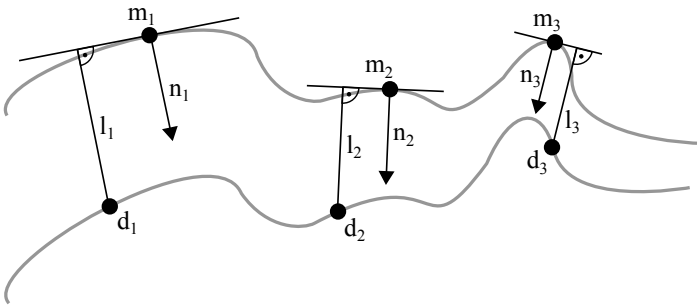
Optimální maticí rotace je tedy $\mathbf{R} = \mathbf{V}\mathbf{U}^T$ a optimální posunutí potom $\mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d$.

3.4.2 Vzdálenost bodu od roviny

Pokud kromě dvojic korespondujících bodů $(\mathbf{m}_i, \mathbf{d}_i)$ máme k dispozici také normály $\mathbf{n}_i$ v těchto bodech, můžeme optimální translaci $\mathbf{t}$ a rotaci $\mathbf{R}$ určit minimalizací chybové funkce

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} [(\mathbf{R}\mathbf{m}_i + \mathbf{t} - \mathbf{d}_i)\mathbf{n}_i]^2,$$

kde N je počet korespondujících dvojic bodů. Tento výraz minimalizuje součet vzdáleností l_i mezi body z mračna D a tečnými rovinami jejich korespondujících bodů z mračna M. Tento postup je podrobně popsán v [16].



Obrázek 3.4: Vzdálenost bodu od roviny

Matici $\mathbf{R}$ můžeme rozepsat jako $\mathbf{R} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z$, kde

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix},$$

$$\mathbf{R}_y = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix},$$

$$\mathbf{R}_z = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

a α, β, γ jsou úhly rotace podle os x, y, z. Protože předpokládáme, že úhly rotace v každé iteraci budou malé, můžeme položit $\cos\phi \approx 1$ a $\sin\phi \approx \phi$. Matici $\mathbf{R}$ potom můžeme vyjádřit jako

$$\mathbf{R} \approx \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix}.$$

Po dosazení do chybové funkce dostáváme

$$\begin{aligned} E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N} [&(\mathbf{m}_{ix} - \gamma\mathbf{m}_{iy} + \beta\mathbf{m}_{iz} + \mathbf{t}_x - \mathbf{d}_{ix})\mathbf{n}_{ix} + \\ &(\gamma\mathbf{m}_{ix} + \mathbf{m}_{iy} - \alpha\mathbf{m}_{iz} + \mathbf{t}_y - \mathbf{d}_{iy})\mathbf{n}_{iy} + \\ &(-\beta\mathbf{m}_{ix} + \alpha\mathbf{m}_{iy} + \mathbf{m}_{iz} + \mathbf{t}_z - \mathbf{d}_{iz})\mathbf{n}_{iz}]^2, \end{aligned}$$

což můžeme přepsat jako

$$\begin{aligned} E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N} [&(\mathbf{m}_i - \mathbf{d}_i)\mathbf{n}_i + \mathbf{t}\mathbf{n}_i + \\ &\alpha(\mathbf{m}_{iy}\mathbf{n}_{iz} - \mathbf{m}_{iz}\mathbf{n}_{iy}) + \cdot \\ &\beta(\mathbf{m}_{iz}\mathbf{n}_{ix} - \mathbf{m}_{ix}\mathbf{n}_{iz}) + \\ &\gamma(\mathbf{m}_{ix}\mathbf{n}_{iy} - \mathbf{m}_{iy}\mathbf{n}_{ix})]^2 \end{aligned}$$

Označíme

$$\mathbf{c} = \mathbf{m} \times \mathbf{n}$$

a

$$\mathbf{r} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$$

a chybová funkce tak dostává tvar

$$E(\mathbf{r},\mathbf{t}) = \sum_{i=1}^{N} [(\mathbf{m}_i - \mathbf{d}_i)\mathbf{n}_i + \mathbf{t}\mathbf{n}_i + \mathbf{r}\mathbf{c}_i]^2.$$

Tento výraz minimalizujeme tak, že položíme parciální derivace podle α, β, γ, $\mathbf{t}_x$, $\mathbf{t}_y$ a $\mathbf{t}_z$ rovny nule:

$$\frac{\partial E}{\partial \alpha} = \sum_{i=1}^{N} 2\mathbf{c}_{ix}[(\mathbf{m}_i - d_i)\mathbf{n}_i + \mathbf{t}\mathbf{n}_i + \mathbf{r}\mathbf{c}_i] = 0,$$

$$\frac{\partial E}{\partial \beta} = \sum_{i=1}^{N} 2\mathbf{c}_{iy}[(\mathbf{m}_i - d_i)\mathbf{n}_i + \mathbf{t}\mathbf{n}_i + \mathbf{r}\mathbf{c}_i] = 0,$$

$$\frac{\partial E}{\partial \gamma} = \sum_{i=1}^{N} 2\mathbf{c}_{iz}[(\mathbf{m}_i - d_i)\mathbf{n}_i + \mathbf{t}\mathbf{n}_i + \mathbf{r}\mathbf{c}_i] = 0,$$

$$\frac{\partial E}{\partial \mathbf{t}_x} = \sum_{i=1}^{N} 2\mathbf{n}_{ix}[(\mathbf{m}_i - d_i)\mathbf{n}_i + \mathbf{t}\mathbf{n}_i + \mathbf{r}\mathbf{c}_i] = 0,$$

$$\frac{\partial E}{\partial \mathbf{t}_y} = \sum_{i=1}^{N} 2\mathbf{n}_{iy}[(\mathbf{m}_i - d_i)\mathbf{n}_i + \mathbf{t}\mathbf{n}_i + \mathbf{r}\mathbf{c}_i] = 0,$$

$$\frac{\partial E}{\partial \mathbf{t}_z} = \sum_{i=1}^{N} 2\mathbf{n}_{iz}[(\mathbf{m}_i - d_i)\mathbf{n}_i + \mathbf{t}\mathbf{n}_i + \mathbf{r}\mathbf{c}_i] = 0.$$

Tato soustava rovnic se dá přepsat do maticového tvaru:

$$\sum_{i=1}^{N}\begin{pmatrix} \mathbf{c}_{i,x}\mathbf{c}_{i,x} & \mathbf{c}_{i,x}\mathbf{c}_{i,y} & \mathbf{c}_{i,x}\mathbf{c}_{i,z} & \mathbf{c}_{i,x}\mathbf{n}_{i,x} & \mathbf{c}_{i,x}\mathbf{n}_{i,y} & \mathbf{c}_{i,x}\mathbf{n}_{i,z} \\ \mathbf{c}_{i,y}\mathbf{c}_{i,x} & \mathbf{c}_{i,y}\mathbf{c}_{i,y} & \mathbf{c}_{i,y}\mathbf{c}_{i,z} & \mathbf{c}_{i,y}\mathbf{n}_{i,x} & \mathbf{c}_{i,y}\mathbf{n}_{i,y} & \mathbf{c}_{i,y}\mathbf{n}_{i,z} \\ \mathbf{c}_{i,z}\mathbf{c}_{i,x} & \mathbf{c}_{i,z}\mathbf{c}_{i,y} & \mathbf{c}_{i,z}\mathbf{c}_{i,z} & \mathbf{c}_{i,z}\mathbf{n}_{i,x} & \mathbf{c}_{i,z}\mathbf{n}_{i,y} & \mathbf{c}_{i,z}\mathbf{n}_{i,z} \\ \mathbf{n}_{i,x}\mathbf{c}_{i,x} & \mathbf{n}_{i,x}\mathbf{c}_{i,y} & \mathbf{n}_{i,x}\mathbf{c}_{i,z} & \mathbf{n}_{i,x}\mathbf{n}_{i,x} & \mathbf{n}_{i,x}\mathbf{n}_{i,y} & \mathbf{n}_{i,x}\mathbf{n}_{i,z} \\ \mathbf{n}_{i,y}\mathbf{c}_{i,x} & \mathbf{n}_{i,y}\mathbf{c}_{i,y} & \mathbf{n}_{i,y}\mathbf{c}_{i,z} & \mathbf{n}_{i,y}\mathbf{n}_{i,x} & \mathbf{n}_{i,y}\mathbf{n}_{i,y} & \mathbf{n}_{i,y}\mathbf{n}_{i,z} \\ \mathbf{n}_{i,z}\mathbf{c}_{i,x} & \mathbf{n}_{i,z}\mathbf{c}_{i,y} & \mathbf{n}_{i,z}\mathbf{c}_{i,z} & \mathbf{n}_{i,z}\mathbf{n}_{i,x} & \mathbf{n}_{i,z}\mathbf{n}_{i,y} & \mathbf{n}_{i,z}\mathbf{n}_{i,z} \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \mathbf{t}_x \\ \mathbf{t}_y \\ \mathbf{t}_z \end{pmatrix} = -\sum_{i=1}^{N}\begin{pmatrix} \mathbf{c}_{i,x}(\mathbf{m}_i - \mathbf{d}_i)\mathbf{n}_i \\ \mathbf{c}_{i,y}(\mathbf{m}_i - \mathbf{d}_i)\mathbf{n}_i \\ \mathbf{c}_{i,z}(\mathbf{m}_i - \mathbf{d}_i)\mathbf{n}_i \\ \mathbf{n}_{i,x}(\mathbf{m}_i - \mathbf{d}_i)\mathbf{n}_i \\ \mathbf{n}_{i,y}(\mathbf{m}_i - \mathbf{d}_i)\mathbf{n}_i \\ \mathbf{n}_{i,z}(\mathbf{m}_i - \mathbf{d}_i)\mathbf{n}_i \end{pmatrix}.$$

Obdrželi jsme rovnici ve tvaru $Cx = b$. Řešením této rovnice je optimální rotace a translace. Protože matice C je symetrická, můžme k nalezení řešení použít Choleského rozklad.

3.4.3 Singulární rozklad

Singulární rozklad [17] (někdy též nazývaný SVD rozklad) je jeden z nejdůžitějších nástrojů maticových výpočtů. Používá se v řadě aplikací numerické matematiky a hraje důležitou roli při použití metody nejmenších čtverců. V této práci je singulární rozklad použit na řešení právě tohoto problému.

Věta 3.1 *Každou matici* $\mathbf{H}$ *typu* (m, n) *lze vyjádřit ve tvaru*

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

kde $\mathbf{U}$ *je ortogonální matice řádu* m, $\mathbf{V}$ *je ortogonální matice řádu* n *a* $\mathbf{\Sigma}$ *je diagonální matice typu* (m, n),

$$\mathbf{\Sigma} \equiv diag(\sigma_1, \sigma_, \ldots, \sigma_p), \quad p = min(m, n)$$

a

$$\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq 0, \quad \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_p = 0,$$

kde $h(\mathbf{H}) = r$ *je hodnost matice* $\mathbf{H}$.

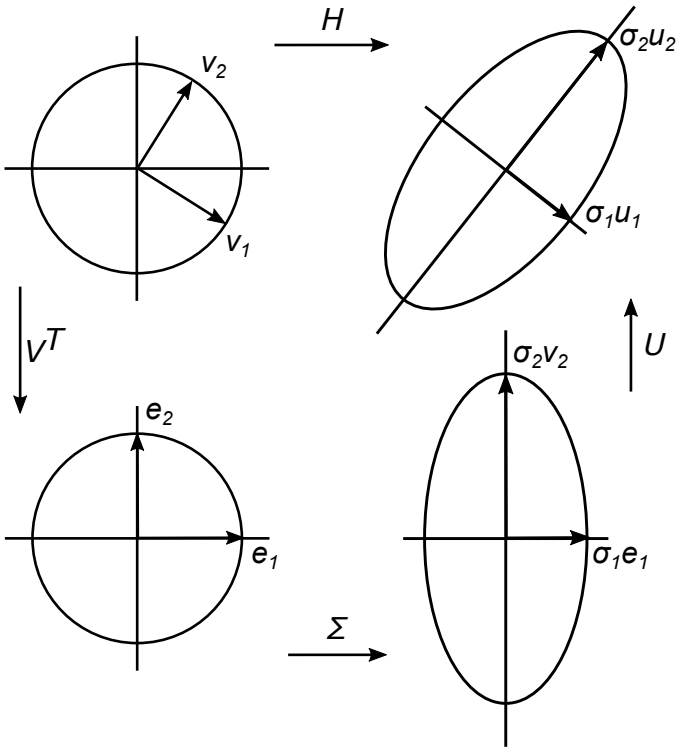
Singulární rozklad matice $\mathbf{A}$ lze vypočítat následovně [18]:

1. Nejdříve se vypočítá matice transponovaná $\mathbf{A}^T$ a $\mathbf{A}^T\mathbf{A}$.
2. Vypočítají se vlatní čísla matice $\mathbf{A}^T\mathbf{A}$ a seřadí sestupně podle absolutní hodnoty. Odmocněním těchto hodnot získáme singulární čísla matice $\mathbf{A}$.
3. Vytvoří se diagonální matice $\mathbf{S}$ s vlastními čísly matice $\mathbf{A}$ na diagonále v sestupném pořadí. Vypočte se její inverzní matice $\mathbf{S}^{-1}$.
4. Pomocí vlastních čísel z druhého kroku se vypočítají vlastní vektory matice $\mathbf{A}^T\mathbf{A}$. Tyto vektory vytvoří sloupce matice $\mathbf{V}$. Vypočítá se matice $\mathbf{V}^T$.
5. Matice $\mathbf{U}$ se vypočítá jako $\mathbf{U} = \mathbf{A}\mathbf{V}^T\mathbf{S}^{-1}$. Singulární rozklad poté získáme jako $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$

Singulární rozklad lze chápat jako posloupnost tři lineárních transformací:

1. Matice $\mathbf{V}$ představuje otočení nebo zrcadlení vektorů v m-rozměrném prostoru.
2. Matice $\mathbf{\Sigma}$ představuje zvětšení nebo zmenšení v jednotlivých směrech souřadného systému.
3. Matice $\mathbf{U}$ představuje otočení nebo zrcadlení vektorů v n-rozměrném prostoru.

Tato geometrická interpretace je zobrazena na obr. 3.5.



Obrázek 3.5: Geometrická interpretace singulárního rozkladu

3.4.4 Choleského rozklad

Choleského rozklad je rozklad symetrické pozitivně definitní matice na součin dolní a horní trojúhelníkové matice [19].

Věta 3.2 *Každou symetrickou pozitivně definitní matici* $\mathbf{H}$ *lze vyjádřit ve tvaru*

$$\mathbf{H} = \mathbf{L}\mathbf{L}^T,$$

kde $\mathbf{L}$ *je dolní trojúhelníková matice. Pro prvky matice* $\mathbf{L}$ *platí*

$$l_{kk} = \sqrt{h_{kk} - \sum_{j=1}^{k-1} l_{kj}^2},$$

$$l_{ik} = \frac{1}{l_{kk}}(h_{ik} - \sum_{j=1}^{k-1} l_{ij}l_{kj}), \qquad i = k+1, k+2, \ldots, n$$

pro $k = 1, 2, \ldots, n$*, kde* n *je řád matice* $\mathbf{H}$*.*

Choleského rozklad se používá pro řešení soustav rovnic ve tvaru $\mathbf{A}\mathbf{x} = \mathbf{b}$, kde $\mathbf{A}$ je pozitivně definitní matice. Tento problém lze vyřešit rozkladem $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ a následným vyřešením soustav rovnic $\mathbf{L}\mathbf{y} = \mathbf{b}$ přímým a $\mathbf{L}^T\mathbf{x} = \mathbf{y}$ zpětným chodem.

4 Implementace

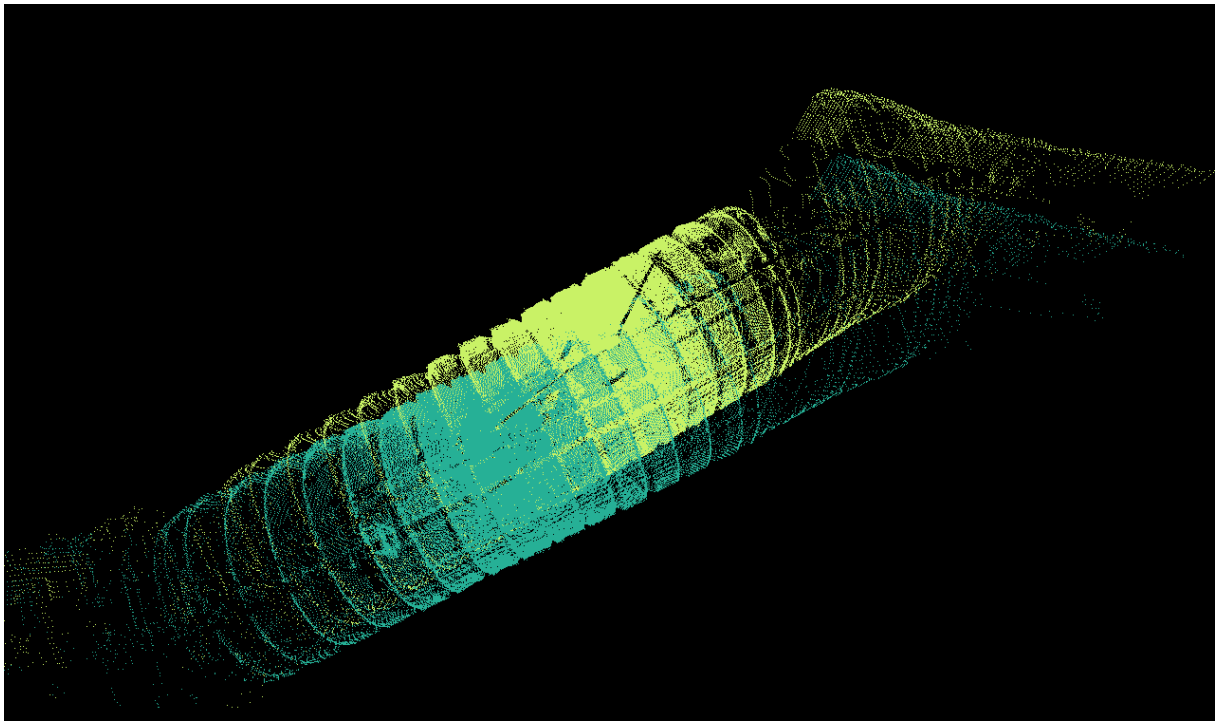
Hlavním cílem bakalářské práce bylo nastudovat algoritmus ICP a vytvořit funkční program, který by pomocí něj dokázal sesazovat mračna bodů. Jako programovací jazyk byl zvolen C#, jako vývojové studio bylo vybráno Visual Studio 2017. Pro vizualizaci mračen byla využita knihovna SharpGL.

Implementace představuje klasický ICP algoritmus. Za odpovídající si body jsou tedy považovány ty nejbližší. Hledání nejbližších bodů je urychleno použitím KD-stromu. Program dále minimalizuje součet čtverců vzdáleností mezi korespondujícími body pomocí SVD rozkladu.

4.1 Nastavení vstupních hodnot

Nezbytným vstupem programu jsou dvě mračna bodů. Tyto mračna mohou být ve formátu PCD, CSV nebo PLY. Mračna bodů na vstupu jsou zobrazeny na obr. 4.1

Program má dále několik volitelných parametrů. Uživatel může zvolit počateční rotaci a posun, pokud vzájemná poloha mračen není na počátku vyhovující. Tímto způsobem se dá zmenšit počet iterací nutný ke správnému sesazení a tím zrychlit chod programu. Protože algoritmus ICP nemusí vždy konvergovat ke správnému řešení, v některých připadech je změna počáteční polohy nutná. Dále si uživatel může zvolit, jak velká část vstupních mračen bude v algoritmu skutečně použita, nebo jestli budou použita mračna celá. Pokud jsou použity pouze části mračen, může dojít ke zhoršení konvergence algoritmu, tato volba však velmi zrychluje chod programu.



Obrázek 4.1: Mračna bodů na vstupu

4.2 Programové zpracování

Prvním krokem algoritmu je načtení mračen bodů. Pokud jsou zvolená mračna v jednom z podporovaných formátů, program je automaticky rozpozná a tyto mračna načte. Mezitím, co uživatel zadává další vstupní parametry jako je počáteční odhad polohy, program začíná vytvářet KD-strom pro první z mračen. Poté dochází k sérii iterací algoritmu ICP. V každé z nich je nejdříve vytvořeno mračno nejbližších bodu k druhému mračnu pomocí KD--stromu. Poté je toto mračno spolu s druhým mračnem posunuto do těžiště a je vytvořena kovariační matice těchto mračen. Optimální rotace a translace je potom nalezena pomocí singulárního rozkladu a tato transformace je aplikována. Algoritmus končí, pokud je rozdíl chyb v posledních dvou iteracích dostatečně malý.

Následující algoritmus shrnuje chod programu.

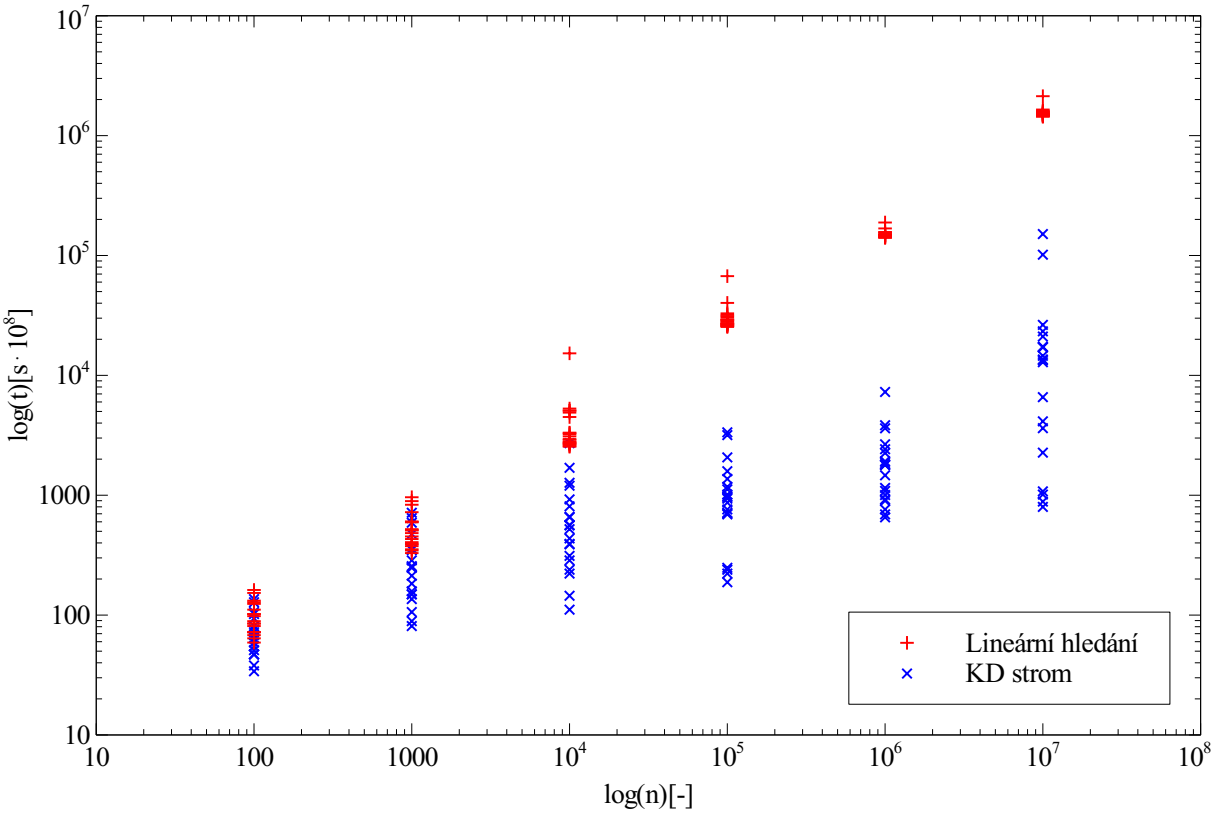
Algoritmus 2 Postup vytvořeného programu

1: **function** ICP($cloudA$, $cloudB$)
2: $treeA$:= KDTREE($cloudA$)
3: **while** ABS($error(k) - error(k-1)$)>ϵ **do**
4: $nearestNeighbours$=NNSEARCH($treeA$, $cloudB$)
5: p = CENTROID($nearestNeighbours$)
6: q = CENTROID($cloudB$)
7: $nearestNeighbours$=$nearestNeighbours$-p
8: $cloudB$=$cloud$-q
9: S = COVARIANCEMATRIX($nearestNeighbours$, $cloudB$)
10: U,V=SVD(S)
11: $R = VU^T$
12: $t = q - Rp$
13: TRANSFORM($cloudB$, R, t)
14: **end while**
15: **return** R, t
16: **end function**

4.3 KD-strom

Při tvorbě programu se jako významné vylepšení algoritmu ukázalo použití KD-stromu při hledání nejbližšího souseda. Oproti lineárnímu hledání výrazně urychlilo chod programu. Pro porovnání obou variant byly změřeny časy hledání v náhodně generovaných mračnech o velikostech $n = 100, 1000, 10000, 1000000, 1000000, 10000000$. Pro každou velikost bylo vygenerováno 20 mračen, ve kterých byly změřeny časy hledání nejbližšího bodu k dalšímu náhodně vygenerovanému bodu.

Výsledky můžeme vidět na obr. 4.2 a v tab. 4.1. Obr. 4.2 zobrazuje závislost času t na počtu bodů n v logaritmických souřadnicích. Tab. 4.1 potom ukazuje závislost průměrů časů $\bar{t}$ na počtu bodů n. Je vidět, že pro velmi malá mračna o velikosti 100 bodů jsou časy ještě srovnatelné. Už při velikosti v desítkách a stovkách tisíc, tedy u mračen s velikostí srovnatelnou s mračny použitými v této práci, je hledání za použití KD-stromu o jeden řád rychlejší. U největší množiny 10 milionu bodů, což je běžná velikost používaných mračen, jsou časy za požití KD-stromu v průměru o dva řády rychlejší.



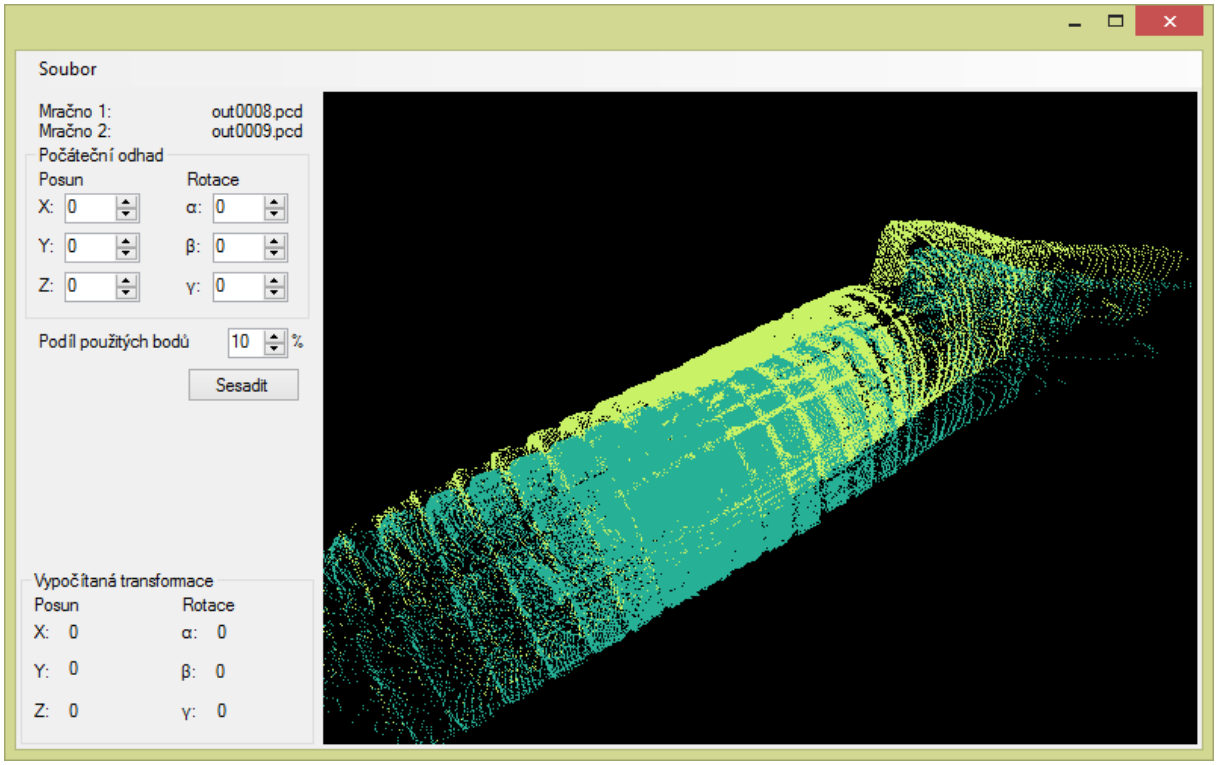
Obrázek 4.2: Porovná časů hledání nejbližšího souseda

	KD-strom	Lineární hledání
$n[-]$	$\bar{t}[s \cdot 10^8]$	$\bar{t}[s \cdot 10^8]$
100	69	97
1000	289	536
10000	705	4015
100000	1119	30755
1000000	1992	150208
10000000	22488	1552921

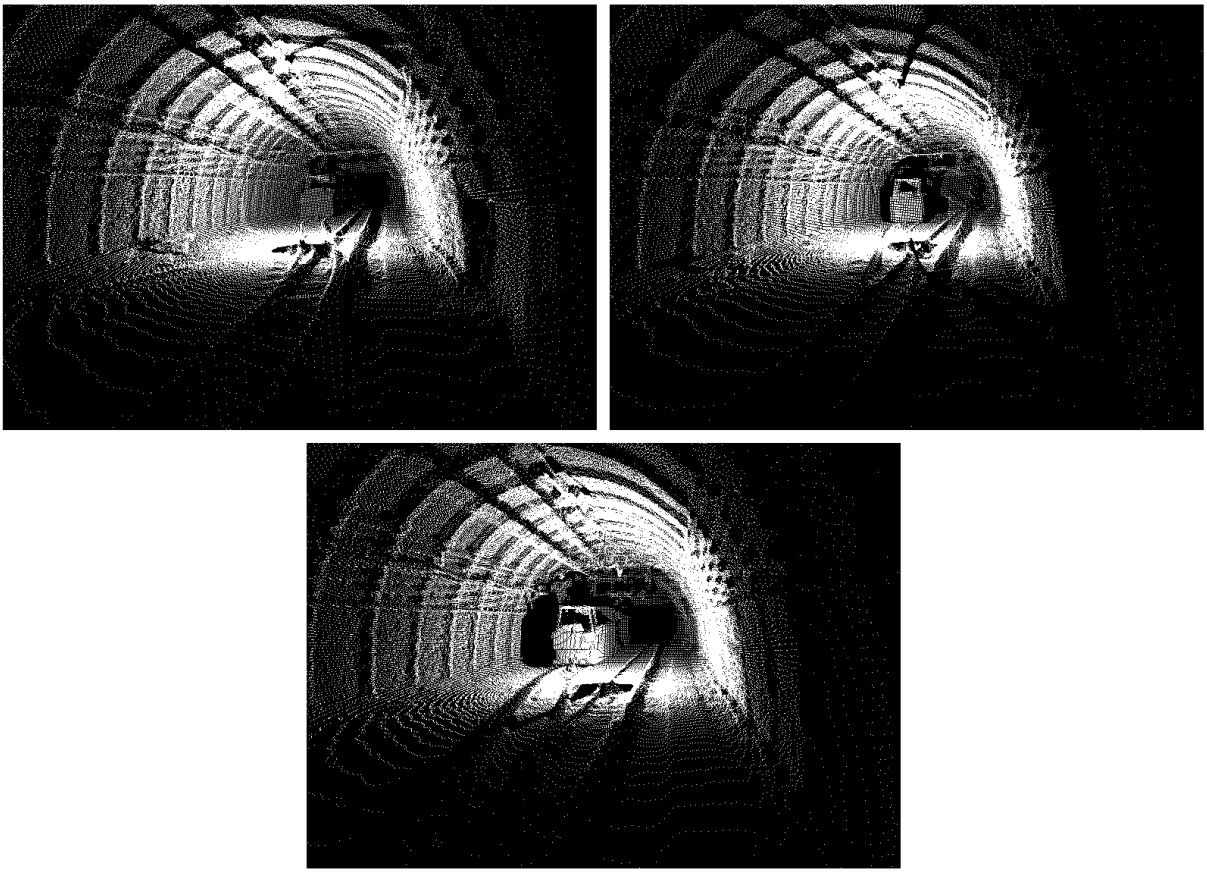
Tabulka 4.1: Průměry časů hledání nejbližšího souseda

4.4 Grafické rozhraní

Grafické rozhraní vytvořeného programu na sesazování mračna bodů pomocí algoritmu ICP má dvě hlavní části. Toto rozhraní je možné vidět na obr. 4.3. V levé části okna se nachází panel se vstupními mračny, které může uživatel zvolit v nabídce soubor v horní části okna. Dále může uživatel v tomto panelu zvolit počáteční translaci a rotaci. Hodnoty X, Y a Z představují posun ve směru souřadných os hodnoty α, β a γ představují rotaci podle jednotlivých souřadných os. Posledním parametrem je podíl použitých bodů zadávaný v procentech. Algoritmus uživatel spustí tlačitkem Sesadit. Vypočítaná transformace je následně vypsána ve spodní části panelu. Sesazené mračno má uživatel možnost uložit pod nabídkou Soubor. V pravé části okna se potom nachází náhled obou mračen. Tento náhled je možno otáčet levý tlačítkem myši, přibližovat a oddalovat kolečkem a posouvat pravým tlačítkem myši.



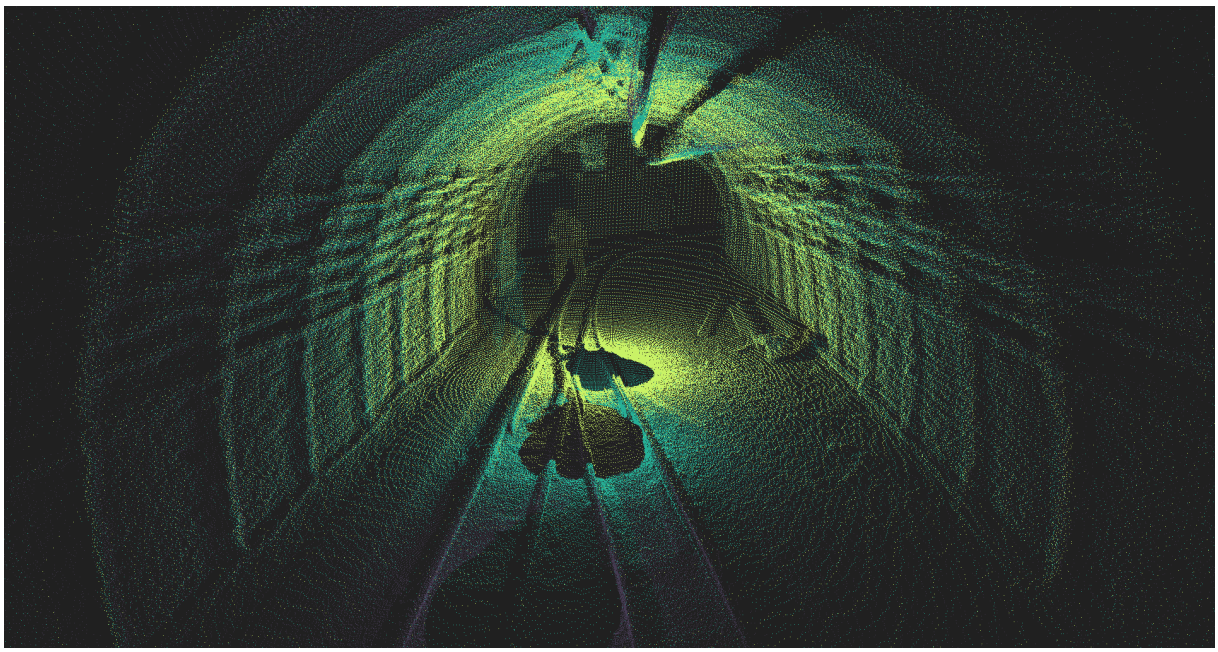
Obrázek 4.3: Rozhraní programu na sesazování mračen bodů



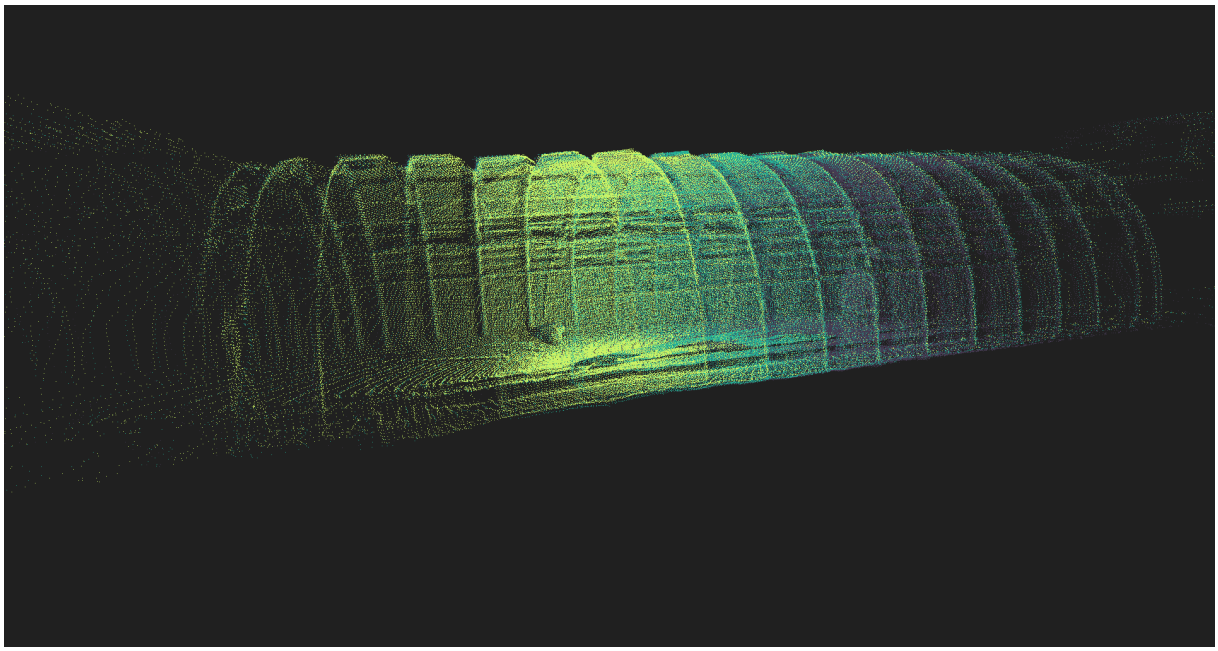
Obrázek 4.4: Testovací data

4.5 Výsledky

Pro účely testovaní vytvořeného programu byly poskytnuty reálná data ze 3D skeneru od Katedry robotiky z Fakulty strojní, VŠB-TU Ostrava. Tato data jsou zobrazena na obr. 4.4 Každé z poskytnutých mračen má velikost přibližně 200 000 bodů. Jednotlivá mračna tvoří po sesazení souvislou část důlní šachty, kterou je možno vidět na obr. 4.5 a 4.6.



Obrázek 4.5: Testovací data po sesazení



Obrázek 4.6: Testovací data po sesazení

5 Závěr

Cílem bakalářské práce bylo nastudovat problematiku týkající se mračna bodů a sesazování mračen bodů pomocí algoritmu ICP (Iterative Closest Point) a vytvořit funkční program, který by pomocí něj dokázal mračna bodů sesazovat. Tento algoritmus je iterační a obsahuje dva klíčové kroky, nalezení korespondujících párů bodů a minimalizaci vzdálenosti mezi těmito body. Práce se věnovala zejména zejména klasickému algoritmu ICP a tuto variantu využívá i vytvořený program. Pro urychlení chodu algoritmu byla využita datová struktura KD-strom. Dále bylo popsáno jedno z možných zlepšení algortimu v oblasti minimalizace chyby.

V části implementace je popsán program využívající algoritmus ICP vytvořený v rámci této bakalářské práce. Je zde otestován vliv použití KD-stromu na rychlost algoritmu. Funkčnost navrženého řešení je zde ověřena na reálných datech důlní šachty poskytnutých Katedrou robotiky z Fakulty strojní, VŠB-TU Ostrava.

Literatura

[1] CHEN, Y. a MEDIONI, G. Object modelling by registration of multiple range images. *Proc. IEEE Conf. on Robotics and Automation, 1991*, 1991, 2724-2729, ISBN 0-8168-2163-X.

[2] BESL, P. a MCKAY, N. A Method for Registration of 3-D shapes. *Trans. PAMI*, 1992, 14 (2), 239-256, ISSN 0162-8828.

[3] OLŠÁK, P. Lineární algebra [online], 2007 [cit. 2018-4-29]. Dostupné z: http://petr.olsak.net/ftp/olsak/linal/linal.pdf.

[4] BEČVÁŘ, J. *Lineární algebra*, Praha: MATFYZPRESS, 2005, ISBN 80-86732-57-6.

[5] GILBERT, G. Positive definitive matrices and Sylvester's criterion. *The American Mathematical Monthly*, 1991, 98 (1), 44-46, ISSN 0002-9890.

[6] DORAI, C., WENG, J. a JAIN, A. Optimal Registration of Object Views Using Range Data. *Trans. PAMI*, 1997, 19 (10), 1131-1138, ISSN 0162-8828.

[7] CRACKNELL, A. a HAYES, L. *Introduction to Remote Sensing (2 ed.)*, Londýn: Taylor and Francis, 2007, ISBN 0-8493-9255-1.

[8] TURK, G. a LEVOY, M. Zippered Polygon Meshes from Range Images. *Proc. SIGGRAPH, 1994*, 1994, 311-318, ISBN 0-89791-667-0.

[9] MASUDA, T., SAKAUE, K. a YOKOYA, N. Registration and integration of multiple range images for 3-D model construction. *Proc. CVPR, 1996*, 1996, 879-883, ISBN 0-8186-7282-X.

[10] SIMON, D. Fast and Accurate Shape-Based Registration. Pittsburgh: Carnegie Mellon University, 1996.

[11] PULLI, K. Multiview registration for large data sets. *Proc. 3DIM, 1999*, 1999, 160-168, ISBN 0-7695-0062-5.

[12] ARUN, K. HUANG T. a BLOSTEIN, S. Least-Square Fitting of Two 3-D Point Sets. *Trans. PAMI*, 1987, 9 (5), 698-700, ISSN 0162-8828.

[13] HORN, B. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 1987, 4 (4), 629-642, ISSN 1084-7529.

[14] KUBÍK, P. Semiadaptivní 3D modeling. Praha: Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, 2006.

[15] SORKINE-HORNUNG, O. a RABINOVICH, M. Least-Squares Rigid Motion Using SVD [online], 2017 [cit. 2018-4-15]. Dostupné z: https://igl.ethz.ch/projects/ARAP/svd_rot.pdf.

[16] RUSINKIEWICZ, S. Derivation of Point-to-Plane Minimization [online], 2003 [cit. 2018-4-1]. Dostupné z: https://www.cs.princeton.edu/ smr/papers/icpstability.pdf.

[17] JIA, Y. Singular Value Decomposition [online], 2017 [cit. 2018-4-29]. Dostupné z: http://web.cs.iastate.edu/ cs577/handouts/svd.pdf.

[18] GARCIA E. Singular Value Decomposition A Fast Track Tutorial [online], 2016 [cit. 2018-5-6]. Dostupné z: http://www.minerazzi.com/tutorials/singular-value-decomposition-fast-track-tutorial.pdf.

[19] ČERMÁK, L. a HLAVIČKA R. *Numerické metody*, Brno: Akademické nakladatelství CERM, 2016, ISBN 978-80-214-5437-8.

6 Seznam použitých zkratek a symbolů

$\mathbb{R}$	množina reálných čísel
$\mathbb{R}^n$	n-rozměrný vektorový prostor reálných čísel
$\mathbb{C}$	množina komplexních čísel
$\mathbf{a} \times \mathbf{b}$	vektorový součin vektorů $\mathbf{a}$ a $\mathbf{b}$
$\mathbf{A}^T$	matice transponovaná k matici $\mathbf{A}$
$\mathbf{A}^{-1}$	matice inverzní k matici $\mathbf{A}$
$tr(\mathbf{A})$	stopa matice $\mathbf{A}$
$h(\mathbf{A})$	hodnost matice $\mathbf{A}$

7 Seznam příloh

Přiložené CD obahuje tři adresáře:

1. **data** Složka s testovacími daty
2. **icp** Složka obsahující spustitelný exe soubor
3. **kod** Zdrojový kód programu