

Univerzita Karlova

Přírodovědecká fakulta

Katedra aplikované geoinformatiky a kartografie

Studijní program: Geografie (bakalářské studium)

Studijní obor: Fyzická geografie a geoinformatika



Tomáš Hřebec

**3D datová reprezentace a vizualizace komplexních prvků
silniční infrastruktury**

**3D data representation and visualization of complex features of
traffic infrastructure**

Bakalářská práce

Vedoucí bakalářské práce: Mgr. Lukáš Brůha, Ph.D.

Praha 2021

Vysoká škola: Univerzita Karlova v Praze

Fakulta: Přírodovědecká

Katedra: Aplikované geoinformatiky a kartografie

Školní rok: 2020/2021

Zadání bakalářské práce

pro Tomáše Hřebce

obor Fyzická geografie a geoinformatika

Název tématu: 3D datová reprezentace a vizualizace komplexních prvků silniční infrastruktury

Zásady pro vypracování

Práce navrhne a implementuje postup 3D vizualizace geometrie silniční sítě odvozené z dat leteckého snímkování. Navržené řešení se zaměří na komplexní situace typu mimoúrovňového křížení a dálničních křižovatek. Řešení práce bude uvažovat proměnlivé parametry silniční sítě či mostních konstrukcí a poskytne uživateli nástroj pro efektivní generování 3D modelů v závislosti na těchto parametrech.

Rozsah grafických prací: dle potřeby

Rozsah průvodní zprávy: cca 40 - 60 stran

Seznam odborné literatury:

KELLY, G., MCCABE, H. (2006): A survey of procedural techniques for city generation. ITB Journal, 14: 87-130.

ULLMAN, S. (1979): The interpretation of structure from motion, Proceedings of the Royal Society of London. Series B, Biological Sciences, 203, 405-426

WONKA, P., WIMMER, M., SILLION, RIBARSKY, W. (2003): Instant architecture. ACM Transactions on Graphics 22(3), 669-677.

Vedoucí diplomové práce: Lukáš Brůha, Ph.D.

Konzultant diplomové práce:

Datum zadání diplomové práce: 01.12.2020

Termín odevzdání diplomové práce: podzim 2021

Platnost tohoto zadání je po dobu jednoho akademického roku.

.....
Vedoucí diplomové práce

.....
Vedoucí katedry

V Praze dne: 1.12.2020

Prohlášení

Prohlašuji, že jsem závěrečnou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

V Praze, 26. 07. 2021

.....

podpis studenta

Poděkování

Chtěl bych poděkovat Mgr. Lukáši Brůhovi, Ph.D za vedení, za pomoc a rady při zpracování této bakalářské práce. Dále bych chtěl poděkovat Ing. MVC. Heleně Hřebcové za pomoc při pořizování dat v terénu, a i zbytku rodiny za podporu při studiu.

Abstrakt

Tato bakalářská práce navrhuje a implementuje postup tvorby a vizualizace 3D modelů komplexních prvků silniční infrastruktury se zaměřením na mimoúrovňové křížení. Teoretická část nastiňuje principy modelování, způsoby sběru prostorových dat pomocí metody Structure from Motion a pomocí leteckých snímků. Také představuje základy softwaru Blender, ve kterém je model tvořen, a jak pracuje software ArcGIS Pro, ve kterém jsou výsledné modely následně vizualizovány, s 3D daty.

Tvorba modelu mostu je vyřešena pomocí doplňku pro program Blender. V této práci je popsáno, jak byl doplněk vytvořen, jak funguje a jak jej ovládat. Nakonec je zde ukázáno, jak modely vizualizovat.

Klíčová slova: Prostorová data, datové struktury, 3D GIS, 3D vizualizace, dopravní infrastruktura

Abstract

This bachelor thesis designs and implements the process of creating and visualizing of 3D models of complex elements of road infrastructure with focus on level crossing. The theoretical part outlines the principles of modeling, methods of collecting spatial data using the Structure from Motion method and aerial images. This thesis also introduces the basics of Blender software, which the model is created in, and how ArcGIS Pro software, in which the final models are visualized, works with 3D data-

Model of a bridge is created by an add-on for Blender. This work describes how the add-on was created, how it works and how to control it. In the end, there is described how to visualize the models.

Keywords: Spatial data structures, 3D GIS, 3D visualization, traffic infrastructure

Obsah

Abstrakt.....	6
Abstract	7
Obsah.....	8
Seznam obrázků	10
1 Úvod.....	12
2 Procedurální modelování.....	13
2.1 Gramatika a techniky procedurálního modelování.....	13
2.1.1 L-systém.....	14
2.1.2 Shape Grammars	15
2.1.3 Split Grammars.....	16
3 Structure from Motion.....	18
3.1 Procedura SfM.....	18
3.2 Využití	19
4 Detekce geometrie silnic z leteckých snímků	21
4.1 Detekce silničních pruhů	21
4.1.1 Klasifikace obrazu.....	22
4.1.2 Úprava rastru	22
4.1.3 Úprava vektorů.....	23
4.1.4 Výstupní data.....	24
5 Software.....	26
5.1 Software Blender	26
5.1.1 Python v Blenderu.....	26
5.1.2 Datová struktura v Blenderu.....	27

5.2	Software ArcGIS Pro.....	28
5.2.1	Python v ArcGIS Pro.....	28
5.2.2	3D data v ArcGIS Pro.....	28
6	Modelování mostů.....	31
6.1	Model vytvořený pomocí SfM	31
6.1.1	Lokalizace mostu.....	31
6.1.2	Snímkování.....	31
6.1.3	Tvorba modelu a výsledné rozměry mostu.....	32
6.2	Generátor mostu	33
6.2.1	Metoda generování	34
6.2.2	Tvorba půdorysu.....	34
6.2.3	Tvorba nosných sloupů.....	35
6.2.4	Tvorba nadjezdu	35
6.2.5	Silnice a jízdní pruhy.....	36
6.2.6	Svodidla.....	36
6.2.7	Detailní prvky	36
6.2.8	Textury	37
6.2.9	Uživatelské rozhraní a ovládání	39
6.2.10	Instalace.....	41
6.3	Vizualizace	42
6.3.1	Tvorba multipatch souborů.....	42
6.3.2	Tvorba surface TINu	43
7	Výsledky	45
8	Diskuse.....	49
9	Závěr	51
	Seznam použitých zdrojů.....	52

Seznam obrázků

Obrázek 1 Model získaný L-systémem	15
Obrázek 2 Modelování budovy pomocí Shape Grammars.....	16
Obrázek 3 Systém gramatiky dělení.....	17
Obrázek 4 modelování stěny pomocí Split Grammars	17
Obrázek 5 Kroky procedury SfM	18
Obrázek 6 Výpočet souřadnic bodů pomocí triangulace.....	19
Obrázek 7 Zužování rastru	22
Obrázek 8 Ukázka skeletonizace polygonu.....	23
Obrázek 9 Manuálně doplněné chybějící části silničních pruhů (červeně).....	23
Obrázek 10 Ukázka dat RSM.shp.....	24
Obrázek 11 Ukázka dat drivinglines.shp.....	25
Obrázek 12 Příklad vytvořené plochy pomocí vrcholů.....	28
Obrázek 13 Způsoby uchování geometrické informace v multipatch souborech.....	30
Obrázek 14 Mapa s modelovaným mostem Zdroj: Mapy.cz.....	32
Obrázek 15 Výsledný model z bodového mračna	33
Obrázek 16 Náčrt půdorysu mostu	35
Obrázek 17 Ukázka zdroje textury pro hranu nadjezdu	37
Obrázek 18 Upravená textura hrany nadjezdu pomocí Bake	38
Obrázek 19 Textura nosného sloupu	38
Obrázek 20 Upravená textura nosného sloupu pomocí Bake.....	39

Obrázek 21 Ovládací panel pro rozšíření Generátor mostů	41
Obrázek 22 Model vygenerovaný v Blenderu	45
Obrázek 23 Vizualizace ve Scéně ArcGIS Pro	46
Obrázek 24 Vizualizace podjezdu	46
Obrázek 25 Vizualizace 2. mostu	47
Obrázek 26 Konstantní výška surface TINu.....	48
Obrázek 27 Surface TIN pracující se 2 modely	48

1 Úvod

S rozmachem informačních technologií v posledních letech přišel i velký rozmach technologií geoinformačních. S ním ovšem přišla i větší poptávka po prostorových datech v různých formách. Jednou z těchto forem mohou být například 3D data.

Cílem této práce je navrhnout a vytvořit postup tvorby a vizualizace 3D modelů komplexních prvků silniční infrastruktury se zaměřením na mimoúrovňová křížení. Tento způsob tvorby by měl být automatický pro obecně jakýkoli most. Toho je dosaženo pomocí skriptu, který pracuje jako rozšíření pro program Blender. Data použitá pro tvorbu mostu jsou z různých zdrojů. Prvním zdrojem dat jsou data z práce Jakuba Růžičky (2020), která nesou informace o geometrii jízdních pruhů a jsou získány z leteckých snímků. Druhým zdrojem dat je metoda Structure from motion, která dokáže vytvořit 3D model za pomoci snímků objektu pořízených ze všech jeho stran.

Tato práce je strukturována tak, že v první části rešerše je pozornost upřena na principy procedurálního modelování, na jejichž základech je poté tvořeno rozšíření pro Blender. V druhé části rešerše se práce zaměřuje na metody pořizování dat a v poslední, třetí části na software Blender a ArcGIS Pro. Je zde popis funkce softwarů a toho, s jakými daty pracují. Tato kapitola je důležitá k pochopení, proč jsou později prováděny některé procesy, jako je například převod dat do formátu multipatch.

V metodice je popsána tvorba rozšíření, jeho funkce a ovládání. Ve zkratce je zde i návod pro instalaci. Dále je zde popsán skript na přípravu dat pro vizualizaci a případnou práci s nimi v programu ArcGIS Pro.

V závěru práce je pak poukázáno na to, že obecné generování sice přináší urychlení tvorby, ale zároveň zde dochází ke ztrátě věrohodnosti a přesnosti.

2 Procedurální modelování

Procedurální modelování je jedna z metod vytváření 3D modelů. Tato metoda je v poslední době využívána zdaleka nejvíce. Využívá se k modelování nejrůznějších objektů, jako jsou rostliny, budovy, silniční sítě, ale třeba i textur. Procedurální modelování však nemá jednu jednoduchou definici. Zahrnuje širokou škálu poloautomatických a automatických generativních technik, které vytvářejí výsledný model na základě několika vstupních parametrů.

Mezi hlavní výhody procedurálního modelování patří, že k vygenerování velkého množství rozdílných modelů nám stačí jen několik málo rozdílných parametrů a pár generativních pravidel. Tato metoda dokonce umožňuje generovat rozdílné modely při použití totožných vstupních parametrů, protože některé pravidla fungují na principu náhodných čísel. Tato vlastnost může ušetřit velké množství času. I přes tyto výhody v některých případech nelze použít pro vytvoření 3D modelu procedurální modelování a musí být využito jiných technik, například ručního modelování. Toto je způsobeno malou kontrolou uživatele nad výsledným modelem (Smelik a kol. 2014).

2.1 Gramatika a techniky procedurálního modelování

Procedurální modelování a jeho jednotlivé techniky jsou založeny na gramatice. Gramatiku si můžeme představit jako čtveřici (N, T, P, S) , kde N je množina neterminálních symbolů, které mohou být nahrazeny. T je množina terminálních symbolů, které již nahrazeny být nemohou. P je množina pravidel pro nahrazování symbolů v množině N a S je počáteční symbol gramatiky. Gramatika je tedy soubor pravidel pro nahrazování symbolů, které je ukončeno tím, že jsou všechny symboly již nenahraditelné (terminální) nebo již proběhl stanovený počet opakování (Melichar 2003).

Příkladem může být gramatika $G = (N = \{a, b\}, T = \{\}, P = \{a \rightarrow ab, b \rightarrow bb\}, S = \{a\})$. Pak výsledný řetězec po daném počtu iterací n vypadá takto:

$n = 0$: a

$n = 1$: ab

$n = 2$: abbb

n = 3: abbbbbbb

Následující techniky procedurálního modelování jsou založeny právě na tomto principu generování řetězce symbolů. Jednotlivé symboly mohou mít různé významy a jejich postupným nahrazováním je určován i celkový význam vzniklého řetězce.

2.1.1 L-systém

Lindenmayerův systém (zkráceně L-systém) byl představen v roce 1968 k popsání růstu živých organismů. Od základní gramatiky se L-systém liší tím, že běžná gramatika nahrazuje symboly postupně, ale Lindenmayer pracuje s nahrazováním symbolů v řetězci současně (Prusinkiewicz a Lindenmayer 1990).

Nejjednodušší definice L-systému vychází podle Prusinkiewiczeva (1986) z želví grafiky. Jedná se o počítačovou grafiku, ve které se pohybuje kurzor (želva) a jeho stav je definovaný trojicí (x, y, α) , kde x a y jsou souřadnice kurzoru a α je úhel, ve kterém kurzor směřuje. Kurzor se poté řídí příkazy, které mu jsou předány programem, a tím je zajištěn jeho pohyb a vytváření samotné grafiky. Příkazy jsou reprezentovány následujícími symboly:

F: pohyb vpřed o vzdálenost d a přesunutí želvy z bodu (x, y) do (x', y') . Mezi body je nakreslena linie.

f: pohyb vpřed o vzdálenost d a přesunutí želvy z bodu (x, y) do (x', y') bez kreslení linie mezi body.

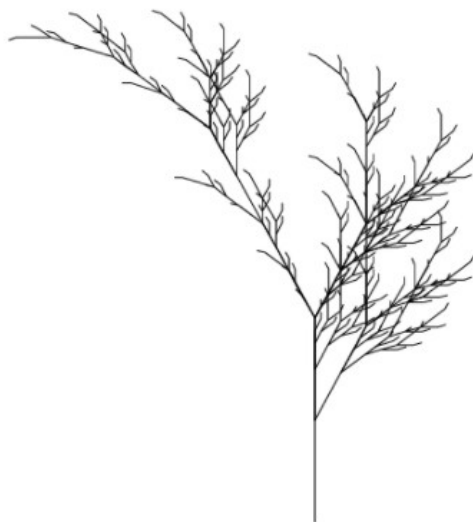
+: otočení želvy vpravo o úhel δ .

-: otočení želvy vlevo o úhel δ .

[: uložení pozice želvy

]: načtení pozice želvy

Když máme takto definovaný L-systém, tak můžeme s jeho pomocí vygenerovat ukázkový příklad. $L = (N, T, P, S)$, kde $N = \{X, F\}$, $T = \{+, -, [,]\}$, $P = \{X \rightarrow F - [[X] + X] + F[+ FX] - X, F \rightarrow FF\}$, $S = \{X\}$. Při vstupu jsou nastaveny parametry $n = 5$ (počet iterací) a $\delta = 22.5^\circ$ (Kelly a McCabe 2006). Výsledný model připomíná rostlinu (obr. 1).



Obrázek 1 Model získaný L-systémem

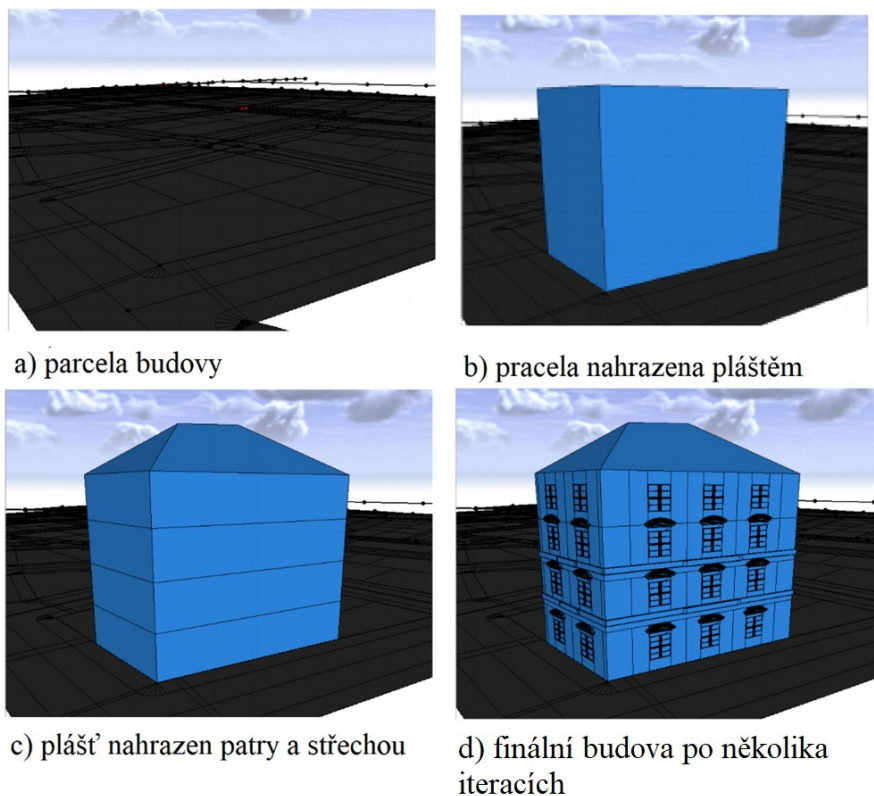
Zdroj: (Kelly a McCabe 2006)

L-System byl vytvořen pro generování modelů rostlin a v dnešní době se používá i ke generování modelů silniční sítě.

2.1.2 Shape Grammars

Shape Grammars, do češtiny přeloženo jako gramatika tvarů, vychází z podobného principu jako L-systém. Je také definována pomocí symbolů, ale ty zde nepředstavují pohyby a operace kurzoru. Tyto symboly reprezentují geometrické objekty, které jsou poté nahrazovány jinými objekty, a to následně vytváří větší objekt tvořený z jednotlivých částí (Stiny 1975).

Díky této podobnosti s L-systémem je práce s gramatikou tvarů obdobná. Definujeme gramatiku a počet iterací n . Jako výsledek je vrácen řetězec symbolů reprezentující finální objekt. Jako příklad nám může posloužit tato modelová situace: počáteční symbol je parcela definující půdorys budovy. Při první iteraci je tato parcela nahrazena pláštěm budovy. V druhé iteraci je plášť nahrazen jednotlivými patry a střechou. Je tedy nahrazen za více tvarů, ale stále jen v rámci svého rozměru. V dalších iteracích se nahrazují pláště jednotlivých pater zdmi, okny a dalšími detailnějšími prvky. Postupnou výměnu tvarů lze pozorovat na obrázku č. 2 (Frantál 2018).

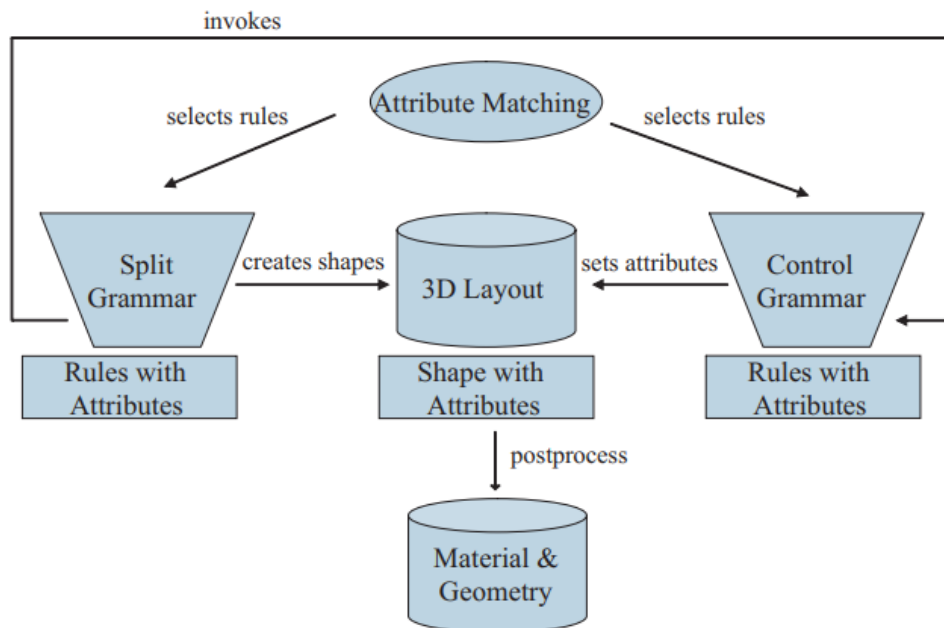


Obrázek 2 Modelování budovy pomocí Shape Grammars
Zdroj: (Frantál 2018)

2.1.3 Split Grammars

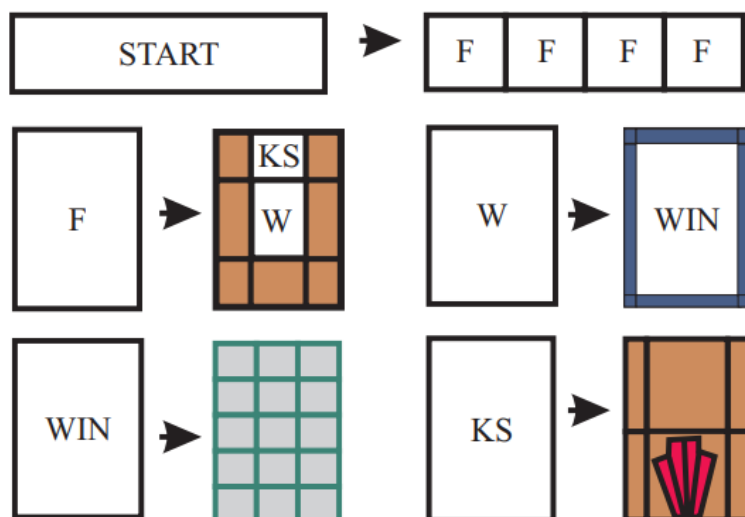
Split Grammars, do češtiny přeloženo jako gramatika dělení, je jednou z dalších gramatik sloužící ke generování převážně budov. S touto metodou přišli Peter Wonka a kol. (2003). Tato technika trochu odbočuje z klasických řetězcových gramatik. Zakládá se na principu takzvaných základních tvarů (Basic shapes). Základní tvary jsou zjednodušeně řečeno stavební bloky této gramatiky. Jedná se o kvádry, válce nebo hranoly.

Gramatika dělení má tedy za úkol rozdělit výsledný model na již nedělitelné části. To si lze představit tak, že budovu rozdělí na jednotlivá patra, ta poté na stěny, které dále dělí na okna, dveře a další části. Nakonec se právě tyto tvary rozdělí až na základní tvary. Další součástí systému Shape Grammar je tzv. Control Grammar. Ta má za úkol kontrolovat, a i popřípadě upravovat, atributy vzniklé při dělení jednotlivých částí objektu. Tyto atributy se totiž přiřazují z původních do nově vzniklých. Poslední část systému se nazývá Attribute Matching. Tato část zajišťuje výběr pravidel pro dělení částí objektu tak, aby co nejlépe odpovídaly jejich atributům (Wonka a kol. 2003). Schéma celé gramatiky je znázorněno na obrázku č. 3.



Obrázek 3 Systém gramatiky dělení
Zdroj: (Wonka a kol. 2003)

Konkrétní příklad využití gramatiky dělení při generování budovy lze vidět na obrázku č. 4. Zde vidíme základní tvar (Start), který znázorňuje stěnu. Ten je následně rozdělen na menší základní tvary (F). Tato část je rozdělena na již nedělitelné tvary (znázorněny hnědě) a okno (W). To je rozděleno na sklo (WIN) a rám okna (modrý rámeček). Sklo je nakonec rozděleno na již nedělitelné čtverce. V konečném kroku je nahrazen KS za ozdobný prvek fasády.



Obrázek 4 modelování stěny pomocí Split Grammars
Zdroj: (Wonka a kol. 2003)

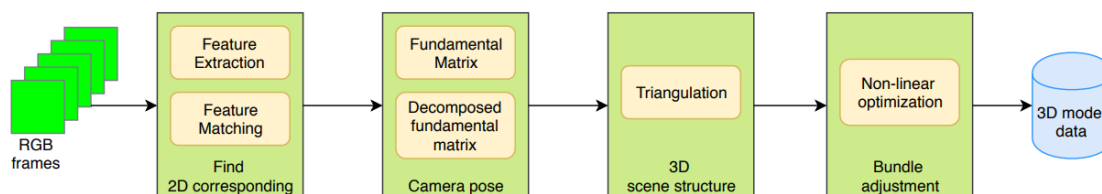
3 Structure from Motion

Structure from Motion (zkráceně SfM) je fotogrammetrická metoda, která umí vytvořit 3D model z fotografií, pořízených „obyčejným“ digitálním fotoaparátem. Největší problém nastává při pořízení fotografií. Fotografie promítnou 3D objekt na 2D plochu a při tom dojde ke ztrátě rozměru hloubky. Metoda SfM je jedna z nejznámějších metod v oboru počítačového vidění (computer vision) a má za úkol obnovit rozměr hloubky a umožnit vytvoření 3D modelu.

Základní princip SfM vychází z výpočtů paralax mezi jednotlivými překrývajícími se mi snímky. Na podobném principu pracuje i lidský zrak a umožňuje nám vnímat svět kolem nás ve 3D. Tato technika byla poprvé představena v roce 1979 (Ullman 1979).

3.1 Procedura SfM

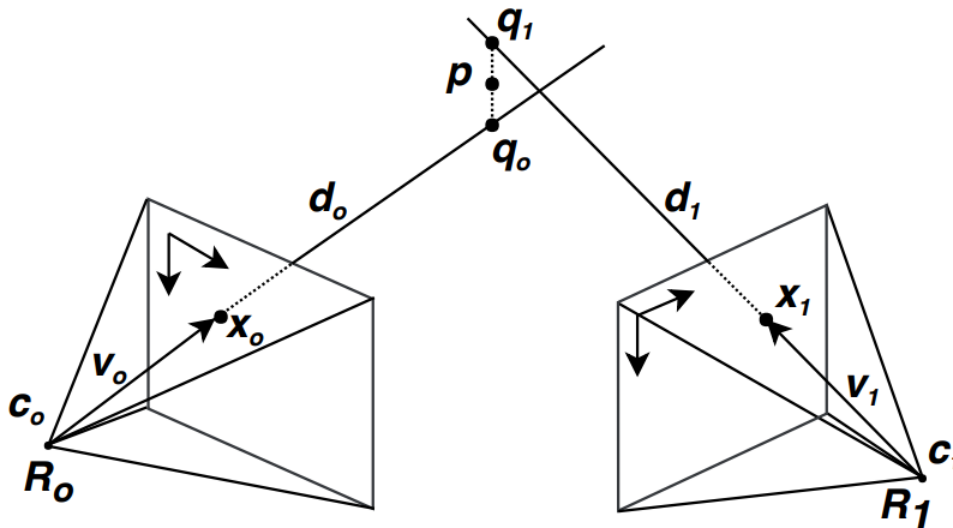
Jak již bylo zmíněno, k využití metody SfM stačí mít digitální fotoaparát. Konkrétní aplikace proto může být pokaždé trochu jiná. Budou se lišit kvalita snímků, potřebná vzdálenost k jejich pořízení a jejich počet. I přes tyto rozdíly zůstává základní procedura stále stejná (obr. 5).



Obrázek 5 Kroky procedury SfM
Zdroj: (Huynh 2018)

Prvním krokem je pořízení snímků podle základních pravidel. Snímky by měly být všechny ve stejné vzdálenosti od objektu a měly by mít co největší překryv. Další krok je nalezení bodů navzájem si odpovídajících mezi dvojicemi snímků. Tento proces obvykle začíná určením bodů zájmu na jednotlivých snímcích. Může jít o rohy budov či jiné výrazné prvky. Poté pomocí metody Scale Invariant Feature Transform (SIFT) jsou k sobě automaticky přiřazeny body s podobnými vlastnostmi.

Třetím krokem je získání relativních pozic kamer, tedy vzájemných pozic, ze kterých bylo snímáno. Toho je dosaženo transformačními maticemi. Po získání relativních souřadnic jednotlivých snímků lze pomocí triangulace získat i souřadnice jednotlivých bodů (obr. 6).



Obrázek 6 Výpočet souřadnic bodů pomocí triangulace
Zdroj: (Huynh 2018)

R_0 a R_1 představují umístění fotoaparátu, q_0 a q_1 jsou body na snímku a p představuje výsledný bod ve 3D získaný pomocí triangulace.

Posledním krokem před vznikem samotného 3D modelu je úprava a kontrola, zda jsou vypočtené body správné. Tento krok se nazývá Bundel adjustment. Tento krok je dělán automaticky v softwaru (Huynh 2018).

3.2 Využití

Tato metoda má v dnešní době velké uplatnění, protože se jedná o relativně levnou a rychlou metodu pro získání 3D modelu reálného objektu. Za první využití v praxi je považováno vygenerování 3D modelu památek z fotek turistů. Snavely a kol. (2006) To nám odhaluje tedy jednu možnost využití, získávání geometrie budov jakéhokoli typu.

Ve studii od HU a kol. (2020) se snaží autoři nalézt algoritmus na zpracování bodového mračna, které bylo získáno pomocí metody SfM a jehož geometrie reprezentuje lanový silniční most. Takto získaná data chtějí využít například při kontrole statiky mostu.

Westoby a kol. (2012) ve své studii porovnávají metodu SfM s jinými metodami sběru prostorových dat. Převážně se zaměřují na laserové skenování (LiDAR). Při porovnání těchto dvou metod dochází k závěru, že LiDAR je sice přesnější, ale potřebujeme k němu daleko dražší přístroj, zatímco k metodě SfM nám stačí jen fotoaparát. Dále laserové skenování většinou trvá déle, ale to záleží na parametrech přístroje. Nevýhodou SfM je naopak to, že narozdíl od laseru neumí projít vegetací.

4 Detekce geometrie silnic z leteckých snímků

Většina prací se zabývá detekcí silnic z leteckých snímků jako celku bez ohledu na počet jízdních pruhů. Na získání jednotlivých jízdních pruhů byl v poslední době kladen výrazně větší důraz kvůli autonomnímu řízení a jiným asistenčním systémům v automobilové dopravě. Tyto výzkumy se nezaměřují na získání silničního značení z leteckých snímků, ale spíše na získávání jejich geometrie z perspektivy automobilu a řidiče. Kvůli tomu není možné tyto metody použít při detekci jízdních pruhů z leteckých snímků, protože jsou pořízovány v jiné perspektivě atd. (Růžička 2020).

Mezi hlavní články zabývajícími se detekcí silnic jako celku patří například práce od Hormese a Seravanana (2016). V této práci se autoři navrhuje metodu založenou na třech plně automatických krocích. První je segmentace obrazu, při které se rozdělí družicové snímky na oblasti s podobnými vlastnostmi. Dalším krokem je poté rozhodování na základě kontinuity a posledním krokem je vektorizace segmentů, které byly rozpoznány jako komunikace. Další z možných přístupů ukázali Shahi a kol. (2015), kteří si pro detekci silniční infrastruktury vytvořili nový spektrální index, který se nazývá Road Extraction Index. Ten se vypočítá pomocí vzorce: $REI = NIR2 - B/NIR2 + B \times NIR2$, kde B je pásmo modrého světla (450-510 nm) a $NIR2$ odpovídá vlnové délce 860-1040 nm.

4.1 Detekce silničních pruhů

V této problematice nebylo provedeno tolik prací, ale mezi nejzásadnější patří například metoda Hinze a Baumgartnera (2003), která se zaměřuje na získávání geometrie silničních pruhů v zastavěné oblasti z leteckých snímků s vysokým rozlišením. Tyto snímky kombinuje s již předem připravenými modely silnice. Tato metoda podle autorů dosahuje až 95% přesnosti a 75% kompletnosti výsledků, ale zmiňují také, že výsledky jsou hodně závislé na přesnosti a kvalitě modelů silnic, které vstupují do výpočtu.

Jin, Feng a Li (2009) jako první využívají jen leteckých snímků, bez použití jiných dodatečných dat. Zaměřují se na silnice mimo města a jakoukoli jinou zástavbu. Tato metoda je založena na detekci pruhů pomocí algoritmu maximální věrohodnosti (maximum likelihood). Takto získané linie jsou následně upraveny algoritmy, které je ztenčí na potřebnou šířku, a následně jsou očištěny od případných nepřesností a šumu.

Ve své diplomové práci Růžička (2020) používá metodu, která získává středy jízdních pruhů. Pro její využití je potřeba leteckých snímků a vektorových dat, které nesou informaci o aktuální poloze komunikace. Tato metoda má tři hlavní kroky:

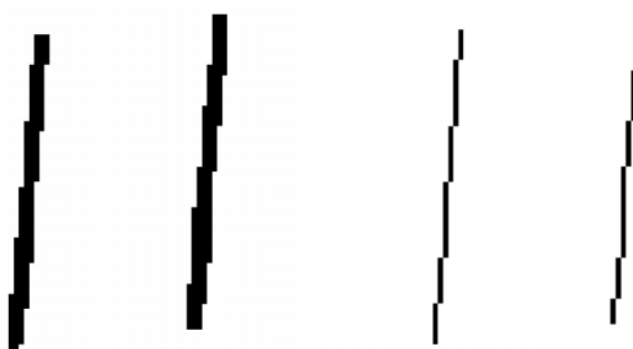
4.1.1 Klasifikace obrazu

První krok je klasifikace obrazu. Ten slouží k rozpoznání silničního povrchu. Z mnoha metod klasifikace byla vybrána object-based image analysis (OBIA). Nejdříve je obraz segmentován pomocí metody multi-resolution segmentation. Tato metoda rozdělí obraz za pomoci několika iterací na postupně se zmenšující části. Tím je zaručeno, že hranice mezi segmenty, které byly vytvářeny při předchozích iteracích, zůstanou stejné po celou dobu segmentace. Tato segmentace je v tomto případě založena na barvě a tvaru.

Další část je klasifikace samotných vzniklých segmentů. Zde se zvolí kritéria, která mají být použita pro klasifikaci. V tomto případě bylo kvůli komplexnosti využito více kritérií. Hlavní kritérium je kontrast mezi tmavým povrchem a světlou silniční čarou. Zde se ale musí oddělit světlé střechy vozů pohybujících se po komunikaci, proto je použito i kritérium tvaru a velikosti.

4.1.2 Úprava rastru

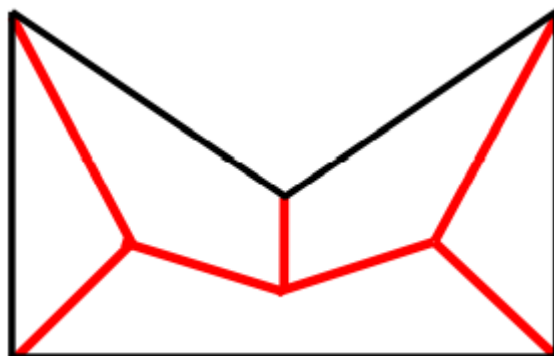
Druhý krok je úprava rastru, který vznikne segmentací. Úpravy se provádějí rovnou na rastru, protože vstupní letecké snímky jsou ve formě rastru a je zbytečné přecházet mezi různými datovými typy, aby nedocházelo ke ztrátě přesnosti. Jedna z těchto úprav je zúžení segmentovaných jízdních pruhů na požadovanou šířku. Metoda je založena na binárním rastru a objekt je postupně od stran ke středu zmenšován (obr. 7).



Obrázek 7 Zúžování rastru
Zdroj: (Růžička 2020)

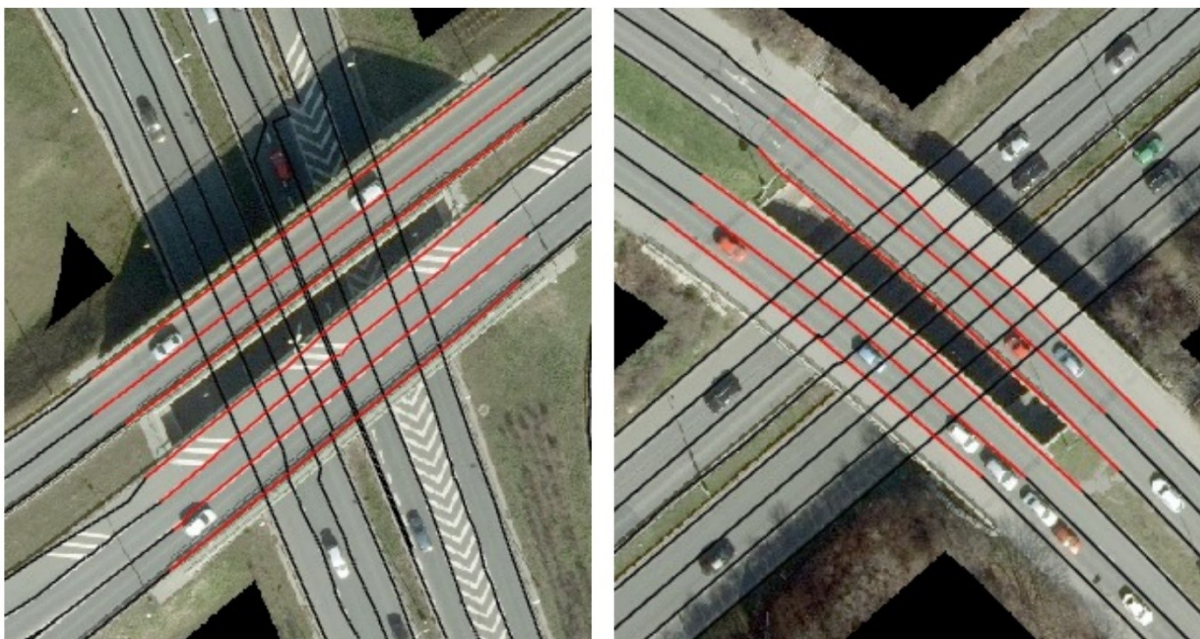
4.1.3 Úprava vektorů

Posledním krokem je převod rastru na vektor a s tím spojené úpravy. Výsledné pruhy jsou z rastru převedeny na polygon a ten je potřeba převést na linii. K tomu slouží skeletonizace. Jedná se o výpočet středu polygonu na základě jeho hran (obr. 8).



Obrázek 8 Ukázka skeletonizace polygonu
Zdroj: (Růžička 2020)

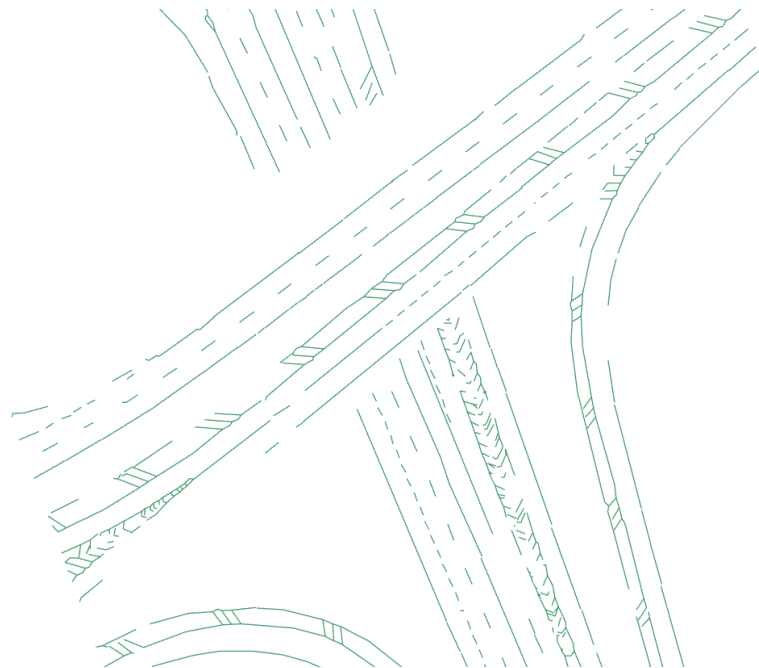
Tato práce pracuje s lokacemi s mimoúrovňovými křižovatkami. Zde nejsou k dispozici snímky v podjezdu, protože jsou zakryty mostem. Algoritmus použitý v této práci s tím neumí pracovat a vzniká tak mezera. Mezera je sice doplněna manuálně, ale tím je ztracena prostorová informace, která z komunikací byla nahoře a která dole (obr. 9).



Obrázek 9 Manuálně doplněné chybějící části silničních pruhů (červeně)
Zdroj: (Růžička 2020)

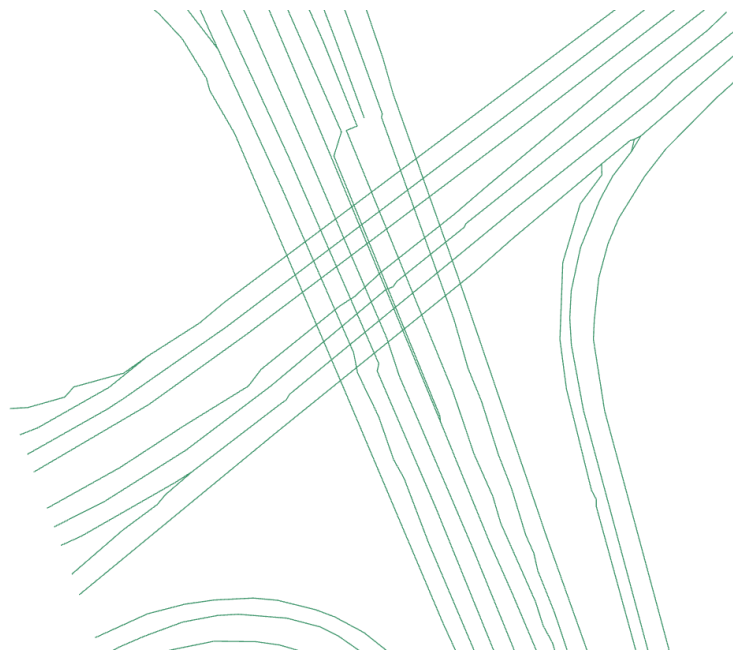
4.1.4 Výstupní data

Výstupem práce Jakuba Růžičky (2020) jsou dva soubory *RSM.shp* a *drivinglines.shp*. *RSM.shp* (obr. 10). Jsou to data, získaná z leteckých snímků, která ještě nebyla spojena do spojitých segmentů. Tato data reprezentují silniční značení a mohou předat informaci o vzájemné poloze jednotlivých silnic, protože ještě neztratila prostorovou informaci (obr. 9).



Obrázek 10 Ukázka dat *RSM.shp*

Druhá data, která jsou výstupem práce Jakuba Růžičky, s názvem *drivinglines.shp* (obr. 11), vznikla úpravou *RSM.shp*. Tato data již jsou spojitější a tím pádem mohou být vhodnější pro případnou automatizaci čtení informací o silničním křížení. Lze z nich vyčíst právě již zmíněný počet jízdních pruhů nebo polohu mostu. Na druhou stranu ztrácejí prostorovou informaci o křížení komunikací.



Obrázek 11 Ukázka dat drivinglines.shp

Tato data poskytují informace pro následnou tvorbu modelu, která je popsána v této práci.

S podobným systémem, který uplatňuje Jakub Růžička (2020), přišel ve své disertační práci Seo (2012). V této práci představil novou metodu kombinující letecké snímky s vektory silnic na pozadí. Při segmentaci pracuje na dvojí úrovni. Nejprve na nízké úrovni a poté na střední (ta odpovídá menšímu měřítku). Na nízké úrovni jsou segmentovány pixely leteckých snímků pomocí iterací a spojovány do skupin, dokud skupiny neodpovídají požadované velikosti. Na střední úrovni jsou identifikovány komunikace a směr jízdy na nich. Poté jsou data z obou úrovní spojena.

5 Software

Pro vytvoření modelu mimoúrovňového silničního křížení bylo využito 2 softwarů. Software Blender, ve kterém je model vytvářen podle zadaných parametrů, a software ArcGIS Pro, ve kterém je poté výsledný model vizualizován.

5.1 Software Blender

Blender je volně dostupný software (free and open-source). Jeho dostupnost a funkčnost na všech možných operačních systémech jako Windows, Mac a Linux zajišťuje široké využití. Nejčastěji je využíván pro 3D modelování, animaci, simulace a třeba i pro motion tracking. Jeho velká výhoda je jeho customizace. Blender nabízí uživatelské rozhraní Python a je tedy možno si upravit Blender dle libosti, vytvořit nové nástroje nebo vytvářet automatické skripty.

5.1.1 Python v Blenderu

Jak již bylo řečeno, Blender poskytuje možnost využívat programovací jazyk Python. Za jeho pomoci se dá Blender plně ovládat, ale dají se také vytvářet nové funkce a zapouštět již připravené skripty. Hlavní knihovna, která obsahuje funkce Blenderu se nazývá *bpy*. S její pomocí lze vytvářet nové objekty buď za pomoci meshe definovaného body, nebo lze používat základní tvary, jako jsou například plocha nebo krychle. Ty se po přidání dají upravovat na požadované tvary (Blender Foundation 2021). Základní funkce knihovny *bpy* jsou tyto:

Context Access (*bpy.context*): přístup k aktuálním datům (vybraným objektům, aktivní oblasti atd.).

Data Access (*bpy.data*): interní data Blenderu (scény, objekty atd.).

Operators (*bpy.ops*): funkce pracující s daty.

Types (*bpy.types*): objekty python třídy, které představují objekty Blenderu.

Property Definitions (*bpy.props*): vlastnosti přiřazené jednotlivým třídám.

5.1.2 Datová struktura v Blenderu

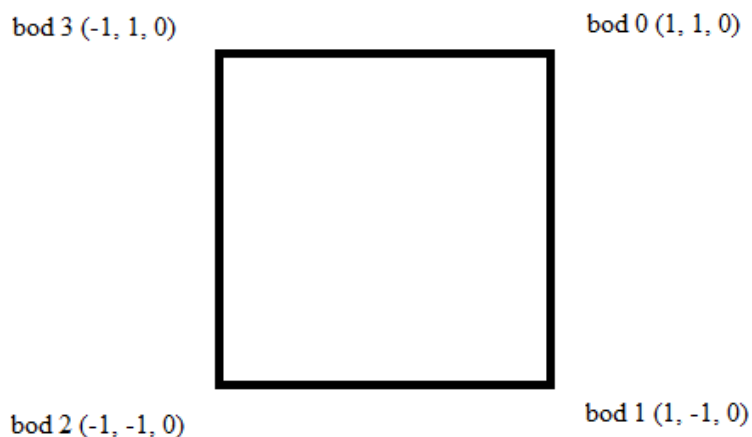
Struktura dat v Blenderu je založena na vrcholech, hranách a plochách. Základním prvkem jsou tedy vrcholy (vertices), které nesou informaci o své poloze x , y a z . Tyto vrcholy jsou při editaci objektů reprezentovány malým oranžovým bodem. Hrany (edges) jsou přímá spojnice dvou vrcholů. Ve výsledném modelu nejsou viditelné. Viditelné jsou pouze ve wireframe zobrazení, kde tvoří kostru objektu. Plochy (faces) slouží k sestavení povrchu objektu. Jsou definovány jako plocha mezi třemi nebo více vrcholy s hranami na okrajích. Dělí se na trojúhelníkové, čtyřúhelníkové a n -úhelníkové. Nejjednodušší práce a vykreslení je s plochami tvořenými třemi vrcholy. Jsou totiž vždy v jedné rovině. Čtyřúhelníkové mají na druhou stranu výhodu, že je lze dobře deformovat, a tudíž jsou vhodné pro deformaci (Blender Foundation 2021).

Vrcholy, hrany a plochy jsou poté uchovávány v polygonové síti (anglicky polygon mesh). Takový mesh lze vytvořit pomocí *bpy* knihovny například takto:

```
verts = [( 1.0,  1.0,  0.0),
          ( 1.0, -1.0,  0.0),
          (-1.0, -1.0,  0.0),
          (-1.0,  1.0,  0.0),]
faces=[(0,1,2,3)]

#tvorba meshe
mesh.from_pydata(verts,[],faces)
```

Na příkladu je vidět, že vrcholy jsou definovány jako list bodů v trojrozměrném prostoru. Plochy se poté definují pomocí listu, který obsahuje n -tice s číslem bodu, které jsou uspořádány v pořadí, v jakém mají být jednotlivé vrcholy za sebou (obr. 12).



Obrázek 12 Příklad vytvořené plochy pomocí vrcholů

5.2 Software ArcGIS Pro

ArcGIS Pro je program pro tvorbu a správu prostorových dat, pro jejich analýzu a pro jejich vizualizaci ve 2D i ve 3D. Software vyvíjí společnost Esri. Podobně jako u Blenderu je s ArcGIS Pro možné pracovat v uživatelském rozhraní, kde je uživateli umožněn přístup k mnoha funkcím na úpravu dat, vytváření geodatabází a jsou mu umožněny různé vizualizace. Ve 2D prostoru je to nejčastěji pomocí map.

5.2.1 Python v ArcGIS Pro

I zde je možnost ovládat většinu funkcí pomocí pythonu. K tomu slouží knihovna funkcí *arcpy*. Python lze používat buď přímo v programu, kde se nachází konzole pythonu, nebo ho lze používat i přímo z příkazového řádku, bez potřeby spustit celý ArcGIS Pro. Pro práci s 3D daty jsou zde funkce z rozšíření 3D Analyst, které je možné volat po aktivaci rozšíření příkazem `arcpy.CheckOutExtension("3D")`. Poté je možné volat funkce `arcpy.3d`.

5.2.2 3D data v ArcGIS Pro

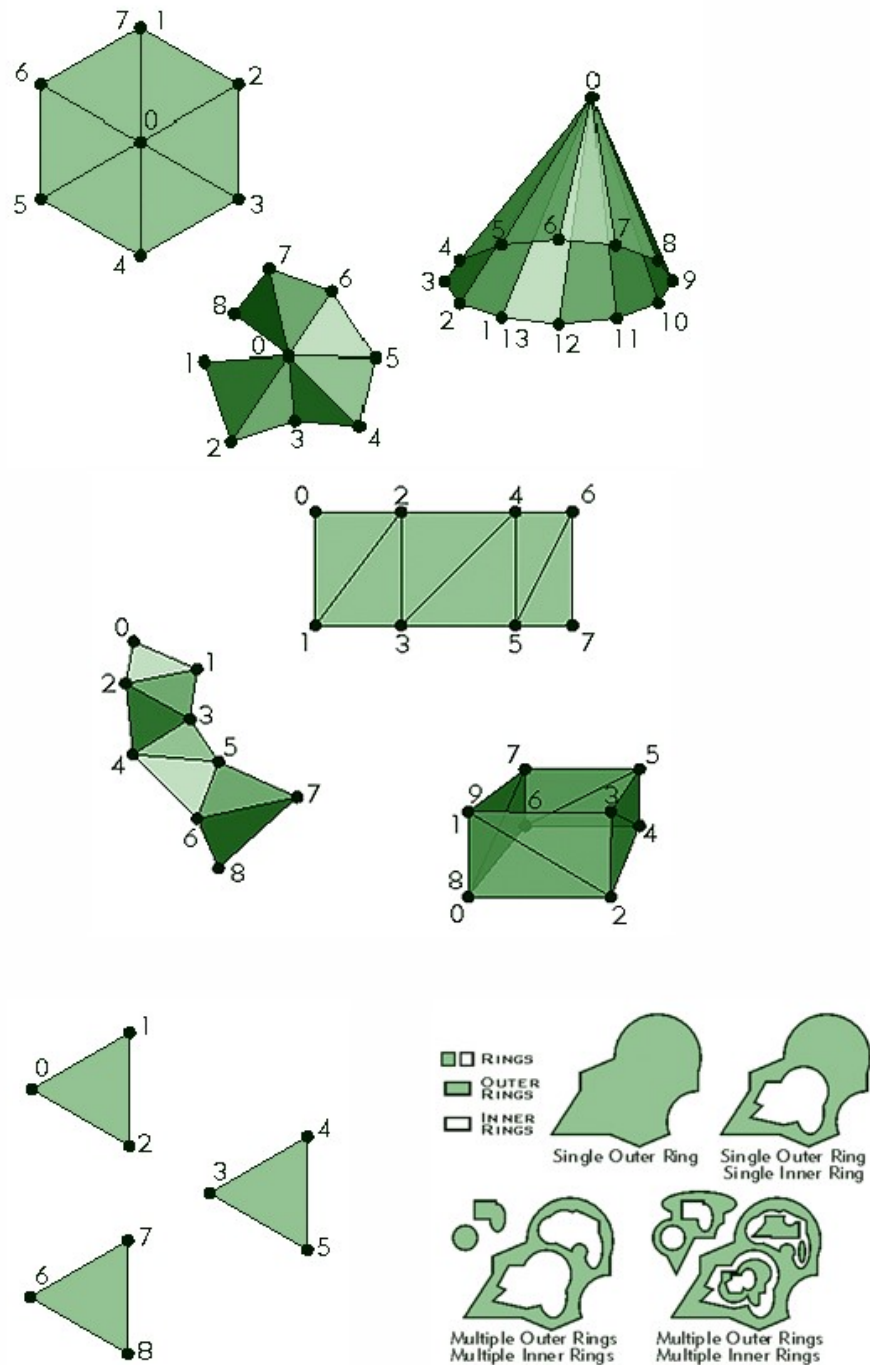
Pro vizualizaci 3D dat slouží takzvaná ArcGIS Pro Scéna. Jsou zde k dispozici 2 režimy scény. První je globální, který slouží k zobrazování velkých oblastí a pracuje se zakřivením zemského povrchu. Druhý je lokální, který slouží pro práci s menšími lokacemi

v projektovaném souřadnicovém systému, nebo v případech, kdy lze zakřivení Země zanedbat. Do scény je možné nahrát data, která nesou informaci o tvaru reliéfu a jeho nadmořské výšce. Podporované formáty pro tato data jsou rastr, TIN nebo webová elevační vrstva (Esri 2021).

3D data jsou zde uchovávána ve formátu multipatch feature. Jedná se sbírku tzv. *patches*, které reprezentují hranice 3D objektu pomocí jediného řádku v geodatabázi. *Patches* ukládají texturu, barvu a geometrickou informaci reprezentující části objektu. Geometrická informace může být zajištěna pomocí trojúhelníků, trojúhelníkových vějířů, trojúhelníkových proužků nebo kroužků (obr. 13).

Všechny *patches*, ze kterých se objekt skládá, musí mít stejnou orientaci bodů, které ji definují. Orientace je ukládána proti směru hodinových ručiček. Jednotlivé *patches* se nesmějí navzájem protínat a nesmějí v nich být mezery či prázdná místa (Esri 2021).

Multipatch feature může být vytvořena importem již existujících 3D modelů v podporovaných formátech. Jedná se o formáty 3D Studio Max (.3ds), SketchUp (.skp), VRML a GeoVRML (.wrl), OpenFlight (.flt), a COLLADA (.dae).



Obrázek 13 Způsoby uchovávání geometrické informace v multipatch souborech.
 Zdroj : <https://desktop.arcgis.com/en/arcmap/latest/extensions/3d-analyst/multipatches.htm>

6 Modelování mostů

Pro vytvoření modelu silničního mostu byl využit software Blender. Jak bylo zmíněno v kapitole 5.1, umožňuje práci s pythonem a tím pádem automatizaci vytváření modelů i jiných mostů na základě parametrů. K získání základních geometrických údajů o mostě byla využita data z práce Jakuba Růžičky (2020), zmíněné v kapitole 4.1. Z těchto dat byla získána poloha jednotlivých mostů a následně počet jízdních pruhů a jejich přibližná šířka. Další geometrické údaje byly získány z modelu vytvořeného pomocí metody SfM zmíněné v kapitole 3.

6.1 Model vytvořený pomocí SfM

Jak již bylo zmíněno výše, lze pomocí metody Structure from Motion vytvořit 3D model objektu z fotografií. Takto vytvořený model byl tedy využit k získání geometrických údajů mostu, jako jsou výška nadjezdu a jeho šířka.

6.1.1 Lokalizace mostu

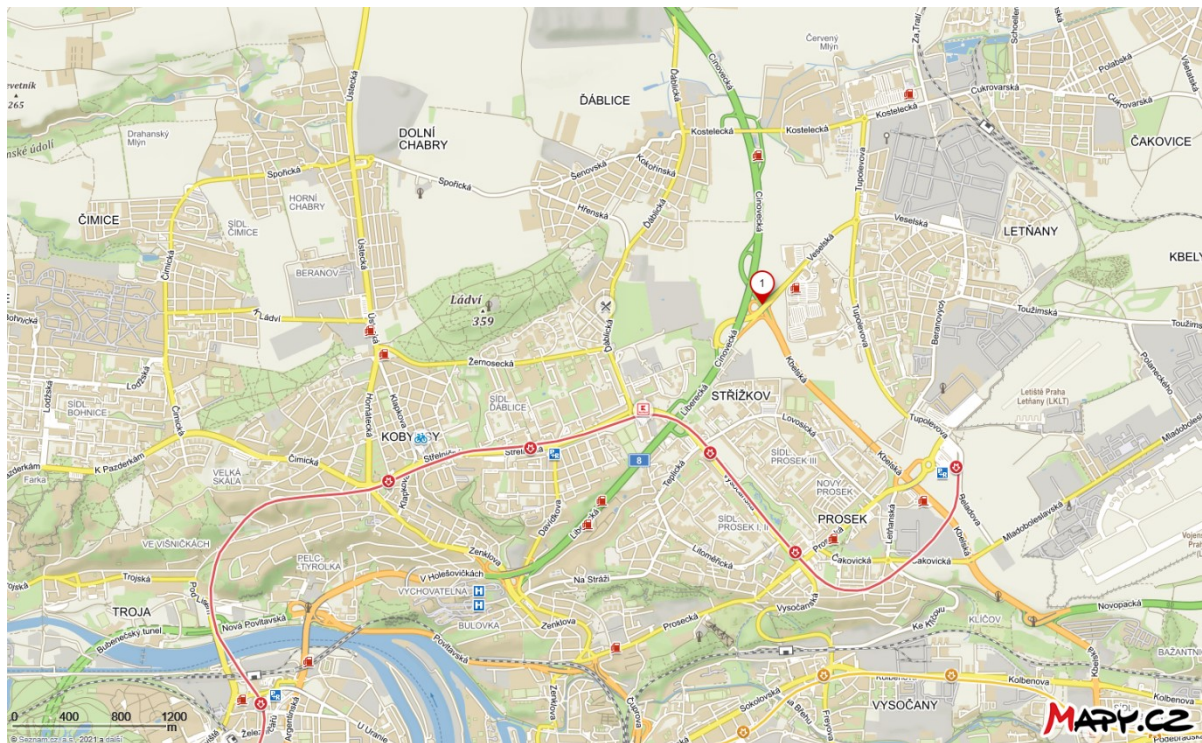
Nejprve bylo potřeba zvolit konkrétní most, za jehož pomoci se budou generovat obecně i další mosty. Tyto mosty již nebudou odpovídat tolik realitě, protože mohou mít jinou konstrukci, ale jako zjednodušený model by mohly v nějakých případech posloužit. Pro co největší přesnost bylo voleno mezi mosty, na kterých byl testován postup získávání jízdních pruhů z leteckých snímků v práci Jakuba Růžičky (2020). Nakonec byl zvolen most ulice Veselská přes ulici Kbelská v Praze-Střížkově (obr. 14).

Tento most byl zvolen, protože se nachází na lépe dostupném místě než druhý most. Druhý most by byl vhodnější díky své velikosti. Je menší, a proto by nemuselo být zapotřebí tak velkého množství snímků, ale naneštěstí se se jeho podjezd nachází v údolí a je obklopen velkým množstvím vegetace. Tyto podmínky zamezují čistému výhledu na most ze všech stran a tím pádem nebyl vhodný k vytvoření pomocného modelu.

6.1.2 Snímkování

Snímky byly pořízeny na kameru Sony CyberShot DSC-H300 s rozlišením fotografie 1920×1080 . Bylo pořízeno 185 snímků mostu, tak, aby mezi jednotlivými pořízenými snímky byla stále stejná vzdálenost. Dále byla udržována konstantní vzdálenost od mostu.

Most byl nakonec nasnímkován jen ze 3 stran. Poslední 4. strana byla vynechána kvůli neprůchodnosti terénu.



Obrázek 14 Mapa s modelovaným mostem
Zdroj: Mapy.cz

Pro zasazení do souřadnicového systému byly rozestaveny před most malé dopravní kužely, u kterých byly pomocí geodetické GPS odečteny souřadnice. Souřadnice byly odečítány i každých 10 fotografiích k určení polohy kamery.

6.1.3 Tvorba modelu a výsledné rozměry mostu

K vytvoření modelu mostu z pořízených snímků byl použit program Agisoft Metashape Professional. Nejprve byla v programu provedena kalibrace fotoaparátu pomocí již připraveného kalibračního pole. Poté byly snímky zorientovány a zpřesněny pomocí společných bodů, které byly zadány manuálně. Jednalo se například o hrany mostu, či dopravní značky. Nakonec byly přidány souřadnice k jednotlivým polohám kamer a vřícovacím bodům. Jako výsledek bylo vráceno husté bodové mračno, které bylo očištěno od přebytečných bodů a od výrazných chyb.

Výsledné mračno již nadále nebylo možno zpracovat například do mesh modelu, protože bodů nebylo dostatek.



Obrázek 15 Výsledný model z bodového mračna

Na obrázku 15 je vidět výsledné bodové mračno. Lze zde vidět, že na přední straně modelu je bodů výrazně méně. Je to způsobeno tím, že z této strany nebylo možné snímkovat. Model také není kompletní ze shora. Z modelu však je možné odečíst nějaké základní rozměry mostu. Bylo možné odečíst výšku nadjezdu a jeho šířku ve směru osy Z, rozměry menších nosných sloupů v podjezdu a šířku silnic v podjezdu. Šířka nadjezdu v kolmém vodorovném směru byla ztracena v nepřesnostech a byla tedy odečtena z leteckých snímků a z dat s jízdními pruhy.

6.2 Generátor mostu

Pro generování více mostů automaticky a nejen jednoho podle zjištěných rozměrů byl vytvořen add-on pro Blender s názvem Generátor mostů, který umožňuje vygenerovat různé mosty na základě počtu pruhů v podjezdu a nadjezdu. Add-on také zvládne lehce měnit úhly mezi podjezdem a nadjezdem, protože křížení není vždy kolmé. Tento add-on byl vytvořen na základě dat z předchozí kapitoly a z dat z práce Jakuba Růžičky (2020). To umožnilo větší věrohodnost konstrukce modelu tohoto konkrétního mostu, ale když bude add-on využit pro jiný most, než je modelový most, tak zde budou větší nepřesnosti a model bude jen orientační. Doplněk pro Blender vytvoří geometrii mostu a poté přidá textury, které byly získány při snímkování modelového mostu.

Jednotlivé části rozšíření pro Blender byly vytvořeny pomocí programovacího jazyka Python, kterým Blender disponuje.

6.2.1 Metoda generování

Doplňěk využívá některé techniky procedurálního modelování. V tomto případě především pravidlo dělení. Kvůli tomu, že není zapotřebí generovat více variant mostů, ale spíše se mění jen velikost, nebylo potřeba využívat více pravidel pro modelování. Procedurální modelování má také nevýhodu v tom, že nad výsledkem nemá uživatel úplnou kontrolu a je určeno spíše ke generování náhodných objektů. Proto bylo pravidlo dělení upraveno do požadované formy a jedná se tedy spíše o inspiraci.

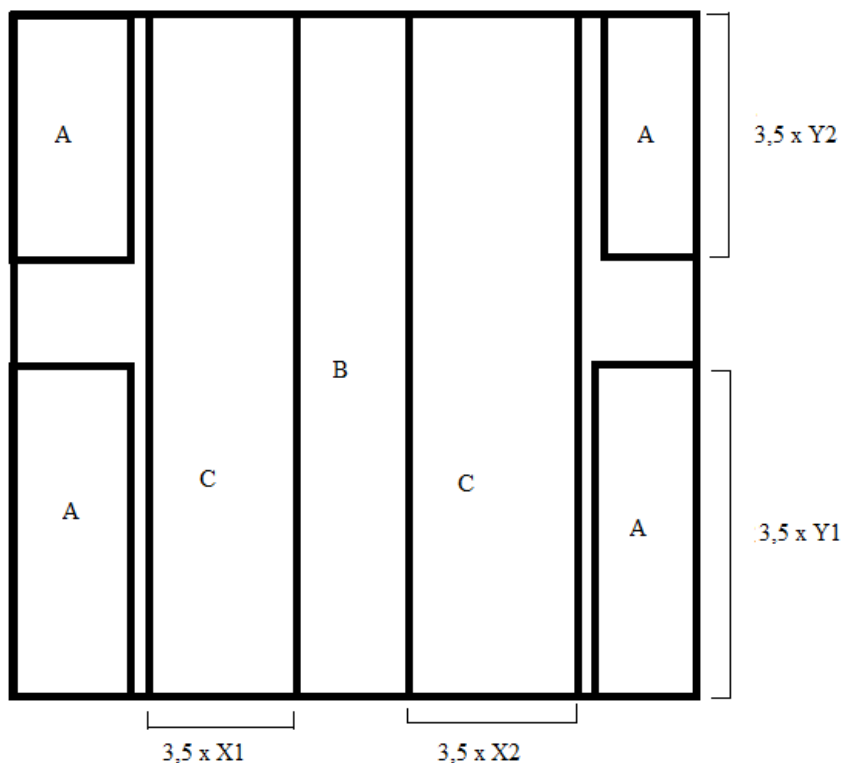
Pravidlo dělení tedy vychází z gramatiky split grammars zmíněné v kapitole 2.1.3. Toto pravidlo bylo definováno tak, že představa výsledného mostu byla rozdělena na již nedělitelné části. Most byl tedy rozdělen postupně na podjezd a nadjezd, ty byly následně rozděleny na silnice a svodidla. Silnice byla rozdělena na jednotlivé pruhy a svodidla na jednotlivé sloupky atd. Dále je zde implementováno kopírování jednotlivých základních prvků, jako jsou sloupky od svodidel nebo přerušovaná čára mezi pruhy. Toto umožňuje rychlejší generování stejných prvků.

6.2.2 Tvorba půdorysu

Pro generování mostu je potřeba jako první zjistit jeho rozměry, respektive rozměry půdorysu. Parametry pro určení velikosti půdorysu jako jsou šířka jízdních pruhů byly odečteny z leteckých snímků a z dat *drivinglines.shp*. Šířka nosných sloupů a mezera mezi směry byly získány z modelu vytvořeného pomocí metody SfM. Velikost půdorysu se vždy mění dle množství pruhů v podjezdu a v nadjezdu. Na obrázku č. 16 je náčrtek půdorysu mostu. Písmeno *A* značí nosné sloupky, na kterých je nadjezd. Jeho šířka ve směru osy *X* je pevně daná z vytvořeného modelu. Šířka ve směru osy *Y* se mění na základě počtu pruhů nadjezdu *Y1* a *Y2*. Písmeno *B* pak značí dělení mezi silnicemi v podjezdu. Jeho rozměr ve směru osy *X* je také pevně dán z modelu. Ve směru osy *Y* se znovu dopočítává z počtu pruhů *Y1* a *Y2*. Poslední část náčrtku je označena písmenem *C* a jedná se o komunikace v podjezdu. Jejich šířka je závislá na počtu pruhů v podjezdu *X1* a *X2*.

Jak již bylo řečeno, šířka jízdních pruhů byla odečtena z dat *drivniglines.shp* a byla zaokrouhlena na 3,5 metru.

Po definování půdorysu a jeho rozměrů je tedy možné vytvořit samotný půdorys pomocí Pythonu v Blenderu. Jedná se o pomocný objekt, který jen předává informace pro další generování jednotlivých objektů.



Obrázek 16 Náčrt půdorysu mostu

6.2.3 Tvorba nosných sloupů

Po získání rozměrů půdorysu mostu je zapotřebí vytvořit modely základních prvků mostní konstrukce. Jako první jsou vytvořeny čtyři nosné sloupy. Toto modelování probíhá jednoduše pomocí funkcí v Blenderu. Nejprve se přidají základní tvary. V tomto případě se jedná o čtyři krychle (jedna pro každý nosný sloup), jejichž umístění se dopočítá z rozměrů daných půdorysem. Následně jsou tyto základní tvary upravovány dalšími funkcemi, aby sloupy dosáhly odpovídající velikosti při daném počtu jízdních pruhů.

6.2.4 Tvorba nadjezdu

Dalším krokem je vytvoření konstrukce nadjezdu. Ta je vytvořena přidáním krychle pro nadjezd v každém směru. Následně je opět zvětšena, aby velikostí odpovídala půdorysu.

Šířka nadjezdu ve směru osy Z je zde odvozena z dat získaných pomocí metody SfM. Mezi jednotlivými nadjezdy je vytvořena mezera, která je velká 5 metrů. Tato velikost byla odečtena z leteckých snímků.

6.2.5 Silnice a jízdní pruhy

Pro vytvoření modelu silnice a značení jízdních pruhu na ní bylo využito dvou funkcí. U vytváření jednotlivých komunikací vše probíhalo jako v předchozích případech. Opět byly přidány základní tvary, ale tentokrát ne krychle, ale jen jednoduché dvourozměrné plochy. Model silnice je zde pro každý směr v podjezdu a nadjezdu vytvářen jako celek. Na jednotlivé jízdní pruhy, jejichž počet určil uživatel při zadávání parametrů, je vozovka rozdělena až pomocí silničního značení.

U silničního značení je zapotřebí zajistit, aby se vždy rozprostíralo po celé délce vozovky se stejně velkými rozestupy jak mezi jednotlivými pruhy, tak i mezi oddíly přerušované čáry. Mezery mezi jednotlivými částmi přerušované čáry a jejich počet jsou vypočítávány vždy z aktuální délky silnice, aby se pruhy co nejvíce blížily předdefinovaným rozstupům, ale aby zároveň nepřesahovaly mimo, či jich nebylo příliš málo. Následně může být tedy vytvořen jeden segment čáry a ten poté zkopírován pomocí cyklu.

6.2.6 Svodidla

Svodidla jsou tvořena ze dvou částí. První jsou sloupky a druhá část je samotné svodidlo. Sloupku podléhají stejnému problému, který nastával u tvoření značení jízdních pruhů. Tento problém je tedy vyřešen obdobně. Samotné svodidlo je poté opět základní krychle upravená tak, aby odpovídala velikosti modelu.

6.2.7 Detailní prvky

Poslední částí modelu, kterou bylo potřeba vytvořit, jsou malé sloupky v podjezdu a obrubník mezi směry v podjezdu. Obrubník je nastaven vždy na stejný rozměr odečtený ze získaných dat pomocí SfM. Pro vytvoření jednotlivých malých sloupů je opět využito pravidlo kopírování a jsou rozmístěny pod každý pruh nadjezdu.

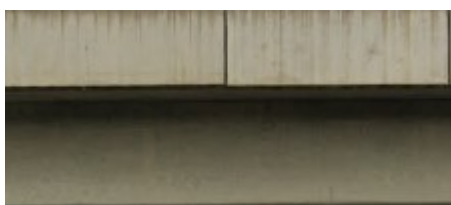
6.2.8 Textury

Další součástí tvorby modelu bylo přidání textur k jednotlivým částem. Při tvorbě jednotlivých částí konstrukce bylo vždy do atributu objektu přidán název materiálu. U některých objektů bylo potřeba více materiálů. To bylo zajištěno jeho rozložením na jednotlivé části. Bylo vytvořeno několik základních materiálů. Tři nejjednodušší s názvy „Asfalt“, „Beton“ a „Pruh“ byly na vytvoření textury nejsnazší. Jedná se jen o nabarvení jednotlivých částí předdefinovanou barvou. Poté bylo zapotřebí vytvořit složitější textury, které by výslednému modelu přidaly detailní prvky.

Složitější textury byly vytvořeny pomocí snímků, které byly pořízeny v terénu. Ze snímků byla vybrána vždy část mostu, ke které byla potřeba přidat texturu, a následně byl snímek oříznut na požadovanou část.

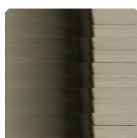
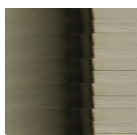
6.2.8.1 Nadjezd

Textura nadjezdu se skládá ze 2 snímků. Z čelní strany bylo zapotřebí přidat texturu betonových panelů a zároveň i nasimulovat stín. Za tímto účelem byla ze snímků vyříznuta část mostní konstrukce (obr. 17).



Obrázek 17 Ukázka zdroje textury pro hranu nadjezdu

Takto oříznutý snímek bylo následně ještě třeba upravit, aby seděl na danou velikost nadjezdu. Toho bylo docíleno v Blenderu pomocí funkce s názvem Bake. Textura byla ručně nastavena na most, aby seděla podle potřeby a poté byla zapečena. Výsledek lze vidět na obrázku 18.

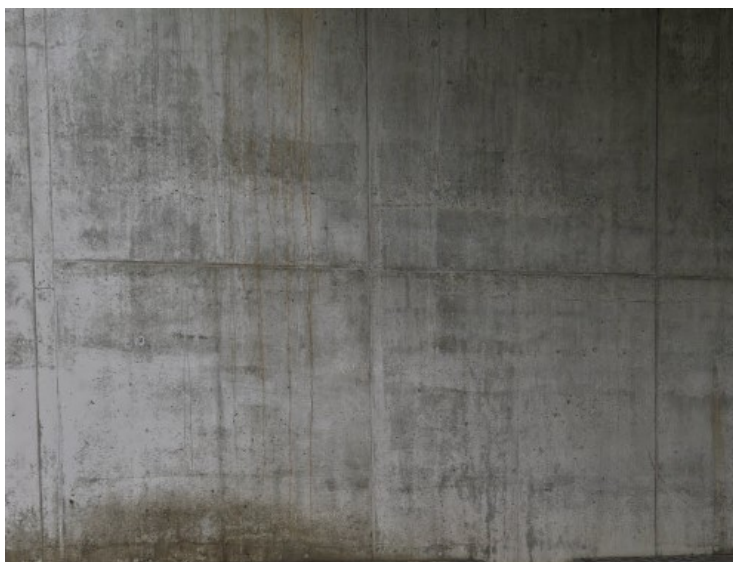


Obrázek 18 Upravená textura hrany nadjezdu pomocí Bake

Takto upravená textura již při automatickém vytvoření modelu drží správnou orientaci a pozici.

6.2.8.2 Nosné sloupy

Další textura, která byla připravena podobně jako textura pro hranu nadjezdu, byla textura pro nosné sloupy (obr. 19). Zde nebylo potřeba texturovat jen z určitých částí, ale celý objekt sloupu. Znovu byla použita funkce Bake a výsledek lze vidět na obrázku 20.



Obrázek 19 Textura nosného sloupu



Obrázek 20 Upravená textura nosného sloupu pomocí Bake

Tímto způsobem byly vytvořeny ještě textury pro menší sloupy v podjezdu, obrubník a svodidla. Vždy bylo přihlédnuto k tomu, zda je třeba texturovat celý objekt, nebo jen některé části.

Tyto připravené textury jsou přiloženy v souboru se samotným kódem rozšíření Blenderu pro tvorbu silničních mostů. Tyto textury je nutno vždy před vytvořením modelu načíst. Díky tomu je možné je zaměnit za libovolné jiné textury se stejným názvem.

6.2.9 Uživatelské rozhraní a ovládání

Vzhledem k tomu, že Generátor mostů je rozšíření pro program Blender, bylo potřeba vytvořit uživatelské rozhraní k jeho ovládání. K tomu byla vytvořena třída definující polohu jednotlivých tlačítek. Část lze vidět v následující ukázce kódu.

```
class Generator_mostu(bpy.types.Panel):
    #vytvoření tlačítek na panelu
    bl_label = "Generátor mostů"
    bl_idname = "SCENE_PT_layout"
    bl_space_type = 'PROPERTIES'
    bl_region_type = 'WINDOW'
    bl_context = "scene"

    props()
    def draw(self, context):
```

```

layout = self.layout

scene = context.scene

# jednotlivá tlačítka.
layout.label(text=" Počet pruhů:")

row = layout.row()
row.prop(scene, "Nadjezd_pruhy_vpravo")

```

Pomocí tohoto kódu je vytvořen jeden řádek, se kterým může uživatel interagovat. Zde se jedná o políčko s názvem `Nadjezd_pruhy_vpravo`.

Samotné vytvoření interaktivního pole nestačí. Je ještě zapotřebí definovat, v jakém formátu bude pole formátu, jaký je interval možných hodnot atd. K tomu slouží funkce z knihovny `bpy.types`. Na následující ukázce kódu je definováno políčko pro „Nadjezd_pruhy_vpravo“.

```

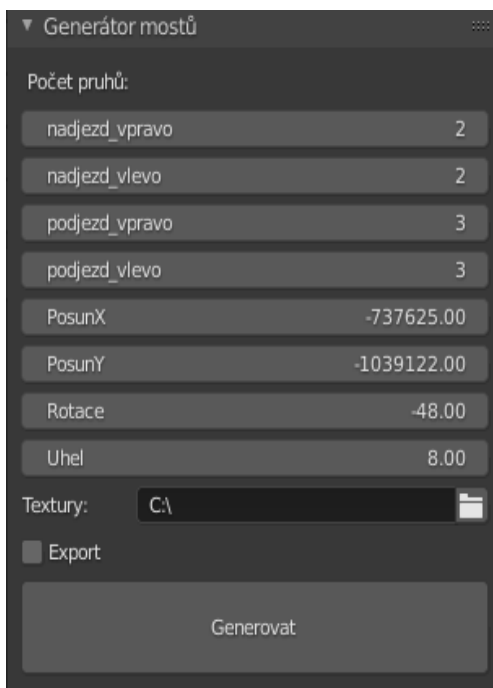
def props():
    # int - čísla
    bpy.types.Scene.Nadjezd_pruhy_vpravo = bpy.props.IntProperty(
        name = "nadjezd_vpravo",
        description = "počet pruhů nadjezu v pravém směru jízdy",
        default=2,
        min = 1,
        max = 99)

```

Lze zde vidět definování formátu pomocí `IntProperty`, která zajistí, že se jedná o vstup ve formátu integer. Dále se zde definuje interval a výchozí hodnota.

6.2.9.1 Ovládání

Tímto způsobem byly vytvořeny všechny potřebné vstupy a tlačítka pro uživatelské rozhraní, až vznikl výsledný ovládací panel (obr. 21).



Obrázek 21 Ovládací panel pro rozšíření Generátor mostů

První čtyři políčka slouží k zadání počtu pruhů v jednotlivých směrech jízdy v podjezdu a nadjezdu. Zde je možná zadat hodnota 1 až 99. Další políčka *PosunX*, *PosunY* a *Rotace* slouží k zařazení modelu do souřadnicového systému. Jedná se jen o jednoduchý posun po ose *X* a *Y* o určitý počet metrů a o rotaci celého objektu o daný úhel. Políčka *Uhel* slouží ke změně úhlu mezi podjezdem a nadjezdem. Ne vždy je křížení kolmé a občas je zapotřebí úhel změnit. Textové pole *Textury* slouží k zadání absolutní cesty ke složce s texturami. Add-on lze díky tomu spustit na různých zařízeních a také je možné textury vyměnit za vlastní. Pomocí zaškrtnutí tlačítka *Export* lze model exportovat ve formátu *.dae* a posledním tlačítkem *Generovat* je celý proces modelování spuštěn.

6.2.10 Instalace

Rozšíření pro Blender je nejprve nutno nainstalovat. Program totiž není schopen běžet mimo prostředí Blender, kde se nacházejí funkce *bpy*. Instalace probíhá v samotném Blenderu v záložce *Edit* → *Preferences...* → *Add-ons* → *Install...* Zde je zapotřebí vybrat příložený soubor s názvem *generator_mostu.py* a doplněk se nainstaluje.

Samotné uživatelské rozhraní by se poté mělo nacházet v záložce *Scene properties*. Doplněk je tvořen pro verzi Blenderu 2.92.0, a proto by měl být instalován na tuto verzi. Na jiných verzích nemusí doplněk fungovat.

6.3 Vizualizace

Model, který byl vytvořen a následně exportován ve formátu *.dae*, je zapotřebí vizualizovat. Jako program pro vizualizaci modelu byl vybrán ArcGIS Pro. Tento software disponuje Scénou, která umožňuje práci s daty ve 3D. Tento program byl vybrán také díky tomu, že lze hodně procesů automatizovat pomocí pythonu. Právě pro automatizaci převodu modelu z Blenderu a jeho následnou vizualizaci byl vytvořen skript s názvem *vizualizace.py*.

Funkce tohoto skriptu jsou načíst modely ve formátu *.dae* a nahrát je do geodatabáze jako multipatch. Dále skript vytvoří model povrchu ve formátu TIN.

Tento skript je vytvořen tak, aby byl spouštěn pomocí příkazového řádku. Při spouštění je zapotřebí napsat název skriptu a za něj 4 atributy. 1. atribut je cesta ke složce s modely, 2. atribut je složka, do které se mají ukládat upravené soubory pro vizualizaci. 3. atribut je název těchto souborů a poslední atribut je složka se vstupními daty pro tvorbu TINu.

6.3.1 Tvorba multipatch souborů

Prvním krokem je načtení modelů ze složky. Tento skript je napsán tak, aby zvládl pracovat s více modely naráz. Proto si z prvního zadaného atributu vezme cestu ke složce, která obsahuje jeden a více modelů. Před samotným převodem modelů na multipatch soubory je zapotřebí vytvořit geodatabázi, aby si modely zachovaly textury. Kdyby byly uloženy jen jako shapefile, jejich textura by se ztratila.

Vytvoření geodatabáze probíhá pomocí funkce z knihovny *arcpy*. Na následující ukázce kódu je tato funkce použita. První atribut je místo, kde bude databáze vytvořena, a druhý atribut je její název.

```
arcpy.management.CreateFileGDB(arcpy.env.workspace, out_name)
```

Nyní je již možné modely převést na multipatch a nahrát do geodatabáze. K tomu slouží funkce *Import 3D files*, která se v pythonu volá takto:

```
arcpy.ddd.Import3DFiles(input_list,out_name + ".gdb\\" + out_name, "#", sr)
```

První atribut je list všech souborů, které chceme povést na multipatch. Tento list byl získán pomocí funkce *glob.glob*, která vytvoří seznam všech souborů ve složce. Druhý atribut je název geodatabáze, kam budou soubory uloženy. Poslední atribut je souřadnicový systém,

ve kterém jsou modely vytvořeny. V tomto případě se jedná o fixní parametr a je pracováno s S-JTSK.

6.3.2 Tvorba surface TINu

Scéna ArcGIS Pro umožňuje vizualizovat jednotlivé prvky na zemském povrchu. Tato Scéna disponuje svým vlastním modelem povrchu, ale ten k vizualizaci zde prezentovaných dat není vhodný. Jednotlivé modely se totiž zanořují pod povrch, nebo jsou některé části naopak nad povrchem. Modely jsou totiž tvořeny rovnoběžně s plochou mezi osami X a Y. Tento problém řeší vytvoření a použití vlastního modelu povrchu. Zde byl zvolen model povrchu ve formátu TIN. Hlavní myšlenka tohoto procesu spočívá v tom, že okolo jednotlivých modelů je vytvořena malá oblast, která má konstantní nadmořskou výšku a je rovnoběžná s rovinou os X a Y.

Prvním krokem je vytvoření pomocného TINu, který nese informace o tvaru povrchu bez ohledu na modely mostů. Toho je docíleno funkcí Create TIN, která se v pythonu volá pomocí `arcpy.ddd.CreateTin`. Data pro vytvoření TINu musí splňovat následující podmínky: Ve složce s daty se musejí nacházet dvojice data. První jsou bodová data *VyskoveKoty.shp* a druhá jsou liniová data s názvem *Vrstevnice.shp*. Oboje data musejí obsahovat atribut *VYSKA*. Zde bylo konkrétně pracováno s daty z databáze ArcČr 500, ale mohou být použita jakákoli data splňující tyto podmínky. Po vytvoření pomocného TINu je potřeba získat půdorys jednotlivých modelů. K tomu slouží následující funkce:

```
arcpy.ddd.MultiPatchFootprint(out_name + ".gdb\\" + out_name, 'footprint.shp')
```

Tato funkce promítne multipatch soubor na rovinu os X a Y a vytvoří shapefile nesoucí informaci o půdorysu objektu. Následně je potřeba přidat informaci o průměrné nadmořské výšce, která je v území tohoto půdorysu. K tomu slouží následující funkce:

```
arcpy.ddd.AddSurfaceInformation('footprint.shp', 'tin_pomoc', 'Z_MEAN', 'LINEAR')
```

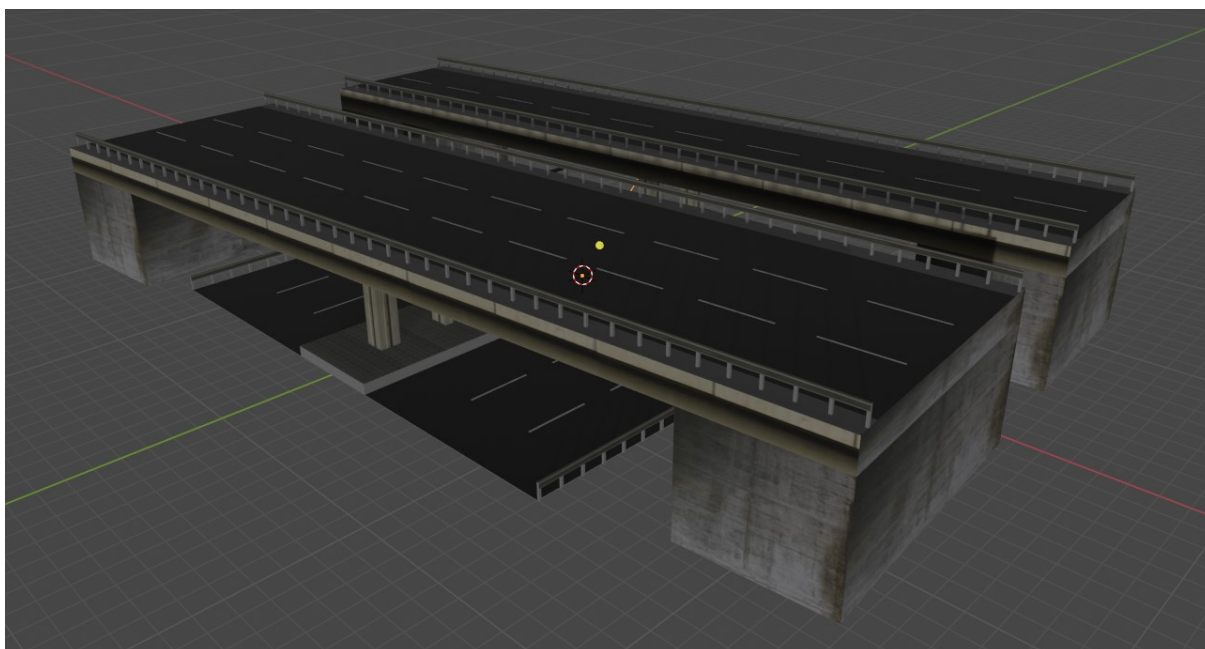
Za pomoci těchto dat již lze vytvořit výsledný surface TIN. Ten je vytvořen ze stejných dat jako pomocný TIN vytvořený v předešlých krocích, ale je zde ještě přidán shapefile s půdorysy.

Nyní jsou již všechno připraveno k vizualizaci. Modely uložené v geodatabázi se přidávají do scény a v záložce *Properties* → *Elevation* se zvolí *Features are on the ground*. Poté je

potřeba ještě vyměnit model povrchu pomocí *Add elevation source* a modely jsou připraveny.

7 Výsledky

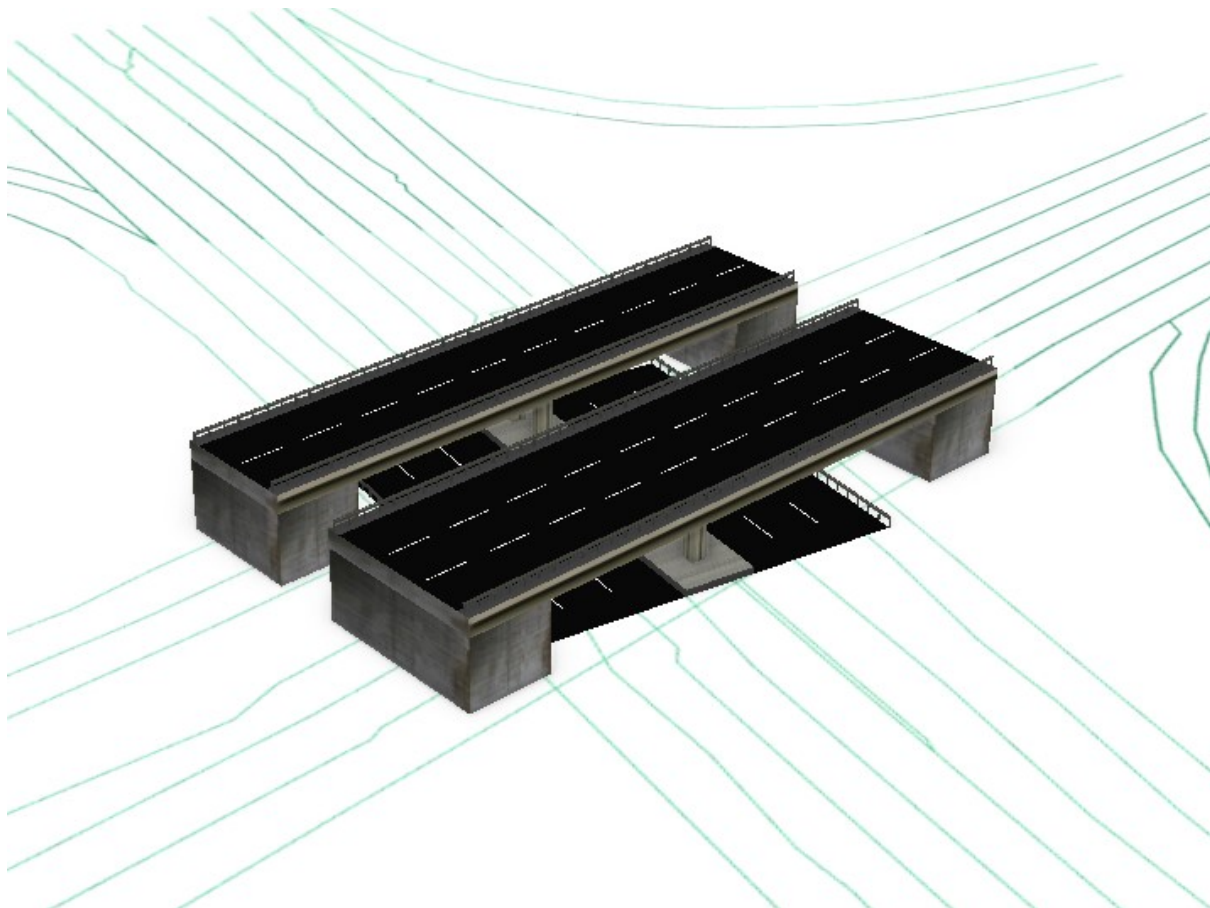
Výsledkem práce jsou dva skripty. Jeden s názvem *generator_mostu.py*, který po instalaci do Blenderu přidá doplněk schopný tvořit modely silničního mimoúrovňového křížení. Druhý skript s názvem *vizualizace.py* připravuje data k vizualizaci. Po instalaci prvního skriptu a nastavení počtu pruhů, který byl získán z dat *drivinglines.shp* popsanych v kapitole 4.1.4, byl vygenerován první most, podle kterého byl celý doplněk dimenzován. Na obrázku 22 je vidět, jak tento most vypadá v prostředí Blenderu. Bylo zde nastaveno, aby měl 3 pruhy v levém horním nadjezdu, 2 v pravém horním nadjezdu a po třech v obou směrech v podjezdu.



Obrázek 22 Model vygenerovaný v Blenderu

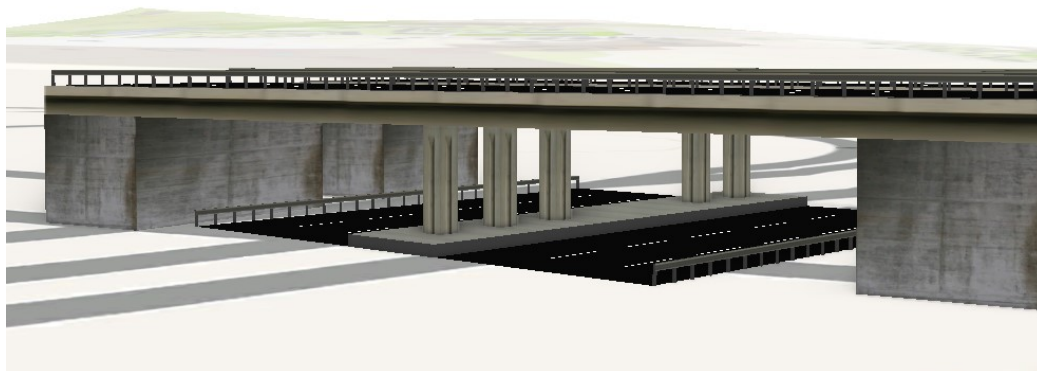
Parametry pro posun a zařazení do souřadnicového systému byly odečteny manuálně z dat *drivinglines.shp*.

Tento model byl poté pomocí skriptu *vizualizace.py* připraven pro vizualizaci v programu ArcGIS Pro. Ve Scéně ArcGIS Pro bylo poté manuálně nastaveno, aby se zobrazovaly modely na povrchu. Povrch byl nahrazen vlastním TINem, který byl vytvořen *vizualizace.py*. Nakonec ještě byly přidány jízdní pruhy z dat *drivinglines.shp* (obr. 23).



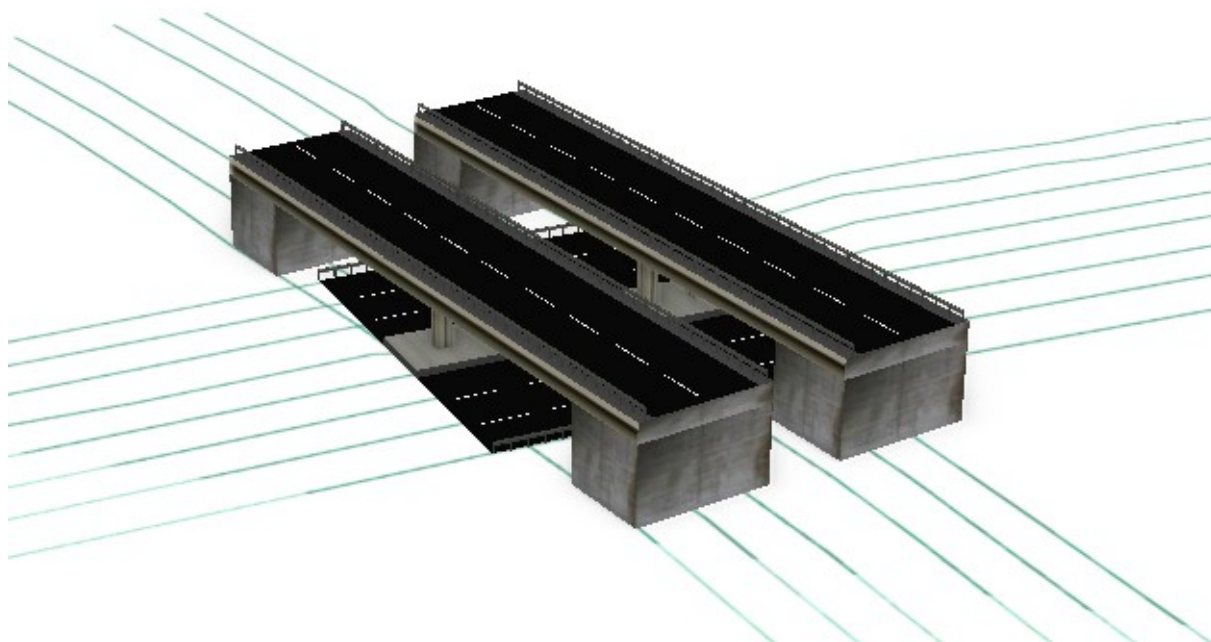
Obrázek 23 Vizualizace ve Scéně ArcGIS Pro

Na dalším obrázku (obr. 24) je ukázán model stejného mostu z pohledu na podjezd a také na tvar reliéfu. Lze zde vidět, že most se nenoří do země.



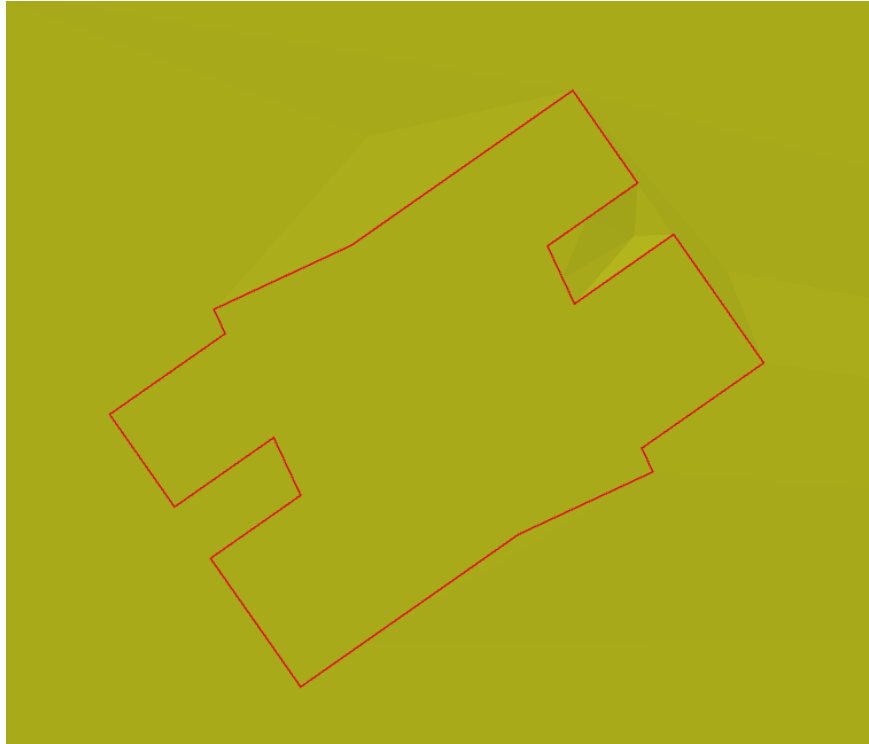
Obrázek 24 Vizualizace podjezdu

Nakonec byly provedeny vizualizace s druhým mostem (obr. 25). zde byly nastaveny 2 pruhy v obou směrech v nadjezdu a 3 pruhy v obou směrech v podjezdu.

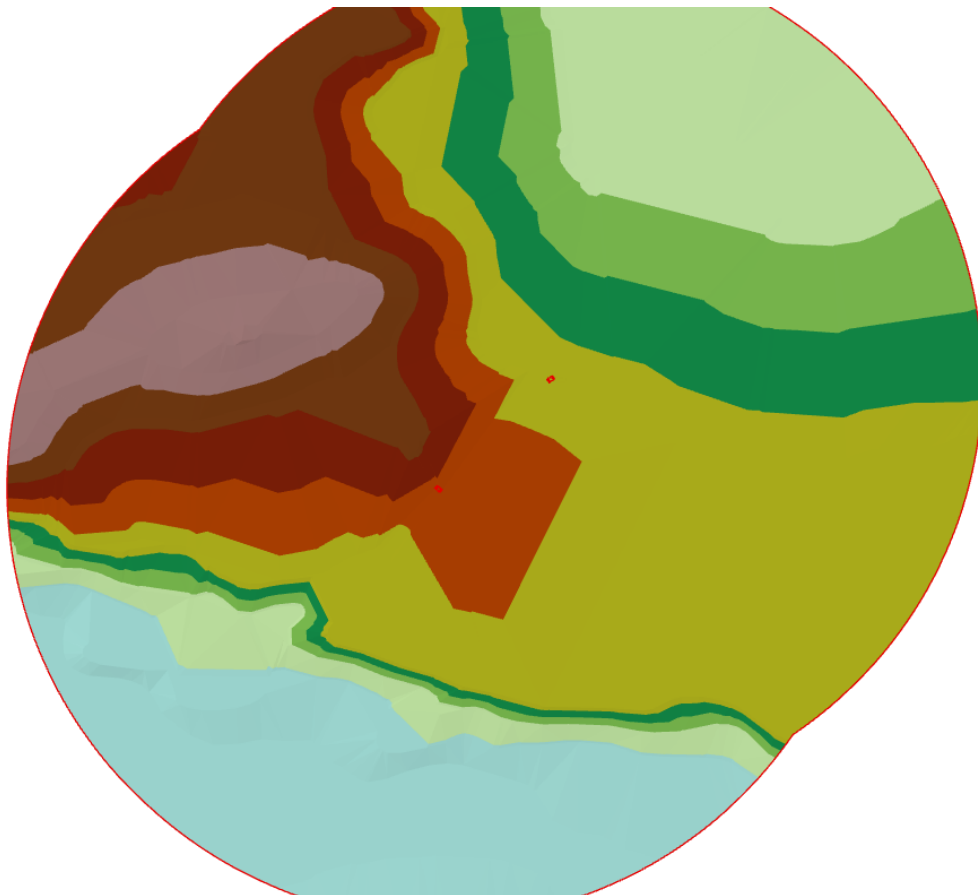


Obrázek 25 Vizualizace 2. mostu

V programu ArcGIS Pro lze také vizualizovat samotný surface TIN (obr. 26 a 27). Na prvním obrázku lze vidět konstantní výšku kolem půdorysu mostu a na druhém obrázku je vidět celý surface TIN oříznutý pro lepší práci s daty. Je zde vidět, že je pracováno s oběma mosty najednou.



Obrázek 26 Konstantní výška surface TINu



Obrázek 27 Surface TIN pracující se 2 modely

8 Diskuse

V předchozí kapitole jsou ukázány některé modely vytvořené pomocí rozšíření pro Blender, který je v této práci popsán. Rozšíření je uzpůsobeno tak, aby bylo schopné generovat model podle počtu pruhů, který uživatel zadá. Následně může být model posunut a otočen pro práci v souřadnicovém systému a nakonec je jej možné exportovat ve formátu *.dae*. Tento formát je zvolen proto, aby byl model kompatibilní s ArcGIS Pro a také, aby si zachoval textury.

Parametry pro generování mostů byly odečítány z konkrétního mostu. Jeho rozměry byly získány za pomoci metody SfM, kde byla vytvořena kostra modelu. Původně bylo zamýšleno větší využití této metody, ale kvůli špatně přístupnému terénu a podmínkám v okolí mostu nebyl výsledný model příliš kvalitní a posloužil jen k odečtení rozměrů. Některé rozměry, jako například rozestupy mezi jednotlivými směry, nešly odečíst ani z kostry a musely být odečteny z leteckých snímků. Z modelu nešly použít ani textury, ty se ale daly nahradit ze snímků, které byly pořízeny za účelem metody SfM.

Když se zkombinovala data o počtu jízdních pruhů z dat z práce Jakuba Růžičky (2020) a z dat pořízených v této práci, byl získán přibližný obraz toho, jak by měl most při daném počtu pruhů vypadat. Nevýhoda je, že takto získané informace sedí převážně jen na most, se kterým bylo pracováno, ale rozšíření je myšleno pro generování obecně. Z toho vyplývá, že když se bude generovat nějaký jiný most, bude jeho model odpovídat skutečnosti výrazně méně. U modelu se dá nastavit úhel křížení podjezdu a nadjezdu, aby případné jiné modely seděly co nejlépe na jízdní pruhy a komunikace.

Při generování jiných modelů je také možné vyměnit textury, které jsou k rozšíření přiloženy. Tyto textury je totiž před vytvořením modelu potřeba nahrát ze souboru. Toto ale umožňuje uživateli si vytvořit a nahrát své vlastní textury, které by mu více vyhovovaly, ale geometrie konstrukce zůstane stejná. Toto s sebou nese i nevýhodu, že soubor se texturami není uvnitř rozšíření a musí být při každém spuštění nahrána cesta k tomuto adresáři.

Na zkušební modelech, které jsou ukázány v kapitole 7 s výsledky, lze pozorovat, že generátor má problém s tím, že pruhy nejsou vždy stejně široké, nebo nejsou rovné. V prvním případě (obr. 23) je problém s pruhy v podjezdu. V reálném světě totiž přesně v podjezdu dochází k odbočení nového pruhu. To je u modelu zanedbáno, aby rozšířené mohlo pracovat s obecnými mosty. Kvůli tomu lze vidět, že nenavazují přesně na mostní

konstrukci. U druhého modelu (obr. 25) je zase problém s tím, že pruh nepatrně zatáčí, ale mostní konstrukce je zjednodušená a je rovná, kvůli tomu také vzniká nepřesnost při navázání pruhů na konstrukci. Tyto nevýhody jsou však daní za možnost využívat rozšíření obecně na jiné mostní konstrukce.

Vzhledem k tomu, že rozšíření pro Blender vytváří model po částech na základě gramatiky dělení, je možné v případě potřeby přidat další detailní prvky, či jiné části mostní konstrukce. Přidat například nájezdy, či jinak udělat plynulý přechod mezi jízdnicí a modelem mostu.

Druhý skript, který zajišťuje přípravu dat pro vizualizaci, je ošetřen, kdyby pro vytvoření surface TINu nebyla vhodná data. Data musí splňovat stejné podmínky, které jsou popsány v kapitole 6.3.2. Výsledný surface TIN je potřeba k správné vizualizaci dat. Jak již bylo řečeno, bez tohoto TINu by byl model zanořený do zemského povrchu a vizualizace by nebyla dobrá.

Generátor mostu by se dal rozšířit v případě potřeby o další skript, který by pracoval s daty o jízdnicích pruzích. Bylo by možné automaticky odečíst z těchto dat souřadnice středu mostu, jeho orientaci, počet pruhů a vrátit tyto informace například jako textový soubor, ze kterého by poté mohly být přepsány do uživatelského rozhraní rozšíření. Úplná automatizace by byla asi komplikovanější, protože každá část pracuje s jinými knihovny, ke kterým nemají programy vzájemně přístup.

9 Závěr

Cílem práce bylo navrhnout a implementovat postup 3D vizualizace komplexních prvků silniční infrastruktury se zaměřením na mimoúrovňová křížení. Hlavní myšlenka byla taková, aby navržené řešení bylo možné použít obecně pro jakoukoli mimoúrovňovou křižovatku. Rešerše by se dala rozdělit na tři části. První část se zaměřuje na principy procedurálního modelování, z jehož základů poté vychází výsledný generátor mostů. Druhá část se zaměřuje na sběr dat. Nejprve popisuje metodu SfM jakožto možnost pořizovat celé modely rovnou z fotografií a poté se zaměřuje na získání informací o geometrii silnic z leteckých snímků. Zde se zaměřuje především na práci Jakuba Růžičky (2020), protože data, která byla vytvořena v jeho práci, slouží jako základní informace pro generování modelů v této práci. Třetí část popisuje použitý software.

Metodická část popisuje hlavní principy, na kterých stojí program generátoru mostů. Je zde také popsáno jeho ovládání. Poté je zde ukázán proces úpravy dat a příprava na jejich vizualizaci v geoinformačním systému ArcGIS Pro.

I přes některé nevyhnutelné kompromisy byl cíl práce splněn. Popsaný postup a na jeho základě vytvořený add-on pro Blender je schopen vytvořit obecný 3D model mimoúrovňové křižovatky.

Seznam použitých zdrojů

BLENDER FOUNDATION (2021): Blender 2.93 Reference Manual, blender.org [online], cit. 16. 7. 2021, dostupné z: <https://docs.blender.org/manual/en/latest/index.html>

ESRI (2021): Scenes, arcgis.com [online], cit. 17. 3. 2021, dostupné z: <https://pro.arcgis.com/en/pro-app/latest/help/mapping/map-authoring/scenes.htm>

ESRI (2021): Multipatches, arcgis.com [online], cit. 17. 3. 2021, dostupné z: <https://desktop.arcgis.com/en/arcmap/latest/extensions/3d-analyst/multipatches.htm>

FRANTÁL, P. (2018): Procedurální modelování měst vyvíjejících se v čase. Praha, Bakalářská práce. České vysoké učení technické v Praze.

HINZ, S., BAUMGARTNER, A. (2003): Automatic extraction of urban road networks from multi-view aerial imagery. ISPRS Journal of Photogrammetry & Remote Sensing, 58, 83-98.

HORMESE, J., SARAVANAN, C. (2016): Automated Road Extraction From High Resolution Satellite Images. International Conference on Emerging Trends in Engineering, Science and Technology (ICETETS) 2015. In: Procedia Technology, 24, 1460– 1467.

HU, F., ZHAO, J., HUANG, Y., LI, H. (2020): Structure-aware 3D reconstruction for cable-stayed bridges: A learning-based method, COMPUTER-AIDED Civil and Infrastructure ENGINEERING, 36, 89-108.

HUYNH, L. (2018): Structure-from-motion using convolutional neural networks. Oulu, Diplomová práce. University of Oulu.

KELLY, G., MCCABE, H. (2006): A survey of procedural techniques for city generation. ITB Journal, 14: 87-130.

MELICHAR, B. (2003): Jazyky a překlady, ČVUT, Praha.

PRUSINKIEWICZ, P., LINDENMAYER A. (1990): The algorithmic beauty of plants, Plant Science, 122, 109-110.

PRUSINKIEWICZ, P. (1986): Graphical Applications of L-Systems, Graphics Interface, 247-253.

RŮŽIČKA, J. (2020): Automatic Detection of Driving Lanes Geometry Based on Aerial Images and Existing Spatial Data, Praha, Diplomová práce, Univerzita Karlova.

SEO, W.-Y. (2012): Augmenting Cartographic Resources and Assessing Roadway State for Vehicle Navigation. Pittsburgh, Disertační práce, The Robotics Institute, Carnegie Mellon University.

SHAHI, K., SHAFRI, H. Z. M., TAHERZADEH, E., MANSOR, S., MUNIANDY, R. (2015): A novel spectral index to automatically extract road networks from WorldView-2 satellite imagery. The Egyptian Journal of Remote Sensing and Space Sciences, 18, 27–33.

SMELIK, R. M., TUTENEL, T., BIDARRA, R., BENES, B. (2014): A survey on procedural modelling for virtual worlds. Computer Graphics Forum 33(6), 31–50.

SNAVELY, N. SEITZ, S., SZELISKI, R (2006): SIGGRAPH '06: ACM SIGGRAPH 2006 Papers July 2006, 835–846

STINY, G. (1975): Pictorial and Formal Aspects of Shape and Shape Grammars: On Computer Generation of Aesthetic Objects [online]. ISBN 978-3-7643-0803-2. Dostupné z: doi:10.1007/978-3-0348-6879-2

ULLMAN, S. (1979): The interpretation of structure from motion, Proceedings of the Royal Society of London. Series B, Biological Sciences, 203, 405-426

WESTOBY, M.J., BRASINGTON, J., GLASSER, N.F., HAMBREY, M.J, REYNOLDS, J.M (2012): Structure-from-Motion photogrammetry: A low-cost, effective tool for geoscience applications. Geomorphology, 179, 300–314.

WONKA, P., WIMMER, M., SILLION, RIBARSKY, W. (2003): Instant architecture. ACM Transactions on Graphics 22(3), 669-677.