

Grafická aplikace umožňující interaktivní prohlížení prostorových rastrových snímků

Jiří Wolker, 4.TL

školní rok 2021/2022

Prohlášení

Prohlašuji, že jsem svou práci praktické zkoušky vypracoval samostatně a použil jsem prameny a literaturu v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Havlíčkově Brodě
dne 13. 4. 2022

Jiří Wolker

Anotace

Tato práce se zabývá aplikací, která umožňuje interaktivní prohlížení objemových dat s použitím grafického uživatelského rozhraní, a také různými metodami vizualizace těchto dat. Objemová data, která v práci používám jsou pořizována tzv. light-sheet mikroskopem.

Vykreslování je prováděno s pomocí hardwarové akcelerace grafickou kartou počítače za použití rozhraní OpenGL. Je zároveň umožněna vizualizace objemových binárních dat, a to buď samostatně nebo souběžně se snímkem. Aplikace umožňuje procházení časových sekvencí a změnu nastavení pohledu jak pomocí klávesnice, tak i myši.

V rámci vývoje aplikace byly prozkoumány další dostupné nástroje a byla vytvořena knihovna pro výpočet transformačních matic pro transformaci vektorů v prostoru.

Annotation

This work describes an application for interactive preview of volume data with use of graphical user interface, as well as various methods used to visualize such data. The data used in this work have been acquired using so called light-sheet microscope.

The application uses hardware-accelerated rendering provided by a graphical processing unit with an OpenGL interface. In addition to the volumetric data rendering, it is possible to visualize volumetric binary data. That can be performed either in combination with rendering of volumetric data or in a stand-alone fashion. The application provides facilities for viewing time-ordered sequences of volumetric data and keyboard- and mouse-based interface to adjust view parameters.

Along with development of the application, research of other software tools has been performed and a library of routines for transformation matrix computation has been created.

Poděkování

Za vedení mé maturitní práce, jeho čas a cenné rady a připomínky bych chtěl touto cestou poděkovat Dr. Ing. Janu Schierovi. Taktéž bych chtěl poděkovat Ústavu teorie informace a automatizace (ÚTIA) AV ČR za to, že mi umožnil v rámci studentské stáže se účastnit práce na vývoji nástrojů pro zpracovávání mikroskopických dat. Ústavu živočišné fyziologie a genetiky AV ČR bych chtěl poděkovat za poskytnutí snímků oocytů ze SPIM (*single-plane illumination microscope*), které můžete vidět v této práci.

V neposlední řadě bych chtěl poděkovat své matce za trpělivost a podporu nejenom během školních prací.

Obsah

Prohlášení	iii
Anotace v českém a anglickém jazyce	v
Poděkování	vii
Úvod	x
1 Formát dat	1
1.1 TIFF	2
1.1.1 TIFF s rozšířeními IMAGEJ	2
1.1.2 OME-TIFF	3
1.2 HDF5	4
2 Prostorová data v paměti	5
2.1 Reprezentace prostorových snímků polem	5
2.2 Optimalizace práce se snímky s virt. pamětí	6
3 Zobrazování 3D snímků	9
3.1 Řezy snímkem	9
3.1.1 Řez kolmý na jednu ze souřadných os	9
4 Stávající řešení	11
4.1 Nástroje dostupné v rámci aplikace IMAGEJ	11
4.1.1 IMAGEJ – prohlížení 2D snímku	11
4.1.2 IMAGEJ – nástroj <i>Z-project</i>	12
4.1.3 IMAGEJ – nástroj <i>3D Project</i>	15
4.1.4 IMAGEJ – nástroj BIGDATAVIEWER	15
4.1.5 IMAGEJ – nástroj <i>3D Surface Plot</i>	17
5 Aplikace HYPERSTACKVIEW	19
5.1 Snímky obrazovky	19
5.2 Použité knihovny	22
5.2.1 PySide (Qt)	22
5.2.2 OpenGL	22
5.2.3 NumPy	22
5.2.4 Knihovna pro vytváření transformačních matic	22
5.2.5 Načítání souborů TIFF – knihovna <i>tiff file</i>	23
5.3 Vykreslování snímků	23

5.3.1	Projekce snímku	24
5.3.2	Projekce maximální intenzity	25
5.3.3	Projekce průměrné intenzity	25
5.3.4	Zpracovávání snímku s objekty	26
5.3.5	Vykreslování projekce	26
5.3.6	Projekce s transformací	30
5.4	Překryvné grafické prvky	31
5.4.1	Náhled rotace snímku	31
5.4.2	Překryvná mřížka	31
5.4.3	Stínování protějších stěn snímku	31
5.5	Filtrování vstupních dat	31
5.5.1	Použití v kombinaci s jinou aplikací	32
5.6	Uživatelské rozhraní aplikace	32
5.6.1	Hlavní okno	32
5.6.2	Ovládání zobrazení myší	34
5.6.3	Klávesové zkratky	36
6	Knihovna pro transformace	39
6.1	Transformační matice ve 3D grafice	39
6.2	Podporované transformace	39
6.2.1	Posunutí	40
6.2.2	Škálování	40
6.2.3	Rotace kolem osy určené vektorem	41
6.2.4	Prohození souřadných os	42
6.2.5	Odstranění souřadné osy	42
6.2.6	Převod do prostoru textur a zpět	43
6.2.7	Perspektiva s jedním úběžníkem	44
6.3	Funkcionální rozhraní	44
6.4	Objektové rozhraní	47
	Závěr	49
	 Seznam obrázků a seznam tabulek	50
	 Bibliografie	52
	 Seznam zkratek	54

Úvod

Některé moderní mikroskopické metody používané v biologii, jako je například konfokální mikroskopie nebo *Single-plane illumination microscope*, nazývaná zkráceně také SPIM, poskytují takzvaná objemová data. To jsou třírozměrná, v určitých případech i vícerozměrná, pole číselných hodnot. Ta zpravidla odpovídají kvádru fyzického prostoru pokrývajícimu zobrazovaný vzorek, přičemž je typicky zaznamenávána fluorescenční odezva vzorku v tomto prostoru.

Objemová data jsou získávána i při jiných měřeních, kterými jsou, kromě jiných, CT (tj. výpočetní tomografie) nebo MRI (tj. magnetická rezonance).

Většina metod získávání objemových dat je založena na zachycování sekvence dvojrozměrných řezů různými částmi zkoumaného objektu. Ty mohou být poté zpracovávány jako třírozměrný popis vlastností analyzovaného vzorku.

Vzhledem k tomu, že tato data mají více než dva rozměry, není možné je přímo vykreslovat na obrazovku počítače, kde mohou být analyzovány člověkem. Je proto nutné vytvořit dvojrozměrnou podobu těchto snímků – vizualizovat je. Nejjednodušším způsobem vykreslení (anglicky *rendering*) je třírozměrný snímek rozložit na jeho jednotlivé roviny. Tento způsob používá například volba *slidemontage* v systému IMAGEJ, která umožňuje zobrazit tabulku (montáž) jednotlivých snímků nebo procházení snímku pomocí posuvníku. Takový přístup je v praxi používán například v lékařství.

Provedení řezu libovolnou rovinou je sice výpočetně náročnější, ale v případě některých druhů dat je nutné pro interpretaci takových dat. Pokud je nutné zobrazit celý snímek najednou, provádí se vykreslení objemových dat, tedy *volume rendering*, kdy je ve výstupním snímku zobrazen celý objem snímku. Další metody zobrazení, jako je například *marching cubes*, jsou založeny na tom, že jsou data analyzována a je vykreslen výsledek této analýzy.

Objemové snímky mohou být zaznamenávány také periodicky po určitém čase a tvoří tak časovou sekvenci. Taková sekvence je označovaná také jako *3D+t data*.

Nástroj HYPERSTACKVIEW, tedy aplikace umožňující zobrazování 3D+t dat¹, získávaných zejména za pomoci pokročilých mikroskopických metod, v reálném čase za pomoci hardwarové akcelerace grafickou kartou, byl vyvinut za účelem usnadnění vývoje systému pro automatizované zpracovávání mikroskopických dat.

¹Jedná se o časové sekvence 3-rozměrných bitmapových snímků. Formát dat je blíže popsán v části 1 (strana č. 1).

Tato aplikace byla původně vytvořena k tomu, aby nahradila stávající řešení za pomoci softwaru IMAGEJ, které vzhledem k použité metodě vykreslování snímků neposkytovalo dostatečně rychlé vykreslování pro snadné prohlížení obrazových dat. Toto řešení je popsáno na straně č. 15.

Postupným vývojem byly přidávány další funkce, například: vykreslování objektu nalezeném za pomoci segmentace (strana č. 29), vybírání objektů podle přiděleného identifikátoru (strana č. 31) nebo vykreslení 3D projekce v perspektivě (strana č. 30).

Pro vytvoření této aplikace byl použit programovací jazyk Python. Grafické uživatelské rozhraní bylo vytvořeno pomocí knihovny PySide (dříve nazýváno PyQt) a samotné vykreslování snímku je provedeno za pomoci rozhraní OpenGL, které umožňuje běh programu pro vykreslování na GPU.

1 Formát dat

Aplikace HYPERSTACKVIEW byla původně vytvořena za účelem vykreslování časových sekvencí snímků ze SPIM¹, což je fluorescenční mikroskop, který pořizuje trojrozměrné snímky, na kterých je zaznamenáno množství světla emitovaného v každém bodě snímaného prostoru. [5] To znamená, že vzorek světlo aktivně vyzařuje. Snímky, které jsou používány, mohou být uloženy jako soubor ve formátu IMAGEJ TIFF (strana č. 2), ve kterém jsou uloženy 16bitové hodnoty jednotlivých *voxelů*² (jednotlivých bodů snímku).

Aplikací HYPERSTACKVIEW jsou podporovány různé varianty formátu TIFF, zejména OME TIFF (strana č. 3), který je spolu s formátem TIFF s rozšířeními IMAGEJ používán v mikroskopii. To je umožněno tím, že byla použita knihovna *tiff* (strana č. 23), která je určena pro čtení a zapisování mnoha v praxi používaných variant formátu TIFF, včetně těch, které jsou používány pro ukládání tří- a vícerozměrných dat.

Ačkoliv je aplikace vytvořena se záměrem zobrazování snímků s šestnáctibitovou hloubkou, je možné použít i soubory s jinými běžně používanými hloubkami, a to zejména 8, 32 bitů s celočíselnými hodnotami, nebo čísla s plovoucí řádovou čárkou dle IEEE 754 s jednoduchou nebo dvojitou přesností. Před prováděním výpočtů jsou snímky převedeny do interního formátu, tedy čísel s plovoucí řádovou čárkou.

Souběžně s fotometrickým snímkem z mikroskopu je možné zobrazovat další data – informace o tom, zda je v daném bodě přítomný zkoumaný objekt. případně který objekt. Požadovaným formátem pro tato data je opět soubor v jedné z podporovaných variant TIFF o stejných rozměrech. V jednotlivých bodech v prostoru tvaru kvádrů popisovaného tímto souborem jsou uloženy číselné hodnoty, kde 0 označuje nepřítomnost objektu a číslo větší než 0 přítomnost objektu. Je-li třeba rozlišit mezi více objekty na jednom snímku, toto číslo je současně i jedinečným identifikátorem tohoto objektu. Tyto

¹ *Single-plane illumination microscope*

² Výrazem *voxel* je označována obdoba pixelu v prostoru. Voxel je obrazový bod třírozměrného rastrového snímku určený třemi souřadnicemi x , y a z a je mu přidělena jedna hodnota. V případě voxelů tzv. vícekanálového snímku je voxelu přidělena pro každý kanál (tj. například určitá vlnová délka zachytávaného světla) jedna hodnota. V jiných kontextech může být jako *voxel* označován i prvek o jednotkové velikosti ve 3D scéně. Slovo *voxel* v tomto smyslu není v této práci použito.

informace jsou použity pro vykreslování objektů souběžně se snímkem (strana č. 29) a vybírání částí snímku ke zobrazení, popřípadě jejich zvýraznění (strana č. 31).

Snímky mohou být pořizovány jako časová sekvence, to znamená, že jsou pořizovány v pravidelných časových intervalech. Sekvence snímků může být uložena do souboru jako 4rozměrné pole čísel, jehož rozměry jsou *čas*, *hloubka*, *výška*, *délka* namísto *hloubka*, *výška*, *délka*. Zobrazování těchto souborů bylo také umožněno. Pro snížení nároků na paměť počítače je v jednu chvíli načten právě zobrazený snímek a případně ostatní snímky, které byly v nedávné době zobrazeny. V případě nedostatku fyzické paměti je tato paměť uvolněna.

Vzhledem k charakteru dat je umožněno přehrávání časové sekvence jako animace. To usnadňuje uživateli prohlížení dlouhých sekvencí snímků.

1.1 TIFF

Tří- a vícerozměrné rastrové snímky jsou často ukládány do souborů TIFF. TIFF je formát pro ukládání obrazových dat s přidruženými metadaty. Možnost rozšířit formát TIFF umožnila vznik několika variant tohoto formátu.

Prohlížeč HYPERSTACKVIEW je určen k použití především s těmito variantami: IMAGEJ TIFF a OME-TIFF.

Obě níže popisované varianty formátu TIFF jsou určeny k zaznamenávání dvou- až pětirozměrných snímků. To, jak jsou jednotlivé rozměry označovány je popsáno na straně č. 3.

1.1.1 TIFF s rozšířeními IMAGEJ

Tato varianta formátu TIFF byla vytvořena jako formát pro ukládání obrazových dat o 2–5 rozměrech (tzv. *hyperstack*). Je používán aplikací IMAGEJ,[10] která je určena pro zobrazování, analýzu a zpracovávání mikroskopických a jim podobných obrazových dat o 2–5 rozměrech (tj. dvoj- nebo trojrozměrný snímek o jednom či více kanálech, popřípadě jejich časová sekvence). I přes to, že oficiální dokumentace tohoto rozšíření TIFF neexistuje,³ jedná se o poměrně rozšířený formát pro ukládání obrazových dat. Tento formát souboru umožňuje ukládání obrazových dat, kde jsou jednotlivé body uloženy jako celé číslo nebo číslo s plovoucí řádovou čárkou. Je možné uložit dvojrozměrný obrázek (s osami x , y) a je možné navíc přidat další rozměry obrázku:

³IMAGEJ je software s veřejně dostupným zdrojovým kódem, takže je možné při vývoji jiného software pracujícího s tímto formátem souborů vycházet z implementace používané aplikací IMAGEJ.

- z – tedy třetí prostorový rozměr obrázku,
- c – tedy *kanály* snímku (je možné použít libovolný počet kanálů),
- t – tedy časový rozměr, který je používán pro reprezentaci časových sekvencí.

V terminologii používané aplikací IMAGEJ je snímek se třemi rozměrovými osami nazýván *stack* a časová sekvence takových snímků *hyperstack*. Jednotlivé roviny snímku jsou potom nazývány *slice*. Na obrázku 1.1 (strana č. 4) můžete vidět grafické znázornění těchto možností uložení snímku.

Kromě samotných obrazových dat jsou ukládána metadata, tedy informace o snímku, jako je například reálná velikost zachyceného prostoru, ze které je možné vypočítat reálnou velikost pixelů, resp. voxelů.

Data v souboru jsou uložena jako pole číselných hodnot, které odpovídají jednotlivým obrazovým bodům. Jednotlivé části snímku, jako jsou například snímky časové sekvence nebo jednotlivé roviny trojrozměrného snímku nejsou zarovnávány na určitou pozici v souboru.

1.1.2 OME-TIFF

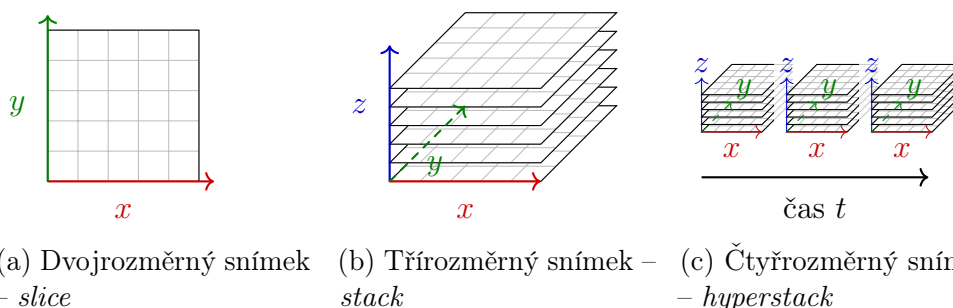
Formát souboru OME-TIFF[3] (*Open Microscopy Environment TIFF*) je variantou formátu TIFF, která je specializovaná na zaznamenávání a přenos obrazových dat získaných za pomoci mikroskopu. K samotným obrazovým datům jsou přidána metadata v textovém formátu, který vychází z formátu OME-XML.[2]

Struktura snímku ve formátu OME-TIFF

Tento formát ukládá data obdobným způsobem jako předchozí jmenovaný formát. Na rozdíl od formátu aplikace IMAGEJ je poskytnuta lepší dokumentace.

Více jednotlivých stacků nebo více dvojrozměrných snímků je možné sdružit do časové posloupnosti, což přidá další rozměr snímku, jenž je označován t . Časová posloupnost stacků je nazývána *hyperstack* (viz obr. 1.1c).[4]

Každý snímek může mít jeden nebo více kanálů. Vícekanálové snímky nachází použití zejména při fluorescenční mikroskopii, kde je toho využíváno tak, že zkoumané struktury buňky jsou obarveny fluorofory, které emitují světlo o různých vlnových délkách. Každá vlnová délka světla je zachycovaná samostatně a každé zachycené vlnové délce odpovídá kanál snímku. *Hyperstack*, který obsahuje obrazová data s více kanály je tedy pětirozměrný.



Obrázek 1.1: Grafické znázornění možností zaznamenání snímků s různým počtem rozměrů.

Ke snímku mohou být přidružena další metadata, stejně jako v případě použití OME-XML, například: informace o použitém mikroskopu (výrobce, model, typ), informace o použitém zdroji světla, o použitém detektoru, dalších optických součástech mikroskopu, informace o experimentu a podobně.[4]

1.2 HDF5

HDF5 je datový formát pro ukládání větších, strukturovaných, heterogenních datových souborů. Jedním z jeho použití je i uchovávání mikroskopických snímků.[12]

Tento formát je zde uveden pouze pro úplnost a není v níže popisované aplikaci používán, ale vzhledem k modulárnímu provedení načítání dat je možné podporu pro tento formát přidat.

2 Reprezentace prostorových dat v paměti

Zatímco kapitola 1 popisovala podobu prostorových dat používanou pro uchovávání snímků na diskových a jim podobných úložištích, v následující sekci je popsána podoba prostorových dat v RAM (tj. operační paměť počítače), jejímž účelem je zpravidla možnost efektivního přístupu k zaznamenaným datům. V sekci 2.1 jsou zmíněny vybrané metody pro urychlení zpracovávání dat v operační paměti a možnosti zmenšení využití paměti.

V současné době naprostá většina počítačových systémů používaných jako desktop nebo server používá *virtuální paměť* s tzv. stránkováním, kde každé běžící aplikaci je paměť rozdělena do určitých úseků, *stránek*, které jsou samostatně aplikaci přidělovány.[1] Každé stránce je určeno zejména to, které části fyzické paměti počítače odpovídá a to, zda je stránka dostupná, tj. nachází se ve fyzické paměti, nebo není. Ve druhém případě je při přístupu do paměti vyvoláno přerušení, jehož cílem může být doplnění dat do fyzické paměti přečtením z diskového úložiště nebo detekce chybných a zakázaných paměťových operací. Tato technika umožňuje realizovat například *stránkování anonymní paměti*, které nachází použití v situacích, kdy je třeba zajistit dostatek volné fyzické paměti přesunutím dat z ní do jiného úložiště, nebo *mapování souborů do paměti*, které je využíváno právě v aplikaci HYPERSTACKVIEW pro zefektivnění čtení vstupních dat. To, jak je možné zefektivnit práci s prostorovými snímky je popsáno v sekci číslo 2.1.

2.1 Reprezentace prostorových snímků polem

S objemovými snímky a jejich časovými sekvencemi je možné v paměti pracovat jako s vícerozměrným polem o rozměrech $t \times z \times y \times x \times c$, kde t je počet snímků v časové sekvenci (v případě samostatného snímku 1), z , y a x jsou rozměry v prostorových osách a c je počet kanálů snímku. V praxi se používají i jiná pořadí – například $t \times c \times z \times y \times x$. V takovém případě se data v jednom kanálu nachází v paměti blíže u sebe.

Vzhledem k tomu, že tato data mají většinou značnou velikost, může být z úložiště načítána pouze část snímku, se kterou se pracuje v RAM. To může být provedeno explicitně v aplikaci, která data zpracovává, nebo může

být využito mapování souboru do paměti (tento přístup je použit v aplikaci popisované v této práci).

2.2 Použití virtuální paměti pro optimalizaci reprezentace prostorových snímků v paměti

Soubory prostorových snímků, stejně jako jakékoliv jiné soubory uložené na diskovém úložišti je možné takzvaně mapovat do paměti. V případě, že jsou data v souboru uložena způsobem, který je vhodný i pro práci s daty v operační paměti počítače, můžeme takto mapovaný soubor, respektive jeho část s daty, používat stejným způsobem, jako by data byla načtena do paměti obvyklým způsobem.

Pokud není způsob uložení dat vhodný pro přímé použití, je nutné soubor číst obvyklým způsobem nebo soubor před použitím převést do jiného formátu. To může být nutné v těchto situacích:

- data v souboru jsou komprimována;
- endianita dat v souboru se liší od endianity systému, na kterém je soubor čten;
- datový typ, který je použitý pro uložení jednotlivých hodnot ve snímku, není vhodný pro zpracovávání, a to z jakéhokoliv důvodu (například pro některé výpočty prováděné s daty nemusí být celočíselné hodnoty vhodné, nebo naopak je vyžadována práce s celočíselnými hodnotami, např. z důvodu rychlosti zpracovávání);
- jednotlivé prostorové či časové osy snímku nejsou ve vhodném pořadí, případně nejsou kanály uspořádány vhodným způsobem.

To, zda je některou z výše jmenovaných situací nutné řešit, se může lišit podle toho, jak jsou data zpracovávána.

Níže jsou popsány důvody toho, proč není možné mapování do paměti v uvedených situacích použít, a také to, jak dané situace je řešit možné.

Čtení komprimovaných dat Je-li soubor komprimován jako celek, je zpravidla nemožné se souborem pracovat tak, aby bylo známe, v jakém místě se který úsek nachází. Pokud jsou komprimovány jednotlivé části souboru (například snímky v časové sekvenci nebo roviny ve snímku) samostatně, je potom možné k nim přistupovat i při použití mapování do paměti. Pokud má použitý kompresní algoritmus konstantní kompresní poměr (tj. stejně velké snímky před kompresí jsou stejně velké i po kompresi), je možné použít

přímočarou metodu přístupu k datům, stejně jako u nekomprimovaných dat. Pokud kompresní poměr není konstantní, je nutné před přístupem k datům zjistit místo v souboru pomocí překladové tabulky, která musí být vytvořena předem.

Odlišná endianita dat Pokud se endianita (pořadí bajtů v celém čísle uloženém v paměti, resp. souboru) souboru liší od endianity systému, na kterém je soubor mapovaný do paměti používán, je nutné při čtení i zápisu do takového souboru převádět endianitu číselných hodnot. Pokud se k datům přistupuje velmi často, může být vhodnější nejprve celý soubor převést do endianity systému.

Odlišný datový typ číselných hodnot V této situaci platí podobná omezení těm, která se týkají práci se souborem s odlišnou endianitou dat.

Nevhodné pořadí os snímku Tato situace sice neznemožňuje mapování snímku do paměti, ale musí se řešit to, že v určitých situacích je vhodné určité pořadí os snímku a je vhodné tomu přizpůsobit i samotný program pracující se snímkem. Například, pokud je pořadí os ve snímku XYZ (tj. adresy hodnot v paměti odpovídajících bodům, které se liší pouze souřadnicí Z, mají nejmenší rozdíl, zatímco adresy hodnot odpovídající bodům s rozdílnou souřadnicí X mají tento rozdíl největší) a má být provedeno vykreslení řezu snímkem v rovině X–Y, dojde k přečtení všech stránek paměti, ve kterých se snímek nachází, takže dojde ke kompletnímu přečtení souboru. Naopak v případě, že se pracuje se souborem s pořadím os ZYX, k přečtení celého souboru dojít nemusí, a tak dojde k provedení zpracování snímku v menším časovém úseku.

3 Zobrazování prostorových (3D) snímků

Jak již bylo uvedeno v úvodu, metody zobrazování třírozměrných snímků můžeme rozdělit do tří kategorií:

- vykreslení řezu snímkem;
- vykreslení projekce snímku přímo, bez jakéhokoliv předchozího zpracování – tzv. *direct rendering*;
- zpracování snímku a vykreslení získaných dat – tzv. *indirect rendering*.

Aplikace HYPERSTACKVIEW, která je v této práci popisována dovoluje provádět první dvě z těchto tří vykreslení snímku.

V následujících částech se nachází přehled běžně používaných metod zobrazování 3D snímků.

3.1 Vykreslování řezů prostorovými daty

Zobrazení řezu snímkem je oproti zobrazování snímku v projekci maximální intenzity jednodušší operací. Můžeme ho provádět jako řez libovolnou rovinou nebo řez kolmý na jednu ze souřadných os.

3.1.1 Řez kolmý na jednu ze souřadných os

Vykreslení tohoto řezu je výpočetně nenáročné, pokud je snímek v paměti uchováván jako pole. V tomto případě postačuje pouze vykreslení části pole nebo procházení pole tak, že se mění pouze pozice na dvou osách.

Předpokládejme, že je snímek uložen jako pole o rozměrech $z \times y \times x$ (tzn. na nejnižší adrese v paměti se nachází část, kde $z = 0$, následuje část s $z = 1$ atd. a pixely, jejichž poloha se liší pouze v x -ové souřadnici, a to o 1, se v paměti nacházejí vedle sebe). Do tohoto pole můžeme přistupovat jako do jednorozměrného pole (potom jsou pixely přístupné v pořadí popsáném výše). Pro vykreslení řezu snímkem kolmo na osu z rovinou, jejíž z -ová souřadnice

má hodnotu z_n , postačí interpretovat úsek dlouhý $x \cdot y$ pixelů začínající na pozici $z_n \cdot x \cdot y$ jako obrázek o rozměrech $y \times x$.

Pro vykreslení řezu v kolmo na jinou osu je potřeba procházet prakticky celý rozsah pole; pokud vykreslujeme například řez kolmý k ose y na pozici y_n , musíme projít všechny pixely se souřadnicemi x_p, y_n, z_p , kde $x_p \in \langle 0; x \rangle$ a $z_p \in \langle 0; z \rangle$ jsou souřadnice ve výsledném snímku.

4 Stávající řešení

Aplikace HYPERSTACKVIEW vznikla z toho důvodu, že nebyla dostupná aplikace, která by umožňovala snadno zobrazovat tří- a vícerozměrné mikroskopické snímky. Při vzniku se vycházelo z principů používaných rozšířeními aplikace IMAGEJ pro vykreslení projekce 3D snímku (strana č. 15).

4.1 Nástroje dostupné v rámci aplikace IMAGEJ

Aplikace IMAGEJ poskytuje mnoho základních nástrojů pro projekci 3rozměrných snímků (v terminologii IMAGEJ *stack*) a vícerozměrných snímků (v terminologii IMAGEJ *hyperstack*).

Většina těchto nástrojů má ekvivalent i v jiných aplikacích určených pro zpracovávání a zobrazování více než dvojrozměrných rastrových snímků, a proto nebudou ekvivalenty funkcí IMAGEJ v jiných aplikacích popisovány.

Sekce popisující jednotlivé nástroje dostupné v IMAGEJ jsou pojmenovávány podle názvu akce v nabídce aplikace, která popisovaný nástroj spouští. U některých nástrojů je v různých částech uživatelského rozhraní tentýž nástroj označován odlišnými názvy.

4.1.1 IMAGEJ – prohlížení 2D snímků a řezů 3D snímkem

Jednou ze základních funkcí aplikace IMAGEJ je zobrazení 2D snímku, kde je jednotlivým pixelům ve snímku přiřazena barevná hodnota pomocí barevné palety (LUT, *lookup table*) nebo jas bodu odpovídá číselné hodnotě v daném bodě. Je možné měnit měřítko zobrazeného snímku. V případě vykreslování vícekanálového snímku je možné mezi zobrazeními jednotlivých kanálů přepínat, nebo je zobrazit současně tak, že je každý vykreslen odlišnou barvou.

Pomocí tohoto zobrazení je možné vykreslit i vícerozměrné snímky. V případě vykreslování vícerozměrného snímku je vždy zobrazen právě jeden *slice* ze snímku určený číselně nebo zadáním pomocí posuvníků.

4.1.2 IMAGEJ – nástroj *Z-project*

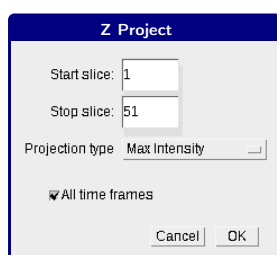
Nástroj *Z-project* v aplikaci IMAGEJ umožňuje provést projekci všech rovin snímku do jedné. To znamená, že je rozměr z snímku snížen na 1. Tato projekce probíhá tak, že pro každý pixel výstupního snímku se souřadnicemi $[t, x, y, c]$ se provede uživatelem zvolená matematická operace, jako je například součet nebo maximum, nad množinou všech voxelů výstupního snímku, které mají souřadnice t , x , y a c shodné s výsledným pixelem.

Možnými operacemi, které je možné provést nad snímky typu *stack* nebo *hyperstack* jsou:

- *Average intensity*,
- *Max intensity*,
- *Min intensity*,
- *Sum slices*,
- *Standard deviation*,
- a *Median*.

Operace používané pro projekci snímků

V tabulce na straně 13 jsou podrobně popsány jednotlivé operace, které je možné provádět v aplikaci IMAGEJ nad tří- a vícerozměrnými snímky. Ve sloupci *Výpočet* této tabulky je uveden matematický výraz, který je vyhodnocen pro každý pixel výsledného snímku. [7]



Obrázek 4.1:
Okno konfigurace
parametrů nástroje
Z Project v aplikaci
IMAGEJ

Ve výrazech ve sloupci *Výpočet* symbol Z označuje množinu všech možných hodnot na souřadnicové ose Z . Počet prvků množiny Z , označovaný $|Z|$, je tedy roven počtu jednotlivých rovin snímku.

Symboły t , x , y a c označují čas, x -ovou a y -ovou souřadnici a kanál, pro který je prováděna projekce snímku. V součtech je také použito z , což je prvek množiny Z . Zápis $\sum_{z \in Z}$ označuje součet všech hodnot, které se promítají do daného pixelu.

Hodnotou funkce $v(t, z, x, y, c)$ je hodnota zaznamenaná v c -tém kanále na souřadnicích (x, y, z) snímku v čase t . Není-li snímek časovou sekvencí, při výpočtu je snímek považován za časovou sekvenci o jednom snímku.

Informace shrnuté v této tabulce byly získány ze zdrojového kódu aplikace IMAGEJ, který je možné získat na adrese <https://imagej.nih.gov/ij/download.html>.

Název operace	Výpočet	Poznámky
<i>Average intensity</i>	$\frac{\sum_{z \in Z} v(t, z, x, y, c)}{ Z }$	Výsledkem je v každém pixelu průměrná hodnota všech bodů snímku, které se promítají do daného pixelu.
<i>Max intensity</i>	$\max_{z \in Z} v(t, z, x, y, c)$	Provedením této operace je vypočítána maximální hodnota všech bodů původního snímku, které se promítají do daného výsledného pixelu. Tato operace je velmi často používána při zobrazování snímků získaných za pomoci fluorescenční mikroskopie, protože větší objekty na snímku jsou označovány hodnotami vyššími než jaké se na snímku nacházejí v okolí objektů.
<i>Min intensity</i>	$\min_{z \in Z} v(t, z, x, y, c)$	Tato operace je podobná operaci <i>Max intensity</i> , ale namísto maxima je vypočítáno minimum.
<i>Sum slices</i>	$\sum_{z \in Z} v(t, z, x, y, c)$	Výsledek této operace je stejný jako výsledek <i>Average intensity</i> , ale odlišuje se tím, že je vynásoben počtem rovin snímku.
<i>Standard deviation</i>	$\sqrt{\frac{\sum_{z \in Z} v(t, z, x, y, c)^2 - \frac{(\sum_{z \in Z} v(t, z, x, y, c))^2}{ Z }}{1 - Z }}$	Vypočíte pro každý pixel snímku směrodatnou odchylku z množiny hodnot, které se promítají do daného pixelu. Pokud je výsledkem záporná hodnota, je nahrazena nulou.
<i>Median</i>	med Z	Vypočíte medián hodnot, které se promítají do daného pixelu.

Údaje obsažené v této tabulce vychází ze zdrojového kódu aplikace IMAGEJ ve verzi 1.53 (archiv se zdrojovými kódy zveřejněn 8. listopadu 2021). Pro vysvětlení významu symbolů $t, z, x, y, c, Z, |Z|$ viz stranu 12.

Tabulka 4.1: Matematické operace, které je možné v aplikaci IMAGEJ provádět nad tří- a vícerozměrnými snímky při projekci nástrojem *Z-project*

Operace *Max intensity*

Tato operace odpovídá tzv. projekci maximální intenzity, kdy je každému pixelu přiřazena barva odpovídající maximální hodnotě, která se vyskytuje v množině hodnot, které se promítají do daného pixelu výstupního snímku. Tato metoda je často používána ve fluorescenční mikroskopii.

Operace *Min intensity*

Tato operace je vhodná pro použití na snímcích, kde jsou objektu přiřazeny nižší hodnoty než pozadí.

Stejného výsledku je možná docílit tím, že provedete projekci za pomoci předchozí metody nad snímkem s invertovanými hodnotami¹ a výsledný snímek projekce je opět invertován.

Operace *Average intensity* a *Sum slices*

Tyto dvě operace se liší pouze tím, že hodnoty získané projekcí snímku o n rovinách metodou *Average intensity* jsou n -krát menší oproti hodnotám získaným metodou *Sum slices*.

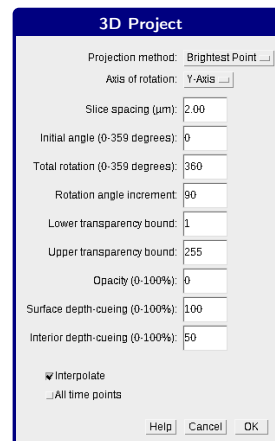
¹Inverzí hodnot se rozumí použití jakékoliv funkce f , pro kterou platí $\forall x, y \in R \wedge x < y, f(x) > f(y)$, kde R jsou všechny hodnoty, které mohou být reprezentovány pomocí datového typu, který je použit pro uložení vykreslovaného snímku. Pro inverzi snímků zaznamenávaných pomocí n -bitových celých čísel bez znaménka se zpravidla používá $f(x) = 2^n - x - 1$ (pro čísla se znaménkem je n o 1 menší).

4.1.3 IMAGEJ – nástroj 3D Project

Aplikace IMAGEJ umožňuje také vykreslování 3D projekce snímku otočeného kolem jedné z jeho prostorových os (osy x , y , nebo z), která je zvolena v rozbalovacím menu *Axis of rotation*.

Pokud je vykreslován stack, výstupem je časová sekvence dvojrozměrných snímků, kde každý snímek je vykreslen pod odlišným úhlem. Je-li vykreslován hyperstack, výstupem je opět hyperstack, kde časový rozměr odpovídá vstupnímu snímku a ve třetím prostorovém rozměru z poloha odpovídá úhlu, pod kterým byl snímek vykreslen. Je možné zvolit počáteční úhel (pole *Initial angle*), úhel, o který se má vykreslovaný kvádr otočit (*Total rotation*), a přírůstek úhlu (*Rotation angle increment*). [6]

Při vykreslování je snímek chápán buď jako několik samostatných rovin nacházejících se v prostoru (ukázka na obr. 4.3), nebo spojitý kvádr hodnot, ve kterém jsou hodnoty v bodech, které se nachází mezi body původního snímku interpolovány trilineární interpolací. Použití první možnosti na snímky s malým počtem rovin vede k nepřehlednému zobrazení snímku, je-li natočen o úhel, který se nachází blízko 90° a 270° . Při zobrazení pod úhlem přesně 90° nebo 270° se snímek vykreslí pouze jako rovnoběžné úsečky. To můžete vidět na obrázku 4.3b. Zřetelněji jsou tyto úsečky vidět na obrázku 4.3c, který je vytvořen stejným způsobem jako obr. 4.3b, ale navíc je zvýšen kontrast zobrazení. Zobrazení snímku v těchto polohách je kvalitnější po zapnutí interpolace. To vede k výsledku jaký můžete vidět na obrázku 4.3d.



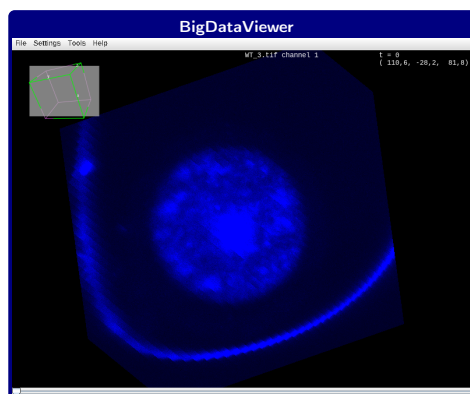
Obrázek 4.2: Okno konfigurace parametrů nástroje 3D Project v aplikaci IMAGEJ

4.1.4 IMAGEJ – nástroj BIGDATAVIEWER

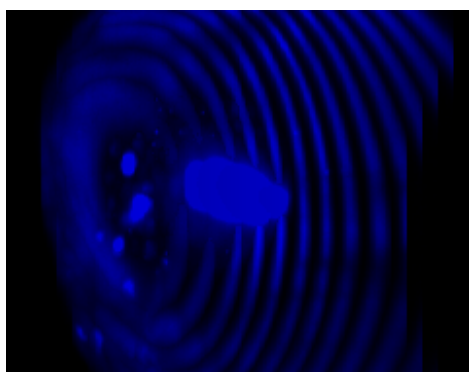
Dalším z nástrojů dostupných jako rozšíření aplikace IMAGEJ je BIGDATAVIEWER, který umožňuje interaktivní zobrazení řezu tří- nebo čtyřrozměrným snímkem s jedním nebo více kanály.

Při vykreslování snímků je možné zvolit ze dvou algoritmů interpolace: nearest-neighbour interpolace a trilineární. Je-li pro vykreslování použita trilineární interpolace, na výstupním obrázku se tolik neprojevují hranice mezi jednotlivými rovinami snímku.

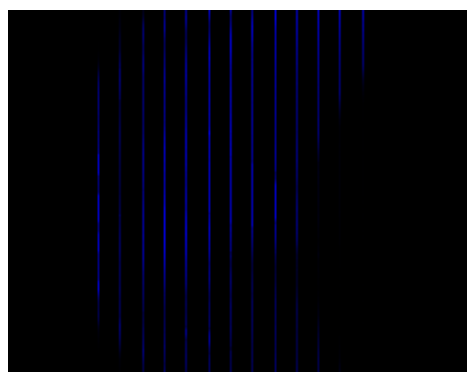
BIGDATAVIEWER používá techniku mipmapping, která dovoluje v případě zobrazení s nižším rozlišením než jaké má vstupní snímek vykreslovat jeho podvzorkovanou verzi. [8]



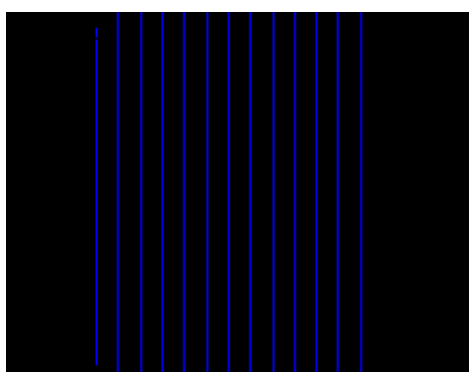
Obrázek 4.4: Okno nástroje BigDataViewer dostupného v aplikaci IMAGEJ



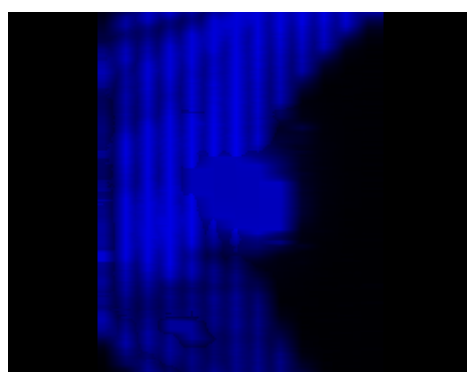
(a) rotace okolo osy Y o 45° , volba *Interpolate* vypnuta



(b) rotace okolo osy Y o 90° , volba *Interpolate* vypnuta



(c) rotace okolo osy Y o 90° , volba *Interpolate* vypnuta, zvýrazněno



(d) rotace okolo osy Y o 90° , volba *Interpolate* zapnuta

Obrázek 4.3: Projekce maximální intenzity třírozměrného mikroskopického snímku s 13 rovinami vytvořená pomocí nástroje *3D Projection* aplikace IMAGEJ

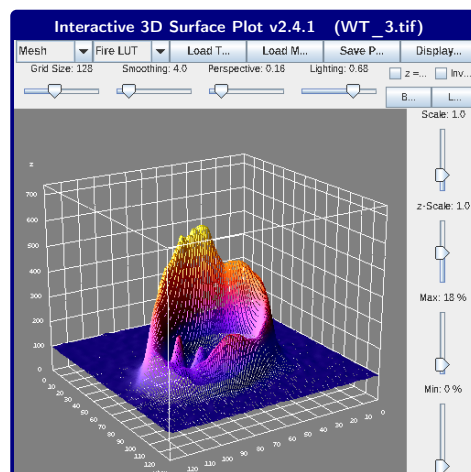
4.1.5 IMAGEJ – nástroj *3D Surface Plot*

Tento nástroj, na rozdíl od ostatních zde popisovaných, vykresluje graf intenzity v jednom dvojrozměrném snímku. Umožňuje vykreslovat graf jako mřížku (tečkami v průsečících mřížky nebo čarami), izoliniemi nebo jako spojitý povrch. Na obrázku 4.5 můžete vidět okno tohoto nástroje v posledním ze jmenovaných režimů.

Tento nástroj také umožňuje provést vyhlazení vykreslovaných dat a změnu měřítka. Graf je vykreslován v jedné barvě nebo může barva odpovídat hodnotě ve vykreslovaném bodě.

Je možné provést vykreslení v perspektivě. To je nastavováno pomocí posuvníku.

Jedinou změnou pohledu, kterou tento nástroj umožňuje je rotace pohledu. Tu je možné provést kliknutím a tažením myši v části okna, kam je vykreslován výstup.



Obrázek 4.5: Okno nástroje *3D Surface Plot* dostupného v aplikaci IMAGEJ

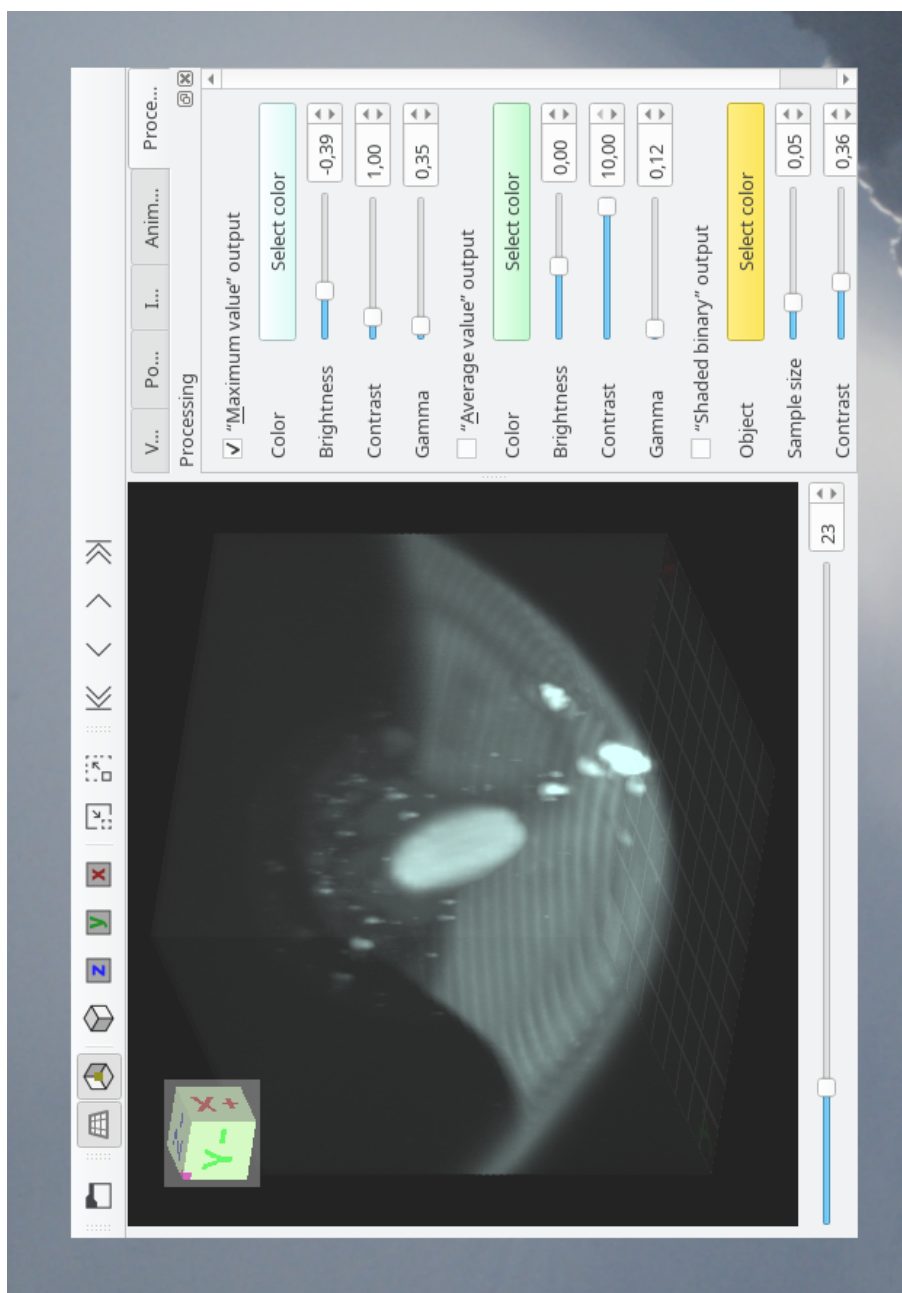
5 Aplikace HYPERSTACKVIEW

Hlavním tématem této práce je aplikace HYPERSTACKVIEW, která umožňuje uživatelsky přívětivým způsobem zobrazovat volumetrické snímky za účelem jejich analýzy. Aplikace umožňuje zobrazování třírozměrných snímků a procházení časové sekvence takových snímků (viz obrázek 1.1 na straně 4). Je umožněno současné zobrazování snímku a vystínované projekce objektů, které se na něm nacházejí. Informace o tom, kde se objektu nacházejí jsou poskytnuty programu pomocí druhého vstupního souboru.

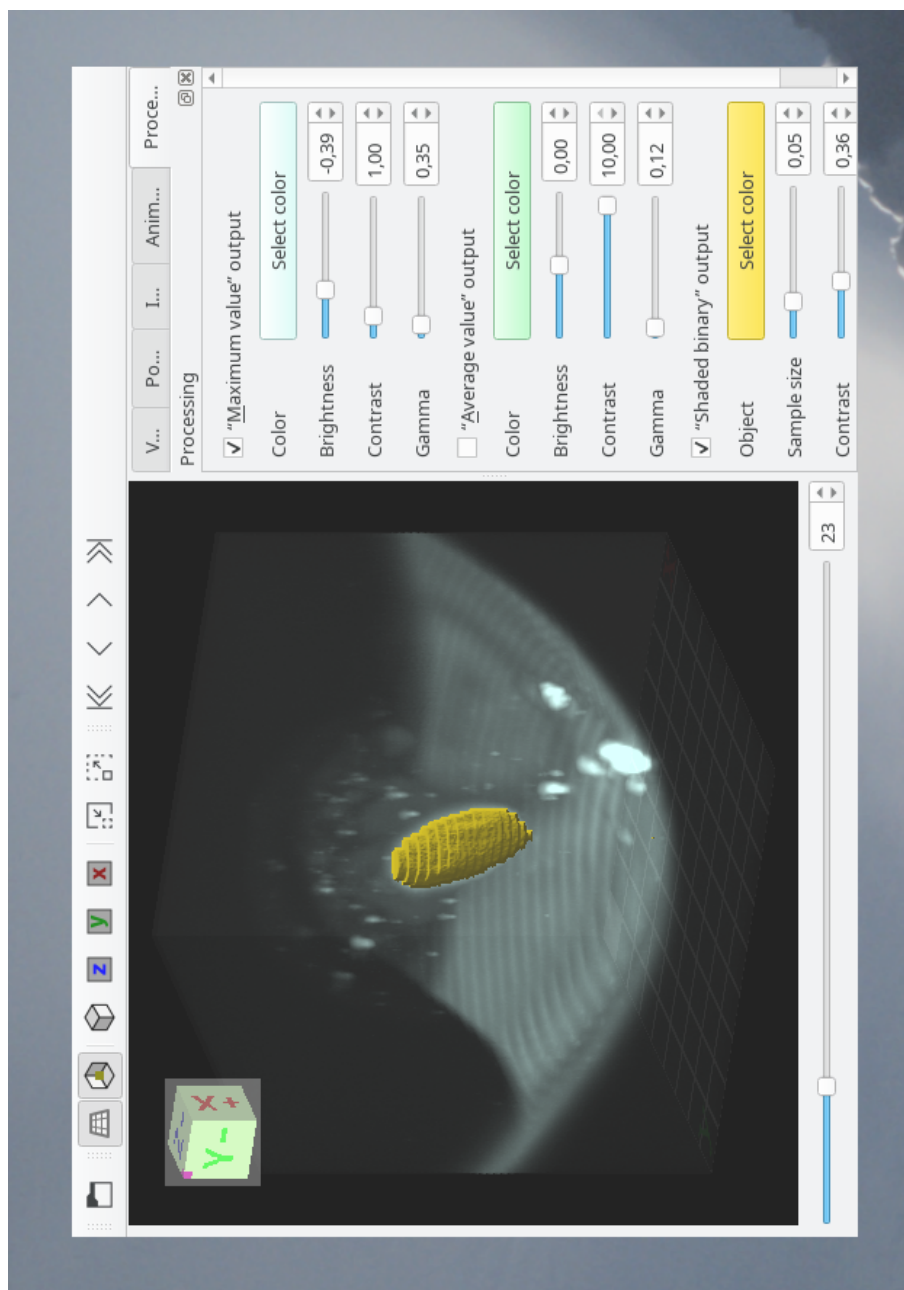
5.1 Snímky obrazovky

Na následující dvojstraně můžete vidět grafické uživatelské rozhraní hlavního okna aplikace. V obou případech je zobrazen třírozměrný snímek pořízený v rámci časové sekvence pomocí light-sheet mikroskopu.¹ Na druhém obrázku je zobrazení snímku doplněno znázorněním objektu, který byl získán za pomoci segmentace.

¹Vzhledem ke konfiguraci systému, na kterém byla aplikace spuštěna, není zobrazena hlavní nabídka aplikace.



Obrázek 5.1: Hlavní okno aplikace. Je zobrazen jeden mikroskopický snímek z časové sekvence v projekci maximální intenzity. (Hlavní nabídka aplikace je skryta.)



Obrázek 5.2: Hlavní okno aplikace. Je zobrazen tentýž snímek s překrytím znázorněním objektu, které bylo získáno pomocí segmentace snímku.

5.2 Použité knihovny

Program je naprogramován v programovacím jazyce Python 3 s použitím jeho standardní knihovny.

5.2.1 PySide (Qt)

Knihovna PySide poskytuje rozhraní v programovacím jazyce Python pro interakci s frameworkem Qt. Byla použita verze PySide6, který odpovídá Qt ve verzi 6.

Qt framework (též Qt) je frameworkem pro vývoj multiplatformních aplikací pro desktopové a embedded systémy, zejména aplikací s grafickým rozhraním. Poskytuje abstrakci nad rozhraním poskytovaným operačním systémem a také do jisté míry řeší tzv. reflexi, tedy možnost analyzovat a upravovat program za jeho běhu. Qt je primárně určeno pro použití s programovacím jazykem C++, ale byla vytvořena řada rozhraní pro použití v jiných jazycích, například již zmiňované PySide (dříve PyQt).

5.2.2 OpenGL

OPENGL je rozhraním, které umožňuje vykonávat program na grafické kartě počítače, kterým je zejména vykreslování třírozměrných scén v reálném čase. V této aplikaci je toto rozhraní (ve verzi 3.2 nebo vyšší) použito pro provedení projekce vstupních dat, filtrování vstupu a následné převedení na barevný rastrový obrázek. Bližší informace o tom jak je tato knihovna najdete v sekci 5.3 (strana č. 23). [9]

5.2.3 NumPy

Knihovny NumPy a SciPy jsou určené k provádění různých výpočetních operací v programovacím jazyce Python. Vzhledem k tomu, že Python je vykonáván většinou pomocí interpretru CPython, který neumožňuje efektivní provádění rozsáhlejších výpočtů, tyto knihovny poskytují implementaci řady matematických operací, které jsou vykonávány jako nativní kód.

Tato knihovna je použita pro převod datového typu snímků, maticové operace a výběr snímku k zobrazení.

5.2.4 Knihovna pro vytváření transformačních matic

Tato knihovna, která je blíže popsána v kapitole 6, která začíná na straně číslo 39, vznikla během vývoje aplikace HYPERSTACKVIEW pro usnadnění

vytváření transformačních matic, které jsou dále používány při výpočtech pomocí OpenGL. Tato knihovna poskytuje objektově-orientované a funkcionální rozhraní, která je možné použít ke snadnému a čitelnému zápisu transformací ve třídimenzionálním prostoru v programovacím jazyce Python. Tato knihovna interně používá pole knihovny NumPy a umožňuje s transformačními maticemi pracovat jako s polem knihovny NumPy.

5.2.5 Načítání souborů TIFF – knihovna *tiffle*

Knihovna *tiffle* slouží ke čtení a zapisování souborů v řadě variant formátu TIFF (viz stranu 2) a umožňuje k vybraným typům souborů přistupovat pomocí metody mapování souboru do paměti.

5.3 Vykreslování snímků

Aplikace HYPERSTACKVIEW je zaměřena na tzv. přímé vykreslování prostorových obrazových dat. To znamená, že nad snímkem je před jeho vykreslením provedeno minimum operací. Odlišným přístupem k vykreslování jsou nepřímé metody, kdy jsou data transformována například do polygonové sítě (anglicky *polygon mesh*).

Použitou metodu vykreslování můžeme rozdělit do několika kroků:

Následující kroky není nutné zopakovat při každém vykreslení snímku:

1. načtení snímku z diskového nebo jiného úložiště²;
2. převod snímku do vhodného formátu, je-li to nutné;

Tyto kroky probíhají při každém vykreslení:

3. výpočet transformačních matic pro (a) převod souřadnic z prostoru obrazovky do prostoru snímku, (b) opačný převod (c) a převod souřadnic z prostoru miniatury snímku do prostoru obrazovky s umístěním na příslušné místo na obrazovce;³
4. vykreslení stěn snímku;

²V případě použití časové sekvence snímku je načítán pouze jeden snímek ze sekvence.

³Tento krok je možné přeskočit, pokud se nezměnila poloha vykreslovaného snímku.

5. projekce snímku (projekce maximální a průměrné intenzity fotometrického snímku a výpočet vzdálenosti bodů objektů zaznamenaných na segmentovaném snímku;⁴
6. převod dat získaných v předchozím kroku na rastrový obrázek s barevnými hodnotami v barevném prostoru RGB
7. a vykreslení miniatury snímku přes vykreslený snímek.

Vynecháním kroků 4 a 7 je možné přizpůsobovat uživatelské rozhraní aplikace.

Výstup kroku číslo 5 (pro podrobnosti viz stránku 24) není přímo zobrazen na obrazovce – je ponechán v RAM GPU a slouží jako vstup pro krok číslo 6, kdy je převeden do barevného obrázku, který je možné zobrazit přímo na obrazovce (tj. v okně aplikace), a to tak, že je výstup projekce maximální a průměrné intenzity použit pro upravení jasu uživatelem zvolených barev a vzdálenosti bodů objektů zachycených na segmentovaném snímku jsou použity pro vykreslení „vystínovaného“ obrázku (strana č. 26).

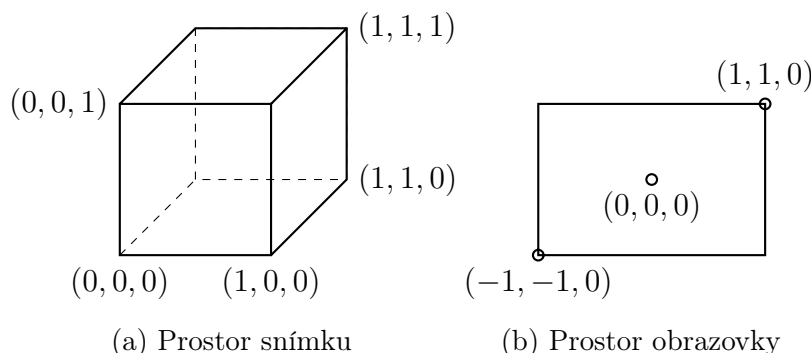
5.3.1 Projekce snímku

První fází vykreslování, kdy probíhá ke zpracovávání snímku samotného je projekce, která je ve zdrojovém kódu a dokumentaci programu označována jako *flatten* (zplošťování), protože zde dochází k převodu třírozměrných dat na data dvojrozměrná.

Při této projekci je používán *prostor snímku*, který je stanoven tak, aby počátek snímku, který je do tohoto prostoru umístěn, (jeden z vrcholů kvádrů prostoru zachyceného na snímku) se nacházel na souřadnicích $(0, 0, 0)$ a protější vrchol snímku se nacházel na souřadnicích $(1, 1, 1)$. Projekce probíhá na průmětnu, která je určena tak, že střed obrazovky (tj. části okna, kam je vykreslována projekce) se nachází na souřadnicích $(0, 0)$, levá a pravá hrana obrazovky má x -ovou souřadnici ± 1 a horní a spodní hrana obrazovky y -ovou souřadnici ± 1 . Protože i nadále je možné cíl projekce považovat za prostor, bude označení *prostor obrazovky* používáno pro prostor, ve kterém je průmětna umístěna na $z = 0$. Třetí rozměr je v tomto prostoru používán při vykreslování některých prvků uživatelského rozhraní. Rozsahy těchto prostorů můžete jsou schématicky znázorněné na obrázku 5.3.

Projekce probíhá tak, že jsou procházeny všechny pixely obrazovky, do kterých může být snímek vykreslen. Každému pixelu je přiřazena přímka v prostoru snímku. Při tom jsou brány v úvahu parametry jako například

⁴Výpočty používané pro projekce, které nejsou součástí výstupu není nutné provádět.



Obrázek 5.3: Schématické znázornění prostoru snímku a obrazovky

otočení snímku nebo nastavení perspektivy. Potom se zvolí určitý počet bodů snímku (tzv. vzorků), které se nacházejí na této přímce, které budou dále zpracovávány. Tento počet je určen parametrem *Z-axis sample count*. Při projekci maximální intenzity a průměrné intenzity je ke každému bodu přiřazena číselná hodnota (*hodnota vzorku*), která je ze snímku získána pomocí trilineární interpolace. Při zpracovávání druhého vstupu, segmentace snímku, je použita *nearest-neighbour* interpolace.

5.3.2 Projekce maximální intenzity

Toto je jedna z nejčastěji používaných projekcí při zobrazování volumetrických snímků získaných pomocí moderních mikroskopických metod. Jak můžete vidět v tabulce 4.1 (strana č. 13), i aplikace IMAGEJ umožňuje použít tuto projekci pro volumetrické snímky.

Probíhá tak, že se spočítá maximální hodnota ze všech hodnot vzorků. Tato projekce poskytuje poměrně zřetelné zobrazení objektů na snímku, kde pozadí přísluší nižší hodnoty než objektům. [11]

Další informace o výpočtu naleznete na straně 13.

5.3.3 Projekce průměrné intenzity

Stejně jako předchozí metoda, i tato je podporována v aplikaci IMAGEJ.

V HYPERSTACKVIEW je implementována tak, že se nejprve provede součet všech hodnot vzorků a následně až ve fázi *final* jsou vyděleny počtem vzorků. Vzorky ležící mimo snímek jsou brány v úvahu, a to jako body s hodnotou 0.

Výpočet je popsán na straně 13.

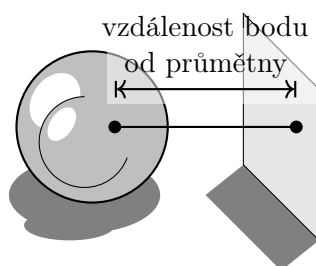
5.3.4 Zpracovávání snímku s objekty

Dalším vstupem aplikace HYPERSTACKVIEW je snímek (třídímenzionální pole) s informacemi o objektech na snímku. Rozměry takového snímku musí být stejné jako rozměry fotometrického snímku. Hodnotami na tomto snímku jsou celá čísla.

Nejjednodušším použitím tohoto vstupu je označení oblastí, na kterých se nachází jednotlivé objekty. Hodnota 0 označuje příslušnost k pozadí (tj. bod nepříslušející objektu) a kladné číslo označuje příslušnost k objektu. Díky této informaci je možné vykreslit vystínovanou projekci těchto objektů.

Pokud je známo, že se na snímku nachází několik různých objektů, je možné každému z nich přiřadit odlišná kladná čísla. V HYPERSTACKVIEW je potom možné vybírat jednotlivé objekty ke zobrazení nebo je zvýrazňovat na snímku.

Ve fázi *flatten* probíhá první krok zpracovávání. Je vypočtena vzdálenost vzorků, které se nachází v objektu, od jednotlivých bodů na průmětně, které odpovídají pixelům obrazovky. Tato vzdálenost je potom použita pro vykreslení vystínovaného zobrazení ve fázi *final*. To je znázorněno na obrázku 5.4.



Obrázek 5.4: Vzdálenost bodu objektu od průmětny

5.3.5 Vykreslování projekce

Následující fázi vykreslování, kdy dochází k manipulaci s obrazovými daty je fáze *final*, tedy finální fáze, ve které jsou dvojrozměrná pole s abstraktními číselnými hodnotami získaná ve fázi *flatten* převedena na dvojrozměrná pole barevných hodnot v barevném prostoru RGB.

Ve fázi *flatten* jsme získali pro každý pixel obrazovky tyto informace (a to i pokud danému pixelu neodpovídá žádný objekt):

- maximální intenzita ve všech vzorcích, které se do daného pixelu promítají;

- průměrná intenzita ve všech vzorcích, které se do daného pixelu promítají
- a nejmenší možná vzdálenost bodu na objektu od průmětny, který se promítá do daného pixelu, je-li v datech takový objekt vyznačen.

V paměti jsou uloženy jako dvojrozměrná pole. První dvě pole byly získány z fotometrického snímku a jde o reálná čísla (v použité implementaci algoritmu čísla s plovoucí řádovou čárkou). Pokud se do daného pixelu nepromítá žádná část snímku, je jim přiřazena hodnota 0.

Každý prvek v poli se vzdálenostmi bodů objektů odpovídá buď nějakému bodu na objektu, který je určen ve vstupních datech, nebo, pokud žádný takový bod neexistuje, je mu přiřazena zástupná hodnota. Může jít o jakoukoliv hodnotu, pokud možno takovou, která nemůže vzniknout v bodu, do kterého se nějaký objekt promítá. Budeme ji označovat ∞ .⁵

Převod intenzity bodu na barevnou hodnotu

Pro vykreslení výsledku projekce maximální intenzity a projekce průměrné intenzity je možné používat takzvanou vyhledávací tabulku (anglicky *lookup table*, LUT). Tabulka je polem barevných hodnot, které jsou při převodu číselné hodnoty na barvu voleny podle hodnoty, která je zobrazována.

Jinou možností je určit funkci, jejímž parametrem je hodnota k převedení a hodnotou této funkce je barva, která má být zobrazena. Tento přístup byl zvolen pro implementaci v aplikaci HYPERSTACKVIEW, protože umožňuje snazší přizpůsobení zobrazení. To probíhá tak, že je umožněno uživateli zvolit několik konstant, které tato funkce používá. Tyto konstanty jsou:

Označení	Popis	Rozsah
b	Jas	−1.0 až 1.0
c	Kontrast	0.0 až 10.0
γ	Gamma	0.0 až 10.0
C	Barva	

Tabulka 5.1: Parametry převodu intenzity na barevnou hodnotu.

Výstup bude obarven barvou **C**, například pro odlišení projekce maximální intenzity od vystínovaných objektů zobrazených v jednom snímku. Barva je

⁵Z důvodu snazší implementace byla v programu použita záporná hodnota, která taktéž nemůže při projekci vzniknout.

tříprvkovým vektorem, kde jednotlivé prvky vektoru určují intenzitu červené složky \mathbf{C}_R , zelené složky \mathbf{C}_G a modré složky \mathbf{C}_B :

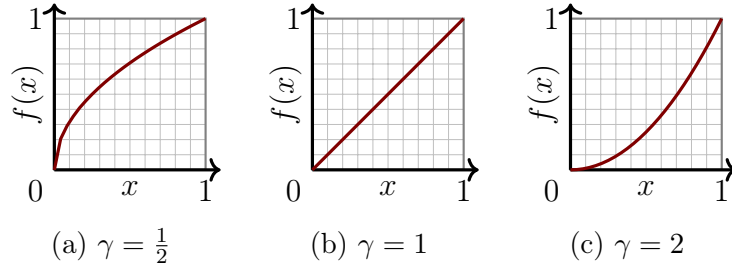
$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_R \\ \mathbf{C}_G \\ \mathbf{C}_B \end{pmatrix}$$

Funkce $f(x)$, která převádí hodnotu x na výslednou barvu (opět jako vektor ve stejném formátu) má předpis:

$$f(x) = \mathbf{C} \cdot (c \cdot x^\gamma + b)$$

Pokud budeme předpokládat, že $0 \leq x \leq 1$ a $\gamma = 1$, potom hodnoty b a c určují rozsah jasů, které mohou být danému pixelu na obrazovce přiřazeny, přičemž jas pixelu je přímo úměrný zobrazované hodnotě.

Zřetelnější zobrazení zpravidla poskytne nelineární mapování hodnot na jas pixelu. Pokud $\gamma > 1$, budou zdůrazněny rozdíly mezi vyššími hodnotami. Pokud je $\gamma < 1$, tyto rozdíly budou naopak potlačeny a jsou zdůrazněny odlišnosti v nízkých hodnotách. Na obrázku 5.5 můžete vidět jaký má vliv změna hodnoty γ na průběh funkce $f(x)$ (přesněji na velikost vektoru, který je její hodnotou)⁶.



Obrázek 5.5: Vliv parametru γ na průběh funkce $f(x)$

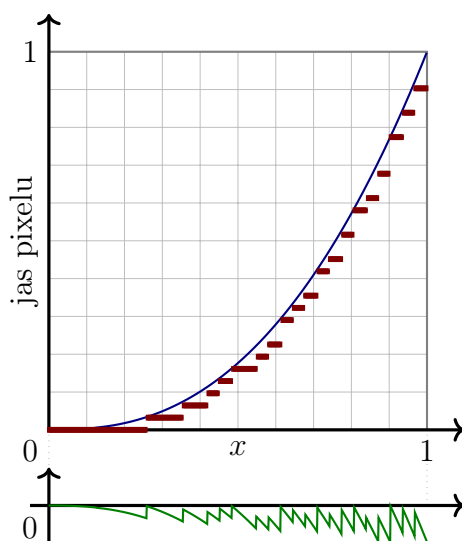
Tento parametr γ , jak již jeho označení napovídá, je možné použít pro tzv. gamma korekci zobrazení, která je používána i při práci s fotografiemi, protože poměry jasů jednotlivých částí výsledného snímku odpovídají více tomu, jak jsou vnímány lidským zrakem.

Na základě porovnávání různých nastavení parametru γ při vykreslování třírozměrných mikroskopických snímků jsem došel k tomu, že hodnoty $\gamma < 1$ poskytují zřetelnější zobrazení.

Aplikace je navržena tak, že hodnoty ze snímku jsou uchovávány jako čísla s plovoucí řádovou čárkou až do provedení této funkce, takže dochází

⁶Byly použity hodnoty $c = 1$, $b = 0$ a $\|\mathbf{C}\| = 1$.

k minimální ztrátě přesnosti zobrazení. Pokud by například byly hodnoty převedeny na 8bitová celá čísla před i po provedení funkce $f(x)$, tak v případě, že $\gamma \neq 1$, by některé hodnoty této funkce po převodu na 8bitové číslo (které je běžně používáno při vykreslování bitmapové grafiky na obrazovky) odpovídaly více než jedné hodnotě x . Na výsledném obrázku by došlo ke zkreslení a neplynulé změně jasu pixelů v místech, kde by měla být plynulá. Použití čísel s plovoucí řádovou čárkou tento problém neodstraňuje zcela, ale zmenšuje jeho projevy tak, že jsou nezřetelné. Na obrázku 5.6 můžete vidět porovnání použití celočíselného datového typu s použitím čísel s plovoucí řádovou čárkou.



Obrázek 5.6: Změna jasu pixelu může být nepravidelná, pokud dojde k přechodu na celá čísla před výpočtem barevné hodnoty pixelů. Pro porovnání jsou vykresleny i hodnoty, které jsou získány bez převodu na celá čísla. Na spodním grafu je zakreslen rozdíl těchto dvou křivek.

Poznámka: Pro zřetelnost byla použita pouze 5bitová čísla. Při použití 8bitového čísla by nepravidelnosti byly takřka neviditelné na grafu, avšak při použití pro určení jasu pixelu znatelné.

Vykreslení objektů na snímku

Při práci s volumetrickými snímky je někdy potřeba zjistit informace o jednotlivých objektech na snímku. Proces, který ze snímku získá informace o příslušnosti jeho bodů (ve 2D pixelů, ve 3D voxelů) k jednotlivým objektům

se nazývá *segmentace*. Aplikace HYPERSTACKVIEW umožňuje výstup získaný segmentací použít, kromě jiného, pro vykreslení objektů současně s původním snímkem nebo namísto původního snímku.

Během fáze vykreslování *flatten* (strana č. 24) již byly získány informace o tom, jak daleko jsou nejbližší body objektů od průmětny (viz obrázek 5.4 na straně 26). Ve fázi *final* jsou převedeny na obrázek, který je následně zobrazen.

Při zobrazování je žádoucí, aby byl znázorněn tvar objektu. Je sice možné vykreslovat objekty získané segmentací, ale potom výsledné zobrazení poskytuje pouze informaci o tvaru obrysu objektu, nikoliv o jeho povrchové struktuře.

5.3.6 Projekce s transformací

Při projekci prostorového snímku je vhodné umožnit zobrazení z několika různých pohledů. To je v aplikaci HYPERSTACKVIEW umožněno tím, že uživatel může pomocí klávesnice nebo myši otáčet pohled na snímek a upravovat jeho zvětšení. (Je možné provádět také další akce, například zobrazení animace.)

Při vykreslování snímku je možné stav otočení a zvětšení snímku chápat jako číselné parametry. Pro efektivní transformaci vektorů používaných při vykreslování je použita tzv. transformační matice. [13] Ta je vypočtena s pomocí knihovny pro práci s transformačními maticemi, která je blíže popsána v kapitole 6 (strana č. 39).

Před zahájením vykreslování je vypočteno několik transformačních matic. Jedna z nich slouží k převodu vektorů (resp. souřadnic) z prostoru obrazovky do prostoru snímku. Ta je použita při přístupu k obrazovým datům. Další jsou používány pro vykreslování grafických prvků do zobrazeného snímku (to znamená, že jde o opačný převod) a zobrazování grafického prvku, který znázorňuje aktuální orientaci snímku, v rohu okna aplikace.

Perspektiva

Při prohlížení projekce trojrozměrných snímků může snazšímu pochopení zobrazeného snímku napomáhat také zobrazení v perspektivě. To znamená, že části snímku vzdálenější od průmětny jsou vykresleny menší než části, které jsou k průmětně blíže. Aplikace HYPERSTACKVIEW umožňuje zobrazit snímek s pomocí jednoúběžníkové perspektivy, která je blíže popsána v kapitole 6 (strana č. 39). Zobrazení v perspektivě je nastavitelné pomocí posuvníku a vstupního pole, kterým je možné nastavit hodnotu, která odpovídá zornému úhlu.

5.4 Překryvné grafické prvky

Další funkcí prohlížeče snímků je možnost souběžně se snímkem zobrazit grafické prvky, které usnadňují orientaci ve snímku. Jsou poskytovány tři takové prvky:

- náhled rotace snímku se znázorněním zobrazené části,
- překryvná mřížka,
- a stínování protějších stěn snímku.

5.4.1 Náhled rotace snímku

Jde o zobrazení kváдру ve stejné pozici jako je zobrazovaný snímek. Stěny kváдру jsou vystínované a označené. Obdélníková oblast odpovídající aktuálně zobrazené části snímku je zvýrazněná.

Tento náhled je zobrazován v levém horním rohu vykresleného snímku.

5.4.2 Překryvná mřížka

Pro přibližné porovnávání vzdáleností je možné použít mřížku, která je do snímku vložena tak, že se nachází v rovině $z = 0$.

5.4.3 Stínování protějších stěn snímku

Tři ze stěn kváдру, který vymezuje zobrazovaný snímek, které jsou nejvzdálenější od průmětny, mohou být volitelně vystínovány. To umožňuje zřetelně zobrazit hranice snímku a to, v jaké je orientaci vykreslen.

5.5 Filtrování vstupních dat

Aplikace HYPERSTACKVIEW umožňuje zobrazovat pouze zvolené objekty na snímku nebo je zvýrazňovat. Je k tomu nutné vytvořit externím nástrojem soubor ve formátu TIFF, ve kterém jsou zaznamenány informace o objektech. Formát snímku s objekty je blíže popsán na straně 26.

Informaci o příslušnosti pixelů k objektům je možné použít pro filtrování snímku, tedy zobrazení pouze těch pixelů, které náleží ke zvolenému objektu. To je provedeno tak, že ve fázi vykreslování *flatten* jsou pixely pozadí a všech objektu, kromě zvoleného, ignorovány.

Dalším použitím je zvýraznění objektů. V případě zapnutí této funkce je vykreslování stínovaných objektů pozměněno tak, že jsou vykresleny pouze pixely, které přísluší ke zvolenému objektu.

5.5.1 Použití v kombinaci s jinou aplikací

Aby nebylo nutné ruční zadávání čísla objektu ke zobrazení, byla do prohlížeče snímků přidána možnost automaticky zvolit objekt, jehož číslo bylo označeno i v jiné běžící aplikaci⁷ nebo číslo, které bylo naposledy uloženo do schránky zobrazovacího serveru (např. při použití rozhraní WAYLAND), příp. schránky operačního systému. To usnadňuje zadávání hodnot.

Během používání aplikace HYPERSTACKVIEW při vývoji metod segmentace trojrozměrných snímků byl prohlížeč snímků používán v kombinaci s tabulkovým procesorem LIBREOFFICE CALC nebo textovým editorem, jehož uživatelské rozhraní je vykreslováno za použití X WINDOW SYSTEM. V tomto případě je objekt k vykreslení zvolen trojitým klepnutím myši na číslo objektu v tabulkovém procesoru, resp. dvojitým klepnutím na číslo objektu v textovém editoru.

5.6 Uživatelské rozhraní aplikace

Uživatelské rozhraní aplikace bylo vytvořeno s pomocí knihovny PySide6, která umožňuje vytváření programů s GUI v programovacím jazyce Python, které mohou být spuštěny na mnoha operačních systémech včetně systémů LINUX (s X WINDOWS nebo WAYLAND) nebo Microsoft Windows.

Výstup programu je vykreslován do *hlavního okna*, ve kterém se nachází i ovládací prvky pro nastavení zobrazení a volbu vstupních souborů.

5.6.1 Hlavní okno

Hlavní okno můžeme rozdělit do třech částí:

- **vykreslený snímek** s posuvníkem pro volbu snímku časové sekvence (vlevo),
- **podokna nastavení** (vpravo)
- a **hlavní nabídka** s panelem nástrojů (nahore)

⁷To je možné provádět pouze na systémech, kde je pro vykreslování grafického rozhraní používán X WINDOW SYSTEM.

Vykreslený snímek

Většinu této části hlavního okna zabírá prostor, kam je vykreslován snímek pomocí knihovny OpenGL. Ve spodní části se nachází posuvník pro zadávání čísla snímku časové sekvence pro zobrazení. (Více informací o těchto posuvnících je na straně 33.) Pokud není zobrazována časová sekvence, posuvník zobrazuje číslo snímku 0 (tj. první snímek) a je vykreslen jako neaktivní (tedy nereaguje na akce uživatele).

Podokna nastavení

Parametry zobrazení jsou zadávány pomocí několika podoken nastavení. Jde o části hlavního okna, které jsou ve výchozím nastavení umístěné na pravé straně okna. Uživatel je může za běhu okna přesunout k jinému okraji hlavního okna, spojit je do jednoho podokna se záložkami nebo je horizontálně či vertikálně rozdělit. Také je možné podokno vyjmout z hlavního okna a vytvořit z něj plnohodnotné okno okenního systému.

V těchto podoknech jsou pro zadávání většiny číselných hodnot použity posuvníky s polem pro číselnou hodnotu (viz stranu 33).

V současnosti aplikace používá tato podokna:

Podokno *View* V tomto podokně se nachází nastavení zobrazení v perspektivě, vzorkování vstupu a překryvných grafických prvků.

Podokno *Position* Zde se nachází ovládací prvky pro otáčení zobrazovaného snímku a nastavení zvětšení.

Podokno *Input* Toto podokno umožňuje nastavit rozsah vstupních hodnot manuálně nebo automaticky.

Podokno *Animation* poskytuje nastavení animace a tlačítko pro spuštění a zastavení animace.

Podokno *Processing* umožňuje nastavit barvu a další parametry jednotlivých projekcí a jednotlivě je zapínat a vypínat.

Posuvníky pro zadávání číselných hodnot

Z důvodu potřeby zadávání číselných hodnot na mnoha místech v aplikaci byl vytvořen nový prvek uživatelského rozhraní, který kombinuje posuvník

a pole pro zadávání číselné hodnoty s několika variantami. Těm odpovídají třídy definované ve zdrojovém kódu v souboru `sliders.py`. Na obrázku 5.7 na straně 35 můžete vidět odpovídající diagram tříd. Variantami těchto ovládacích prvků jsou:

- **SliderWithNumber** — posuvník s polem pro zadávání čísla s nastavitelným počtem desetinných míst a jednotkou;
- **RoundSliderWithNumber** — obdobný prvek, který ale vykresluje namísto lineárního posuvníku kulaté kolečko;
- **LabelledSliderWithNumber** — posuvník s popiskem nahoře nebo nalevo;
- **SliderWithNumberAndResetButton** — posuvník s tlačítkem pro vynulování dole nebo napravo;
- **RotationSliderWithNumber** — kombinace třech předchozích variant specializovaná na zadávání úhlu pohledu zobrazení (ke kombinaci vlastností je využita vícenásobná dědičnost poskytovaná programovacím jazykem Python);
- **AnimationAngularSpeedSlider** a **AnimationTimeSpeedSlider** — posuvníky s nulovacím tlačítkem označeným „Stop“, které nastavují rychlost otáčení zobrazení a procházení časové sekvence snímků.

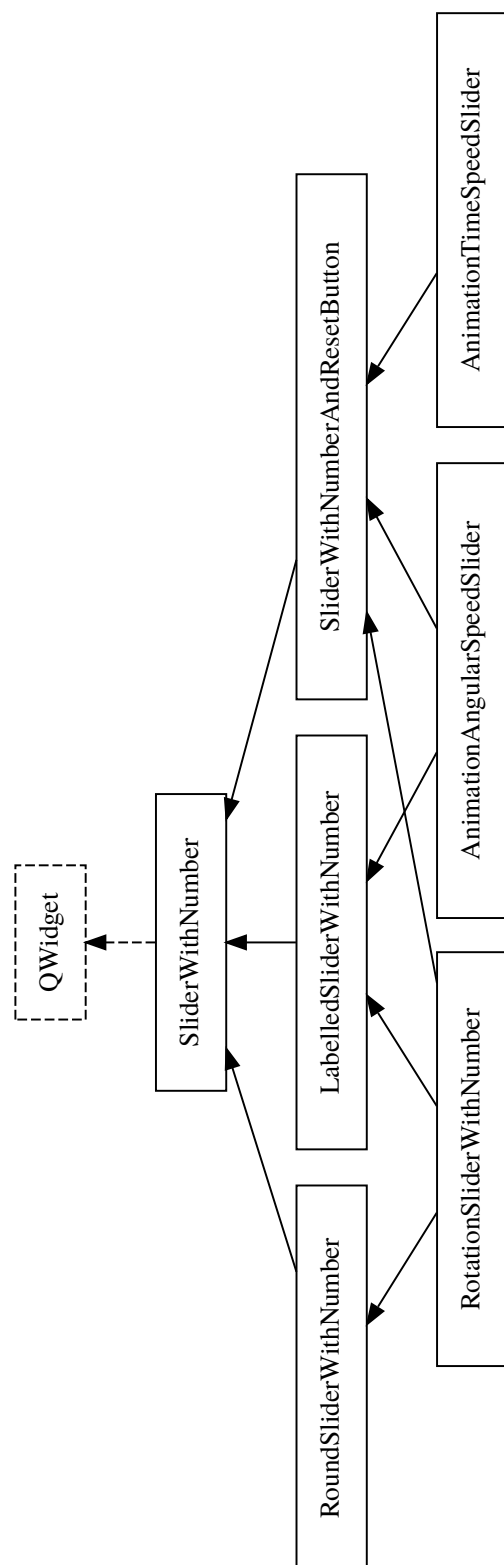
V aplikaci je nejpoužívanější varianta, kde se vedle sebe nachází posuvník a pole s číslem, které je doplněno tlačítky pro jemné zvýšení a snížení hodnoty. Muže být zadáváno celé nebo desetinné číslo, u kterého může být zobrazována měrná jednotka.

5.6.2 Ovládání zobrazení myši

Stisknutím levého tlačítka myši nad zobrazeným snímkem a tažením nahoru, dolů či do stran je možné upravit otočení snímku.

Stisknutím prostředního tlačítka myši nad zobrazeným snímkem a tažením nahoru či dolů je změněno zvětšení zobrazovaného snímku.

Otáčením kolečka myši nad zobrazeným snímkem je také upravována velikost zobrazení.



Obrázek 5.7: Strom dědičnosti tříd používaných pro posuvníky.

5.6.3 Klávesové zkratky

V aplikaci je dostupné ovládání prostřednictvím řady klávesových zkratk. Pro činnosti, které jsou definované v *Qt Framework*, byly použity klávesové zkratky z něj, které se liší podle toho, na kterém operačním systému aplikace běží. Zpravidla jsou použity tyto klávesové zkratky:

Položka nabídky	Akce	Zkratka ⁸
File → Open TIFF...	Otevře soubor	Ctrl-O
→ Unload all images	Zavře všechny soubory	Ctrl-W
→ Quit	ukončí aplikaci	Ctrl-Q

Tabulka 5.2: Klávesové zkratky, které se mění v závislosti na používaném operačním systému

Na následujících obrázcích je znázorněno přiřazení funkcí jednotlivým klávesám na numerické klávesnici a číselných klávesách v alfanumerické části klávesnice:



Obrázek 5.8: Přiřazení funkcí klávesám alfanumerické klávesnice



Obrázek 5.9: Přiřazení funkcí klávesám numerické klávesnice

Další klávesové zkratky nejsou závislé na používaném operačním systému:⁹

	Položka nabídky	Akce	Zkratka
Aplikace	File → Open TIFF...	Otevřít soubor TIFF	<i>Ctrl-O</i>
	→ Unload all images	Zavřít všechny soubory	<i>Ctrl-W</i>
	→ Object selector	Nástroj pro filtraci podle objektů	Ctrl-Shift-O
	→ Quit	Ukončit aplikaci	<i>Ctrl-Q</i>
Překryvné prvky	View → Enable grid	Zobrazit/skrýt mřížku	Shift-G
	→ Enable cube backdrop	Zobrazit/skrýt stínování protějších stěn	Shift-B
	→ Enable preview of rotation	Zobrazit/skrýt náhled rotace	Shift-O
Změna pohledu	View → Initial position	Výchozí poloha	0
	→ Rotate X+	Rotovat X+	8
	→ Rotate X-	Rotovat X-	2
	→ Rotate Y+	Rotovat Y+	7
	→ Rotate Y-	Rotovat Y-	3
	→ Rotate X+	Rotovat Z+	6
	→ Rotate X-	Rotovat Z-	4
	→ Zoom in	Přiblížit	Ctrl- +
	→ Zoom out	Oddálit	Ctrl- -
Časové sekvence	Multi-frame → First frame	První snímek	Home, Ctrl-1
	→ Previous frame	Předchozí snímek	Page Up, 1
	→ Next frame	Následující snímek	Page Down, 9
	→ Last frame	Poslední snímek	End, Ctrl-9
	→ Go to frame	Zadat číslo snímku ke zobrazení	Ctrl-G

Tabulka 5.3: Klávesové zkratky aplikace HYPERSTACKVIEW

⁹Pro úplnost tabulky byly uvedeny i zkratky závislé na operačním systému. Jejich obvyklé přiřazení je napsané kurzívou.

6 Knihovna pro transformace

V této aplikaci je nutné během zobrazování výstupu nutné pracovat s transformačními maticemi. Existující knihovny, které umožňují výpočet transformačních matic, jako například *NumPy* nebo *SciPy*, poskytovaly rozhraní, které nebylo pro programátora snadno použitelné, a tak byla vytvořena minimalistická knihovna v programovacím jazyce Python, která obsahuje funkce pro výpočet matic odpovídajících běžně používaným transformacím, skládání těchto transformací a transformaci vektorů pomocí těchto matic.

Knihovna je navržena tak, aby bylo možné vypočtené transformační matice použít také tam, kde je očekávaná matice ve formě pole knihovny *NumPy*, která je v aplikaci HYPERSTACKVIEW používána při zpracovávání snímků.

6.1 Transformační matice ve 3D grafice

Ve 3D grafice jsou často používány transformační matice o velikosti 4×4 , které jsou maticí o velikosti 3×3 rozšířenou o jeden další sloupec a řádek, takže je možné pomocí nich popsat i posunutí a s použitím dalších výpočtů i zobrazení v perspektivě. Jak je znázorněno na obrázku 6.1, otočení, škálování a některé další operace ovlivňují pouze část matice, která odpovídá transformační matici o velikosti 3×3 . Pravý sloupec matice, tedy hodnoty t_x , t_y a t_z , je změněn pouze posunutím (pravá spodní hodnota zůstává vždy 1). Spodní řádek je potom použit spolu s dalšími operacemi pro vytvoření zobrazení v perspektivě. [13]

Transformace je možné skládat pomocí maticového součinu. Pomocí násobení dvou matic je možné vypočítávat transformační matice, které provádí určitou posloupnost transformací, v jednom kroku. Transformace vektoru je provedena součinem matice s vektorem. [13]

6.2 Podporované transformace

Tato knihovna obsahuje několik základních transformací, ze kterých je možné skládat složitější transformace. Níže je výčet všech takových transformací včetně ukázek použití. Je možné je používat dvěma způsoby:

- **objektovým**, kdy se nejprve vytvoří objekt reprezentující transformační matici, na kterém je možné volat metody, které odpovídají jednotlivým transformacím;
- nebo **funkcionálním**, kde jsou volány funkce z této knihovny, které vytvářejí transformační matice nebo transformace aplikují na vektor či jinou matici, která jim je předána parametrem.

Tyto způsoby použití jsou blíže popsány v sekcích 6.4 a 6.3.

6.2.1 Posunutí

Použití funkce <code>transform.translate</code>	
Objektové rozhraní: <pre>import transform transform.identity() \ .translate([x, y, z]) \ .apply_to(vector)</pre>	Funkcionální rozhraní: <pre>import transform transform \ .translate([x, y, z], \ apply_to=vector)</pre>

Posunutí o danou vzdálenost po ose x , y a z o vzdálenost t_x , t_y a t_z . Výsledná matice:

$$\mathbf{M} = \left(\begin{array}{ccc|c} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

Vektor může mít 1 až 4 prvky. Pokud je prvků méně než 3, je provedeno posunutí po ose x , popř. osách x a y . Při použití čtyřprvkového vektoru se předpokládá, že poslední prvek je roven jedné.

6.2.2 Škálování

Použití funkce <code>transform.scale</code>	
Objektové rozhraní: <pre>import transform transform.identity() \ .scale([x, y, z] <i>nebo</i> n) \ .apply_to(vector)</pre>	Funkcionální rozhraní: <pre>import transform transform \ .scale([x, y, z] <i>nebo</i> n, \ apply_to=vector)</pre>

Pokud je funkci předána skalární hodnota s nebo vektor o jednom prvku s ,

provede rovnoměrné škálování. Je vytvořena následující matice:

$$\mathbf{M} = \left(\begin{array}{ccc|c} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

Pokud je funkci předán vektor o třech prvcích $\begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix}$, je provedeno nerovnoměrné škálování, kterému odpovídá tato matice:

$$\mathbf{M} = \left(\begin{array}{ccc|c} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

6.2.3 Rotace kolem osy určené vektorem

Použití funkce `transform.rotate_about`

Objektové rozhraní:

```
import transform
transform.identity() \
    .rotate_about([1, 2, 3], \
                  3.14 / 2) \
    .apply_to(vector)
```

Funkcionální rozhraní:

```
import transform
transform \
    .rotate_about([1, 2, 3], \
                  3.14 / 2, \
                  apply_to=vector)
```

Této funkci je předáván vektor a se třemi prvky $\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$, který určuje osu, kolem které bude otáčení probíhat, a úhel α v radiánech.

Této transformaci odpovídá následující matice:

$$\mathbf{M} = \left(\begin{array}{ccc|c} & & & 0 \\ & \mathbf{R} & & 0 \\ & & & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right),$$

kde

$$\mathbf{R} = \begin{pmatrix} n_x^2 \cdot (1 - c) + c & n_y n_x \cdot (1 - c) - n_z \cdot s & n_z n_x \cdot (1 - c) + n_y \cdot s \\ n_x n_y \cdot (1 - c) + n_z \cdot s & n_y^2 \cdot (1 - c) + c & n_z n_y \cdot (1 - c) - n_x \cdot s \\ n_x n_z \cdot (1 - c) - n_y \cdot s & n_y n_z \cdot (1 - c) + n_x \cdot s & n_z^2 \cdot (1 - c) + c \end{pmatrix},$$

$$c = \cos \alpha,$$

$$s = \sin \alpha,$$

$$n = \frac{a}{\|a\|}$$

6.2.4 Prohození souřadných os

Použití funkce `transform.swap_axes`

Objektové rozhraní:

```
import transform
transform.identity() \
    .swap_axes(0, 1) \
    .apply_to(vector)
```

Funkcionální rozhraní:

```
import transform
transform \
    .swap_axes(0, 1, \
                apply_to=vector)
```

Této funkci jsou předána dvě čísla $a, b \in \langle 0; 2 \rangle$, která určují, které dvě souřadné osy zaměnit. Tato funkce vrací jednotkovou matici, ve které je a -tý a b -tý řádek prohozen.

6.2.5 Odstranění souřadné osy

Použití funkce `transform.drop_axis`

Objektové rozhraní:

```
import transform
transform.identity() \
    .drop_axis(0) \
    .apply_to(vector)
```

Funkcionální rozhraní:

```
import transform
transform \
    .drop_axis(0, \
                apply_to=vector)
```

Tato funkce vytvoří matici, která odstraňuje informaci o jedné souřadné ose. Následující matice odpovídají odstranění os x , y a z (v tomto pořadí):

$$\mathbf{M}_x = \left(\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$
$$\mathbf{M}_y = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$
$$\mathbf{M}_z = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

6.2.6 Převod do prostoru textur a zpět

Použití funkce `transform.texture_to_screen`

Objektové rozhraní:

```
import transform
transform.identity() \
    .texture_to_screen() \
    .apply_to(vector)
```

Použití funkce `transform.screen_to_texture`

Objektové rozhraní:

```
import transform
transform.identity() \
    .screen_to_texture() \
    .apply_to(vector)
```

Tyto dvě transformace jsou určeny pro použití speciálně při používání rozhraní OpenGL. V něm je při přístupu do textur používán rozsah $x, y \in \langle 0; 1 \rangle$, avšak souřadnice na obrazovce jsou v rozsahu $x, y \in \langle -1; 1 \rangle$. Tyto transformace převádí souřadnice mezi těmito dvěma rozsahy.

6.2.7 Perspektiva s jedním úběžníkem

Použití funkce `transform.perspective`

Objektové rozhraní:

```
import transform
transform.identity() \
    .perspective(.5) \
    .apply_to(vector)
```

Tato transformace je aplikací `HYPERSTACKVIEW` používána pro zobrazování snímků v perspektivě. Po provedení této transformace je nutné následně vektory po provedení transformace vydělit jejich posledním prvkem.

Parametrem této transformace je koeficient f (zpravidla $f \in \langle 0; \infty \rangle$), který určuje míru zkreslení. Pokud $f = 0$, výsledkem je jednotková matice (tj. žádné zkreslení). Vliv hodnoty f na výsledek je znázorněn na obrázku 6.10.

Je použita následující transformační matice:

$$\mathbf{M} = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & f & 1 \end{array} \right)$$

Je možné také s pomocí složitějších výpočtů vytvářet transformační matice, které slouží pro vykreslování v perspektivě s určeným zorným úhlem a směrem a středem pohledu určeným vektorem. Z důvodu, že takové zobrazení není v aplikaci `HYPERSTACKVIEW` potřebné, tento výpočet není knihovnou podporován.

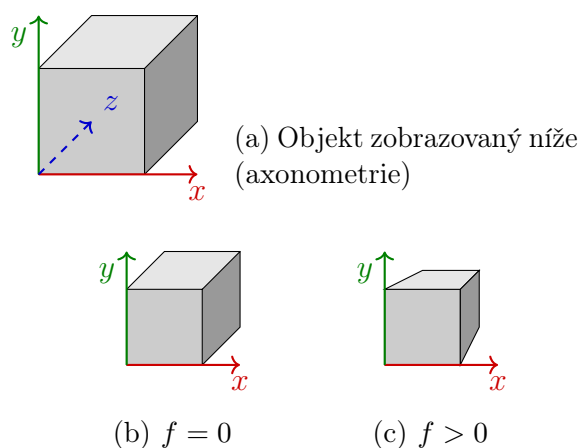
6.3 Funkcionální rozhraní

Knihovna definuje pro každou výše popsanou transformaci funkci. Jejimi parametry jsou parametry transformace (viz výše) a volitelný parametr `apply_to`. Pokud není hodnota parametru `apply_to` určena, funkce vrátí instanci třídy `Transform`, která reprezentuje transformační matici. Pokud je hodnota parametru `apply_to` vektorem (resp. jednorozměrným polem o třech nebo čtyřech prvcích), funkce provede transformaci přímo na daném vektoru. Pokud je hodnota parametru `apply_to` maticí (tj. dvojrozměrným polem nebo instancí třídy `Transform`), funkce vrátí součin matice, kterou by vrátila, pokud by

$$\mathbf{A} = \left(\begin{array}{ccc|c} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ \hline p_1 & p_2 & p_3 & 1 \end{array} \right)$$

$$\mathbf{A} = \left(\begin{array}{ccc|c} \text{škálování, rotace, ...} & & & \\ \hline \boxed{\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array}} & \boxed{\begin{array}{c} t_x \\ t_y \\ t_z \end{array}} & \text{posunutí} & \\ \hline \boxed{\begin{array}{ccc} p_1 & p_2 & p_3 \end{array}} & 1 & & \\ \text{perspektiva} & & & \end{array} \right)$$

Obrázek 6.1: Transformační matici o velikosti 4×4 můžeme rozdělit na části, podle toho, jak jsou ovlivňovány jednotlivými transformacemi.



Obrázek 6.10: Vliv koeficientu f na zobrazení v perspektivě.

`apply_to` nebylo určeno, s maticí předanou funkcí v `apply_to`. Tato pravidla jsou shrnuta v následující tabulce:

Použití funkce	Výsledek
$funkce(\dots)$	$= \mathbf{T}$
$funkce(\dots, \text{apply_to}=\mathbf{M})$	$= \mathbf{T} \cdot \mathbf{M}$
$funkce(\dots, \text{apply_to}=b)$	$= \mathbf{T} \cdot v$

\mathbf{T} je transformační matice, která odpovídá prováděné transformaci.

\mathbf{M} je libovolná jiná transformační matice. v je libovolný vektor.

Tabulka 6.1: Použití parametru `apply_to` některých funkcí knihovny.

Další funkcí definovanou knihovnou, která není dostupná v objektovém rozhraní je `identity`. Její návratovou hodnotou je nová instance třídy `Transform`, která odpovídá jednotkové matici.

6.4 Objektové rozhraní

Pro snadnější zápis transformací v kódu bylo umožněno využití metod třídy `Transform` pro skládání transformací. Výčet těchto metod se nachází v sekci 6.2.

Voláním metody na instanci třídy `Transform` získáme novou instanci třídy `Transform`, která odpovídá kombinaci původní transformace s nově provedenou transformací. Původní instance tím není nijak ovlivněna. Díky tomu je možné zřetěžit transformace:

```
import transform

translate_then_rotate = transform.Transform() \
    .translate([10, 2, 3]) \
    .rotate_about([1, 0, 0], 3.14)
```

Obrázek 6.11: Ukázka použití objektového rozhraní pro řetězení transformací

Transformace je možné také řetěžit pomocí metod `apply_to` a `concat`. První z nich aplikuje transformaci, na které je volána, na jinou transformační matici nebo vektor. Druhá naopak na transformaci, na které je volána aplikuje jinou transformaci.

Třída `Transform` je podtřídou `numpy.ndarray` (z knihovny *NumPy*) a je možné její instance používat na místech, kde je očekávána instance třídy `numpy.ndarray`. Je také možné používat funkci pro maticový součin `numpy.dot` a operátor `@` pro skládání transformací a transformaci vektorů.

6 Závěr

V práci jsem popsal aplikaci pro zobrazování 3D+t dat, kterou jsem vytvořil.

Je provedena rešerše existujících nástrojů pro zobrazování mikroskopických 3D+t dat zaměřená zejména na nástroje prostředí IMAGEJ, které je nejspíše nejpoužívanějším v oblasti kvantitativní analýzy laboratorních obrazových dat.

U aplikace HYPERSTACKVIEW, kterou jsem vytvořil, jsem kladl silný důraz na možnost vizualizace dat v obecném pohledu na data v reálném čase s možností interaktivního a intuitivního ovládání. Je v ní možné načítat soubory v několika variantách formátu TIFF používaných pro uchovávání 3D a 3D+t dat pro urychlení načítání je použita technika mapování souboru do paměti.

Snímky jsou vykreslovány pomocí projekce maximálních („max projekce“) a průměrných hodnot, nebo jako řez libovolnou rovinou. Souběžně nebo namísto snímku je možné zobrazit binární data popisující přítomnost jednotlivých objektů na snímku. Také je možné vybírat data ke zobrazení podle čísla objektů, které jsou zadávány přímo nebo s použitím externí aplikace. To umožňuje například najít objekt ve snímku, který odpovídá určitému řádku v tabulce příznaků objektů. Vykreslování probíhá s použitím hardwarové akcelerace na GPU s použitím rozhraní OpenGL.

V aplikaci je možné upravovat řadu parametrů pomocí grafického uživatelského rozhraní a klávesových zkratk, zejména: otočení a zvětšení snímku, perspektivní zkreslení, barvu zobrazení, jas, kontrast, parametr γ , rozsah vstupních hodnot (s možností automatického nastavení), kontrast a ostrost vykreslování objektů apod.

Je umožněno procházení časové sekvence snímků a změna pohledu na snímek pomocí klávesových zkratk a ovládání myši. Také je možné spustit animaci, na které je procházena časová sekvence snímků, mění se pohled na snímek, nebo jsou prováděny obě akce zároveň.

Aplikaci jsem zveřejnil na veřejně dostupném repositáři PYPI. Je možné ji spustit na několika platformách. Byla otestována na operačních systémech MICROSOFT WINDOWS a LINUX. Aplikace je prakticky používána v Ústavu teorie informace a automatizace AV ČR při analýze časosběrných záznamů vývoje oocytů pořízených light-sheet mikroskopem.

Seznam obrázků

1.1	Grafické znázornění možností zaznamenání snímků s různým počtem rozměrů.	4
4.1	Okno konfigurace parametrů nástroje <i>Z Project</i> v aplikaci IMAGEJ	12
4.2	Okno konfigurace parametrů nástroje <i>3D Project</i> v aplikaci IMAGEJ	15
4.4	Okno nástroje <i>BigDataViewer</i> dostupného v aplikaci IMAGEJ	15
4.3	Projekce maximální intenzity třírozměrného mikroskopického snímku s 13 rovinami vytvořená pomocí nástroje <i>3D Projection</i> aplikace IMAGEJ	16
4.5	Okno nástroje <i>3D Surface Plot</i> dostupného v aplikaci IMAGEJ	17
5.1	Hlavní okno aplikace. Je zobrazen jeden mikroskopický snímek z časové sekvence v projekci maximální intenzity. (Hlavní nabídka aplikace je skryta.)	20
5.2	Hlavní okno aplikace. Je zobrazen tentýž snímek s překrytím znázorněním objektu, které bylo získáno pomocí segmentace snímku.	21
5.3	Schématické znázornění prostoru snímku a obrazovky	25
5.4	Vzdálenost bodu objektu od průmětny	26
5.5	Vliv parametru γ na průběh funkce $f(x)$	28
5.6	Změna jasu pixelu může být nepravidelná, pokud dojde k přechodu na celá čísla před výpočtem barevné hodnoty pixelů. Pro porovnání jsou vykresleny i hodnoty, které jsou získány bez převodu na celá čísla. Na spodním grafu je zakreslen rozdíl těchto dvou křivek.	29
5.7	Strom dědičnosti tříd používaných pro posuvníky.	35
5.8	Přiřazení funkcí klávesám alfanumerické klávesnice	36
5.9	Přiřazení funkcí klávesám numerické klávesnice	36
6.2	Použití funkce <code>transform.translate</code>	40
6.3	Použití funkce <code>transform.scale</code>	40
6.4	Použití funkce <code>transform.rotate_about</code>	41
6.5	Použití funkce <code>transform.swap_axes</code>	42
6.6	Použití funkce <code>transform.drop_axis</code>	42
6.7	Použití funkce <code>transform.texture_to_screen</code>	43
6.8	Použití funkce <code>transform.screen_to_texture</code>	43

6.9	Použití funkce <code>transform.perspective</code>	44
6.1	Transformační matici o velikosti 4×4 můžeme rozdělit na části, podle toho, jak jsou ovlivňovány jednotlivými transformacemi.	45
6.10	Vliv koeficientu f na zobrazení v perspektivě.	45
6.11	Ukázka použití objektového rozhraní pro řetězení transformací	47

Seznam tabulek

4.1	Matematické operace, které je možné v aplikaci IMAGEJ provádět nad tří- a vícerozměrnými snímky při projekci nástrojem <i>Z-project</i>	13
5.1	Parametry převodu intenzity na barevnou hodnotu.	27
5.2	Klávesové zkratky, které se mění v závislosti na používaném operačním systému	36
5.3	Klávesové zkratky aplikace HYPERSTACKVIEW	37
6.1	Použití parametru <code>apply_to</code> některých funkcí knihovny.	46

Bibliografie

- [1] Intel Corporation. *Intel® 64 and IA-32 Architectures Software Developer's Manual, Combined Volumes: 1, 2A, 2B, 2C, 3D, 3A, 3B, 3C, 3D and 4*. 2017. URL: <http://www.intel.com/design/literature.htm>.
- [2] The Open Microscopy Environment. Lis. 2021. URL: <https://docs.openmicroscopy.org/ome-model/5.6.3/ome-tiff/>.
- [3] Ilya G. Goldberg et al. „The Open Microscopy Environment (OME) Data Model and XML file: open tools for informatics and quantitative analysis in biological imaging“. In: *Genome Biology* 6.5 (květ. 2005), R47. ISSN: 1474-760X. DOI: 10.1186/gb-2005-6-5-r47. URL: <https://doi.org/10.1186/gb-2005-6-5-r47>.
- [4] Ilya G. Goldberg et al. „The Open Microscopy Environment (OME) Data Model and XML file: open tools for informatics and quantitative analysis in biological imaging“. In: *Genome Biology* 6.5 (květ. 2005), R47. ISSN: 1474-760X. DOI: 10.1186/gb-2005-6-5-r47. URL: <https://doi.org/10.1186/gb-2005-6-5-r47>.
- [5] Jan Huiskens et al. „Optical Sectioning Deep Inside Live Embryos by Selective Plane Illumination Microscopy“. In: *Science* 305.5686 (2004), s. 1007–1009. ISSN: 0036-8075. DOI: 10.1126/science.1100035. eprint: <https://science.sciencemag.org/content/305/5686/1007.full.pdf>. URL: <https://science.sciencemag.org/content/305/5686/1007>.
- [6] Michael Castle a vývojáři aplikace ImageJ. *ImageJ. Projector.java*. ver. 1.53. Lis. 2021. URL: <https://imagej.nih.gov/ij/source/ij/plugin/Projector.java>.
- [7] Patric Kelly. *ImageJ. ZProjector.java*. ver. 1.53. Lis. 2021. URL: <https://imagej.nih.gov/ij/source/ij/plugin/ZProjector.java>.
- [8] Tobias Pietzsch et al. „BigDataViewer: visualization and processing for large image data sets“. In: *Nature Methods* 12.6 (čvn. 2015), s. 481–483. ISSN: 1548-7105. DOI: 10.1038/nmeth.3392. URL: <https://doi.org/10.1038/nmeth.3392>.
- [9] Mark a Kurt AKELEY SEGAL. *The OpenGL Graphics System: A Specification (Version 4.6 (Core Profile) - October 22, 2019)*. 2018. URL: <https://www.khronos.org/registry/OpenGL/specs/gl/glspec46.core.pdf>.

- [10] Johannes Schindelin et al. „Fiji: an open-source platform for biological-image analysis“. In: *Nature Methods* 9.7 (čvc. 2012), s. 676–682. ISSN: 1548-7105. DOI: 10.1038/nmeth.2019. URL: <https://doi.org/10.1038/nmeth.2019>.
- [11] J.W. Wallis et al. „Three-dimensional display in nuclear medicine“. In: *IEEE Transactions on Medical Imaging* 8.4 (1989), s. 297–230. DOI: 10.1109/42.41482.
- [12] Wikipedia contributors. *Hierarchical Data Format* — *Wikipedia, The Free Encyclopedia*. Lis. 2021. URL: https://en.wikipedia.org/w/index.php?title=Hierarchical_Data_Format&oldid=1049571370.
- [13] Jiří Žára. *Moderní počítačová grafika*. 2004. 625 s. ISBN: 80-251-0454-0.

Seznam použitých zkratek

2D

Vysvětlení zkratky: dvojrozměrný

3D

Vysvětlení zkratky: třírozměrný

3D+t

Vysvětlení zkratky: časová sekvence třírozměrných snímků

CT

(z anglického *Computed tomography*)

Vysvětlení zkratky: výpočetní tomografie

GPU

(z anglického *Graphics processing unit*)

Vysvětlení zkratky: grafická krata počítače

GUI

(z anglického *Graphical user interface*)

Vysvětlení zkratky: grafické uživatelské rozhraní

IEEE

(z anglického *Institute of Electrical and Electronics Engineers*)

Vysvětlení zkratky: mezinárodní organizace, která provádí kromě jiného i standardizaci

Této práce se týká zejména standard IEEE 754, který popisuje formát pro ukládání čísel s plovoucí řádovou čárkou a jejich zpracovávání.

HDF5

(z anglického *Hierarchical data format 5*)

Vysvětlení zkratky: formát pro ukládání hierarchických dat, mimo jiné i rozsáhlých polí čísel

MRI

(z anglického *Magnetic resonance imaging*)

Vysvětlení zkratky: magnetická rezonance

OME

Vysvětlení zkratky: *Open microscopy environment*

LUT(z anglického *Lookup table*)**Vysvětlení zkratky:** vyhledávací tabulka

Vyhledávací tabulky jsou používány pro převod abstraktních číselných hodnot na barvu. Tabulka je polem, které obsahuje barevné hodnoty seřazené podle jim náležící číselné hodnoty. Přístupem do tohoto pole je získána příslušná barevná hodnota k vykreslení.

RAM(z anglického *Random-access memory*)**Vysvětlení zkratky:** operační paměť počítače**RGB**(z anglického *Red, green, blue*)

Vysvětlení zkratky: barevný model s aditivním mícháním barev, kde jsou barvy reprezentovány třemi hodnotami určujícími zastoupení červené, zelené a modré barvy.

TIFF(z anglického *Tag image file format*)**Vysvětlení zkratky:** formát souborů pro ukládání obrazových dat**x86_64**(též *Intel® 64*, **x64** nebo *AMD 64*)

Vysvětlení zkratky: V této práci použito pro označování architektury procesorů Intel® 64 a kompatibilních, tj. např. architektura AMD 64, případně pro označování jejich instrukční sady.

V souladu se zákonem č. 121/2000 Sb. (autorský zákon) souhlasím s použitím tohoto díla pro potřeby školy podle § 35, odst. 3 tohoto zákona.

V Havlíčkově Brodě
dne 13. 4. 2022

Jiří Wolker