

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Root Tracker

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Root Tracker

Root Tracker

Autor: Tadeáš Fryčák

Škola: Gymnázium Jana Opletala, Litovel, Opletalova 189

Kraj: Olomoucký

Konzultanti: Mgr. Jan Humplík, Ph.D.

RNDr. Tomáš Fürst, Ph.D.

Litovel 2023

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupnění 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 Litovli dne: _____

Tadeáš Fryčák

Poděkování

Rád bych poděkoval svým kolegům, Mgr. Janu Humplíku, Ph.D. a RNDr. Tomáši Fürstovi, Ph.D., za poskytnutí příležitosti účastnit se tohoto projektu a za poskytování cenných rad.

Anotace

Tato práce popisuje zhotovení deterministického modelu pro sledování kořenů bez nutnosti rozsáhlých anotací dat, jako je tomu například u konvolučních neuronových sítí. Pro účely této práce jsem vyvinul nezávislé algoritmy počítačového vidění detekující kořeny z obrazu, který je pravidelně snímán DSLR fotoaparátem, případně skenován skenerem ve velmi vysokém rozlišení. Následně jsem aplikoval vlastní algoritmus, který je schopný automaticky spojovat, sledovat a predikovat jednotlivé kořeny i pro složitější případy, kdy se kořeny kříží, nebo částečně chybí.

Klíčová slova

Root Tracker; AI; obrazová analýza; deterministický model; kořeny

Annotation

This thesis describes the development of a deterministic model for tracking roots without the need for extensive data annotations, as is necessary, for example, with convolutional neural networks. For the purposes of this thesis, I have developed two independent computer vision algorithms that detect parts of roots from images regularly taken with a DSLR camera or a scanner at very high resolution. Subsequently, I devised an algorithm capable of automatically merging, tracking, and predicting individual roots, even when they overlap with the roots of other plants present in the same image.

Keywords

Root Tracker; AI; Computer Vision; Deterministic Model; Roots

Obsah

Úvod	8
1 Definice pojmů	9
1.1 Programovací jazyk Python	9
1.2 Moduly pro snímání	10
1.2.1 In vitro	10
1.2.2 Rhizotron	11
1.3 Metody počítačového vidění	12
1.3.1 Prahování	12
1.3.2 Skeletonizace	13
2 Pořízení obrazu	14
2.1 Rhizotron	14
2.2 In vitro	15
2.2.1 3D model	15
3 Moduly programu	18
3.1 ROI	18
3.1.1 Manuální	19
3.1.2 Ořez pomocí prahování	20
3.1.3 Ořez pomocí mediánu	21
3.1.4 Další funkce	22
3.2 Detekce počátků	23

3.2.1	Pomocí detekce středu zelené plochy	23
3.2.2	Pomocí nejvyššího možného bodu	24
3.3	Předzpracování obrazu	25
3.4	Detekce kořenů	28
3.4.1	Detekce pomocí prahování	28
3.4.2	Detekce pomocí SNR	31
3.5	Predikce chybějících kořenů	35
3.6	Další funkce	36
3.6.1	Časová informace	36
3.6.2	Automatické čtení čárových kódů	36
3.6.3	Paralelní výpočet	36
4	Vyhodnocení	37
4.1	Extrakce vlastností	37
4.2	RGR	38
4.3	Logistická křivka	39
	Závěr	41
	Seznam zdrojů	42
	Seznam zkratk	43
	Seznam obrázků	45
	Seznam rovnic	46
	Seznam příloh	47

Úvod

Tato práce se zabývá dosud nedořešenou problematikou – automatickým sledováním růstu kořenů s využitím inovativní algoritmizace pro sledování a vyhodnocování složitých křížících se kořenových systémů pro různá biologická prostředí.

Cílem práce je navrhnout standardní snímací sestavu, vytvořit modulární a konfigurovatelný deterministický model počítačového vidění v programovacím jazyce Python pro detekci celého kořenového systému a extrakce jeho parametrů (jako například tloušťka, délka, počet vedlejších kořenů, úhly kořenů, atp.) bez nutnosti rozsáhlých anotací dat, jako je tomu například u konvolučních neuronových sítí. Následně se práce zaměřuje na výběr korektních metod pro vyhodnocení (RGR, logistická funkce).

Využití a aplikaci Root Trackeru a jeho algoritmů budu demonstrovat na dvou typech snímků:

- Na rostlinách zasazených v substrátu v tzv. "**rhizotronech**", kde mohou být části kořenů skryté za substrátem a je nutné je predikovat.
- Na rostlinách zasazených v čirém živném médiu (hnojivo a nosné médium agar) v modulech nazývaných tzv. "**in vitro**".

Kapitola 1

Definice pojmů

1.1 Programovací jazyk Python

Python je jeden z moderních programovacích jazyků, který je:

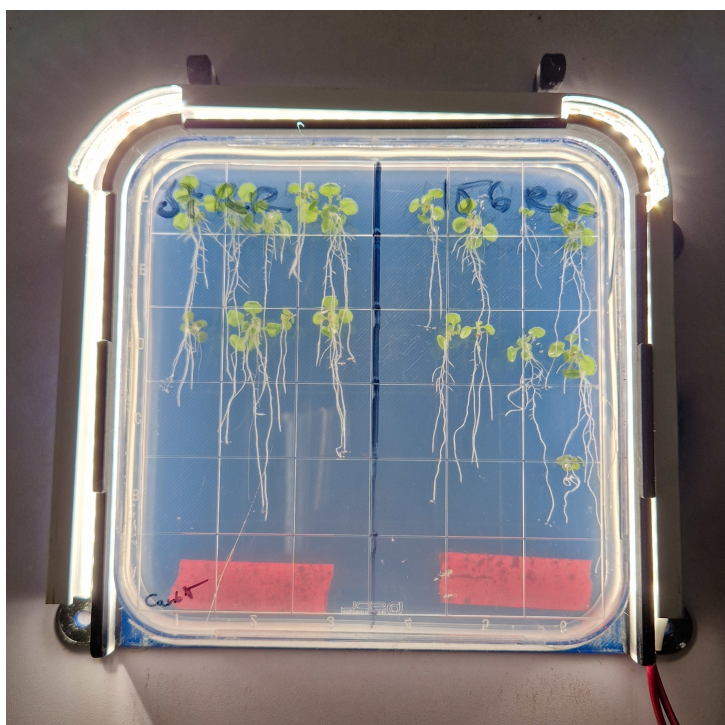
- Univerzální – široce použitelný, je možné ho využít například pro backendový vývoj webových stránek, vývoj počítačových programů, či pro matematické výpočty a predikce
- Rychle naučitelný, skvělý pro výuku
- Jednoduše instalovatelný
- Velmi rozšířený[1]

1.2 Moduly pro snímání

1.2.1 In vitro

In vitro modul může být například pomyslná krabička vytvořená z dvou Petriho misek, ve které se nachází modelová rostlina (například *Arabidopsis*) zasazená v živném médiu složeného z tekutého hnojiva (například Krystalonu) a nosného média agaru. Živné médium je čiré až mírně nažloutlé, proto je velmi jednoduché skrze něj snímat kořeny, které v něm rostou.

Jelikož tyto moduly je nutné používat jen jednorázově, není ekonomicky možné zasazovat do jednoho in vitro modulu pouze jednu rostlinu. Proto se v jednom modulu většinou nachází 6-20 modelových rostlin najednou.



Obrázek 1.1: Ukázka in vitro modulu

1.2.2 Rhizotron

Rhizotron je laboratorně vytvořené podzemí typicky ve tvaru kvádru, které slouží pro studium kořenového systému. Jedna ze stěn rhizotronu je složená například ze skla, nebo plexiskla. Díky její průhlednosti je poté možné kořeny sledovat v čase. Aby nedošlo k reakci kořenů na světlo, je nutné průhlednou část rhizotronu při nesnímání zakrývat neprůhledným materiálem. Aby bylo zajištěno, že kořeny rostou co nejbližší u průhledné části, jsou rhizotrony umístěné v držácích ve sklonu mezi 15°- 45°. Oproti in vitro modulům rhizotron zásadní výhodou, je možné jej využívat opakovaně. [2]



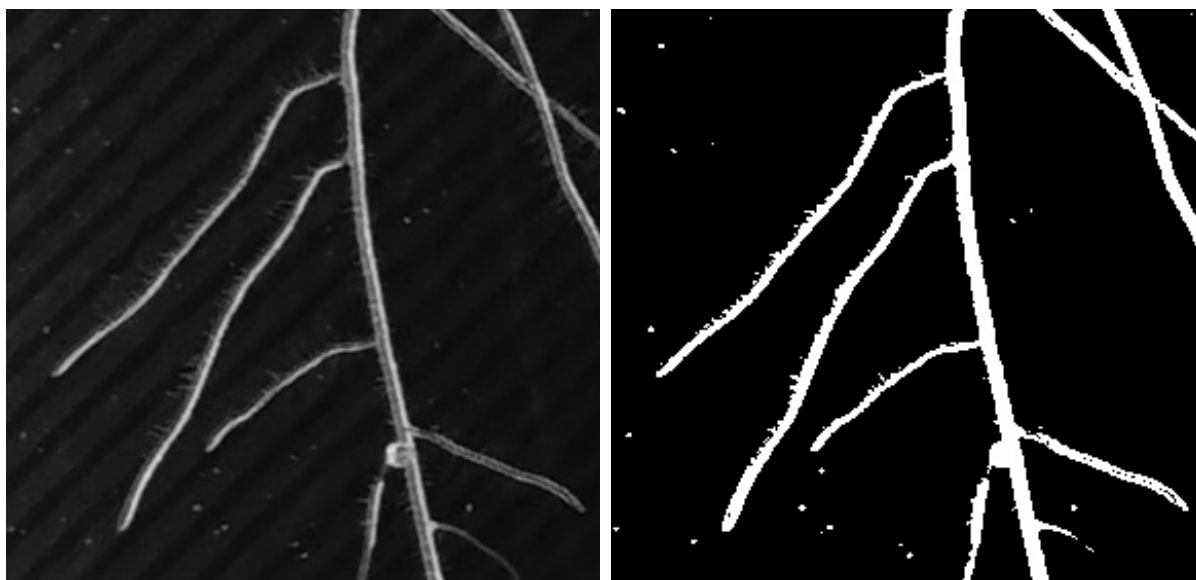
Obrázek 1.2: Ukázka rhizotronu

1.3 Metody počítačového vidění

1.3.1 Prahování

Prahování, neboli binarizace, je proces, kdy jsou hodnoty všech pixelů změněny na hodnotu A (v zpracování obrazu typicky plně černá barva) v případě, že je původní jas pixelu menší než zvolená prahovací hodnota, nebo hodnotu B (ve zpracování obrazu typicky plně bílá barva), pokud je původní hodnota pixelu větší nebo rovna zvolené prahovací hodnotě.

Klíčové při binarizaci je správné zvolení prahové hodnoty, tu lze zadat fixně, nebo počítat automaticky. V případě automatického výpočtu se často používá například Otsuova metoda. Mezi oblíbené patří také výpočet hodnoty z průměrné nebo medianové jasové hodnoty celého obrázku.



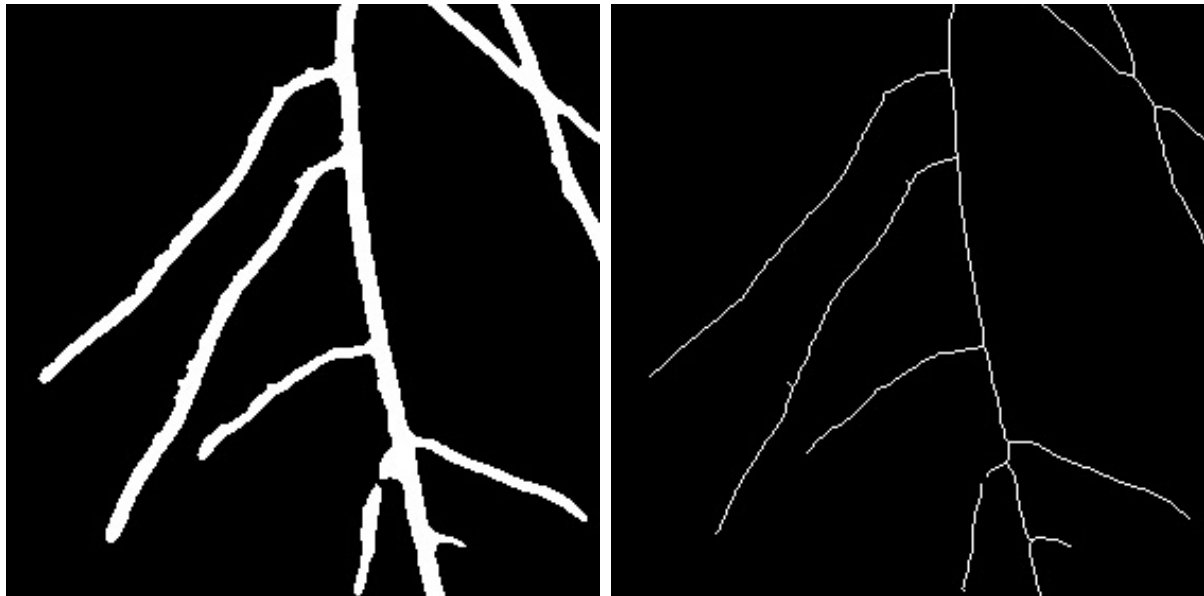
(a) Původní obraz

(b) Binarizovaný obraz

Obrázek 1.3: Příklad binarizace

1.3.2 Skeletonizace

Skeletonizace je v digitálním zpracování obrazu proces, při kterém je binarizovaný obraz s objektem o různé tloušťce přetransformován na reprezentaci, která má šířku přesně 1 px a nachází se uprostřed binarizovaného obrazu. Tato transformace zachovává veškerá spojení. [3]



(a) Binarizovaný obraz

(b) Skeleton binarizovaného obrazu

Obrázek 1.4: Příklad skeletonizace

Kapitola 2

Pořizení obrazu

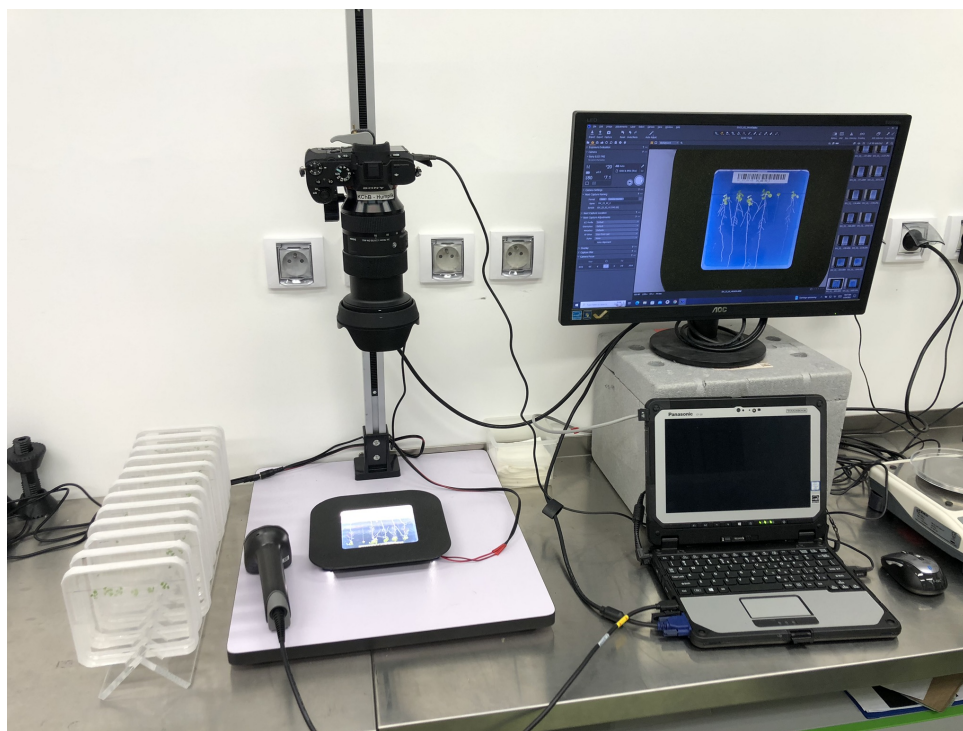
Pořizení obrazu, všeobecně známé také jako "image acquisition", je první část počítačového vidění – zabývá se procesem správného zachycení požadovaného jevu pro následné zpracování. Kvalita pořizení obrazu je velmi důležitá pro pozdější zpracování. Zejména velké potíže mohou způsobit různé nedokonalosti v obraze, které vznikají použitím nevhodné metody pro pořizení obrazu.

2.1 Rhizotron

Při pravidelném snímání rhizotronů jsme se rozhodli využít v laboratoři dostupného bezkontaktního skeneru [Fujitsu ScanSnap SV600](#) umístěného v krytém prostředí pro minimalizaci odlesků od průhledné části rhizotronu.

2.2 In vitro

Při snímání in vitro modulů jsme použili dostupnou a zároveň nejjednodušší a nejrychlejší možnost pro iterativní snímání – fotografování pomocí DSLR umístěné ve stativu s možností přenosu dat do počítače.

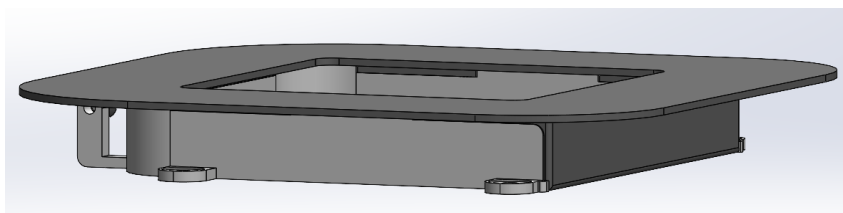


Obrázek 2.1: Ukázka sestavy pro fotografování in vitro

2.2.1 3D model

Abychom mohli pořizovat snímky co nejvyšší kvality, rozhodli jsme se využít výhod 3D tisku a průsvitnosti živného média. V programu Solidworks jsem proto navrhl 3D model držáku pro in vitro moduly. Při návrhu jsem kladl důraz na to, aby:

- model byl jednoduše tisknutelný na jakékoli běžné 3D tiskárně
- držák na in vitro modulu bylo možné přišroubovat pevně k podložce, a tím zajistit fixní pozici
- jednoduchou manipulovatelnost s in vitro modulem
- bylo možné vhodně nasvítit kořeny in vitro pomocí LED pásku umístěného okolo vnější hrany držáku

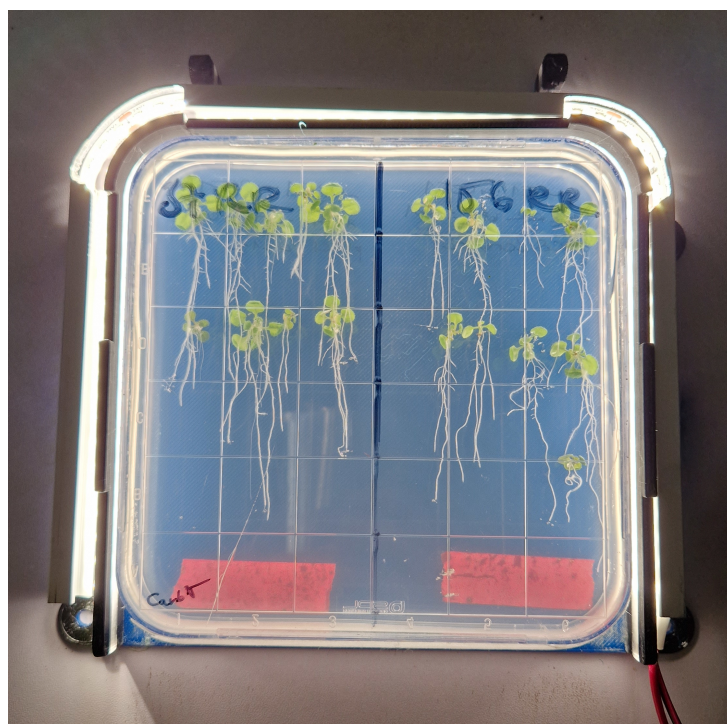


Obrázek 2.2: Ukázka sestavy pro fotografování kořenů in vitro

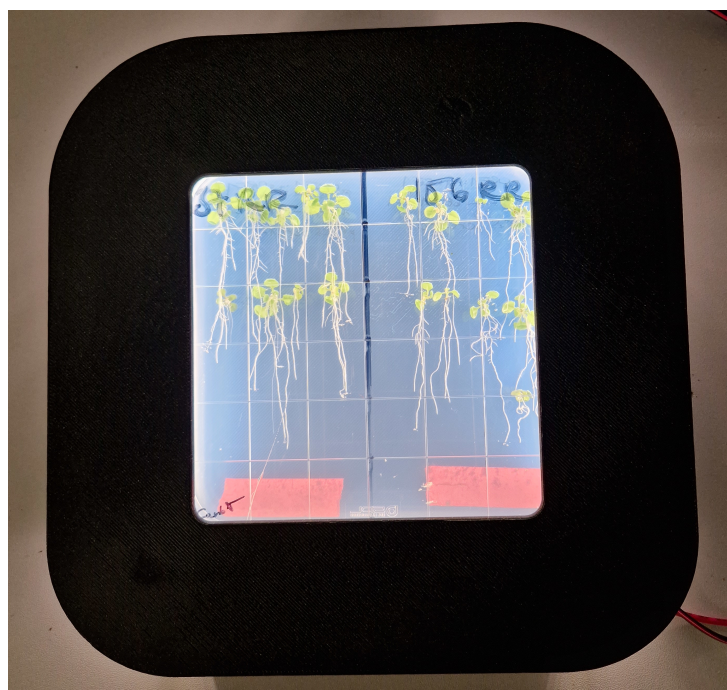
Volba filamentu

Abychom dokázali co nejlépe nasvítit rostliny a kořenový systém v živném médiu a co nejvíce zredukovali odrazivost světla od krytu, rozhodli jsme se vytisknout model ze tří barev PETG filamentu v následujícím pořadí:

- **Modrá** – z této barvy je vytisknuté pozadí pro průhledný in vitro modul, slouží pro jednoduchou filtraci pozadí z obrazu, ideálně je tato vrstva vytisknutá za použití možnosti "ironing", která vyhladí strukturu povrchu vytvořenou při 3D tisku
- **Černá** – z této barvy je vytisknutá první třetina výšky in vitro modulu, redukuje nasvícení spodní vrstvy krytu in vitro modulu, a tím omezuje zobrazení možných rýh a vrubů v lesklém povrchu krytu ve fotografii, také brání nasvícení a zvýraznění vodních kapek nacházejících uvnitř in vitro modulu.
- **Čirá** – z této barvy je vytisknutá druhá třetina výšky in vitro modulu, umožňuje průchod světla LED umístěných okolo in vitro modulu
- **Černá** – z této barvy je vytisknutá třetí třetina výšky in vitro modulu, opět zabraňuje nasvícení části krytu, které může být poškozené rýhami



Obrázek 2.3: Ukázka sestavy pro fotografování in vitro bez krytu



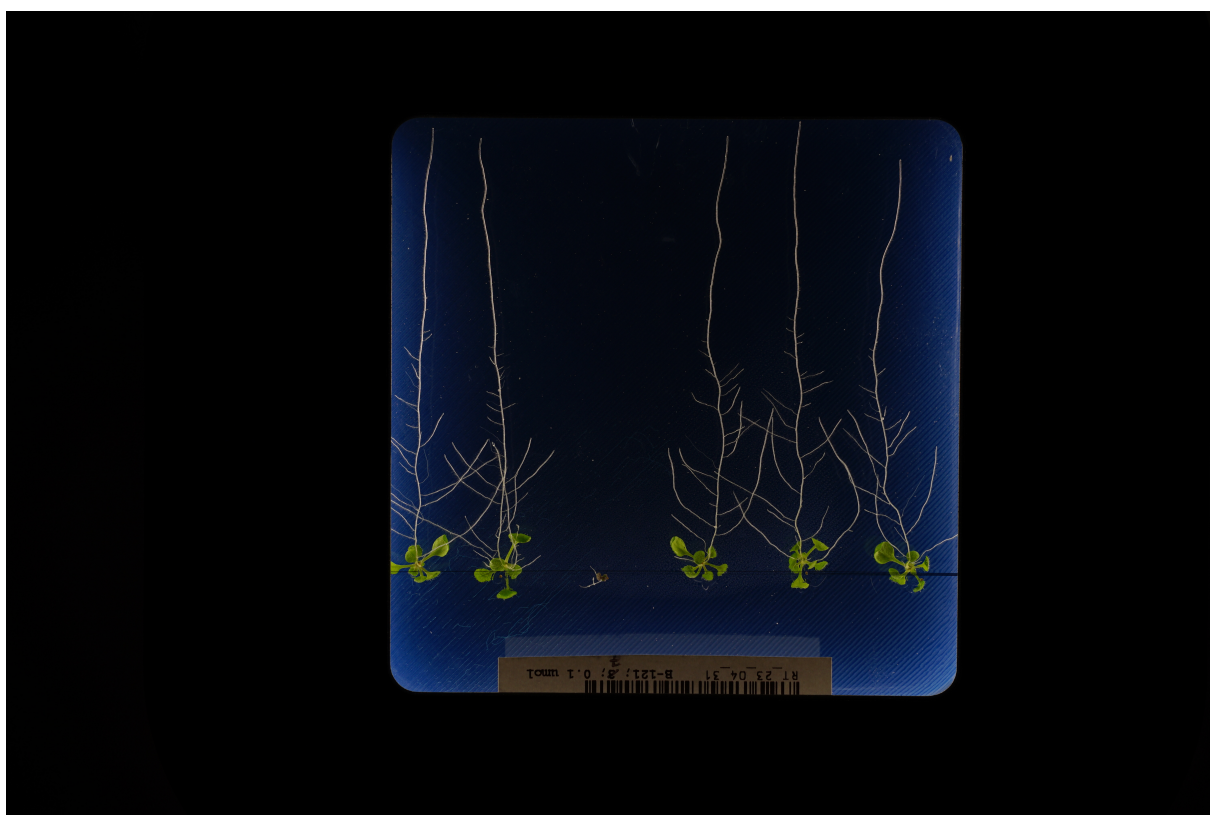
Obrázek 2.4: Ukázka sestavy pro fotografování in vitro s krytem

Kapitola 3

Moduly programu

3.1 ROI

Pro správnou analýzu obrazu je nutné provést rozměrovou standardizaci. Pro tento účel slouží ořezání a rotace, neboli ROI (z anglického Region of Interest). Root Tracker nabízí několik možností získání ROI:

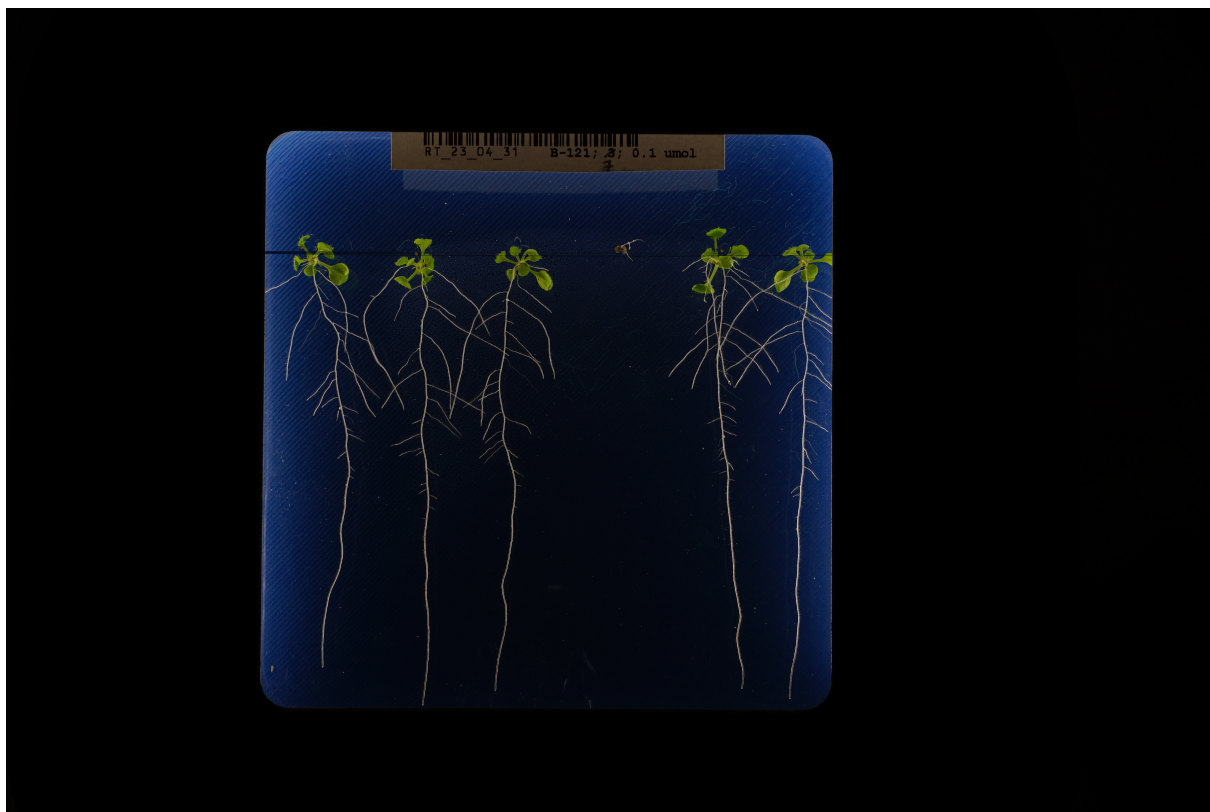


Obrázek 3.1: Ukázkový originální vstupní obraz in vitro

3.1.1 Manuální

Základní metodou je opakované ořezání snímků a rotace obrazu dle zadání uživatele pomocí konfiguračního souboru. Tuto metodu je možné použít i v kombinaci s dalšími automatickými metodami popsanými níže.

Nevýhodou použití manuální metody samostatně je nutnost pořizování všech snímků v totožné pozici vůči DSLR. Fixní upevnění je možné zajistit například pomocí připevnění 3D vytištěného držáku in vitro ke snímací aparatuře.



Obrázek 3.2: Ukázka manuální ROI s rotací 180° a ořezem 0 %

3.1.2 Ořez pomocí prahování

První automatickou možností je algoritmus automatického ořezání pomocí prahování. Nejprve se provede již zmíněné prahování v HSV barevném prostoru (v tomto případě pro modrou barvu – tato hodnota je zadaná uživatelem pomocí konfiguračního souboru).

Alternativně je možné využít i Otsu práhovací metody. Tato metoda ale funguje jen v případě, že snímaný objekt zabírá zhruba polovinu obrazu a kolem snímaného objektu se nachází stejné homogenní pozadí tmavšího odstínu.

Po aplikaci prahování se odstraní nejmenší detekované objekty pomocí iterativní morfologické operace – eroze. Morfologickou operaci jsem zvolil zejména z důvodu dosažení významně vyšší rychlosti výpočtu než při filtrování všech objektů dle velikosti pomocí *for* smyčky.

Následně se provede kontrola všech detekovaných objektů a vypočítá se poloha a velikost nejmenšího možného ohraničujícího obdélníku tak, aby veškeré detekované objekty byly uvnitř právě tohoto obdélníku. Poté se od obrazu oddělí okraje mimo tento obdélník, díky čemuž získáme oblast našeho zájmu.

Tento algoritmus má pro zajištění účinnosti celkem dva předpoklady:

- snímaný objekt musí mít pozadí nepatrně zabarvené do jakékoli barvy nevyskytující se ve snímaných rostlinách
- okolo snímaného objektu musí být homogenní plocha jiné barvy než barvy pozadí o šířce alespoň 1-2 cm



Obrázek 3.3: Ukázka automatického ořezání pomocí prahování pro in vitro

3.1.3 Ořez pomocí mediánu

V případě, že není možné zajistit efektivní fungování předchozí metody (například pokud barva pozadí koliduje s jinými barvami ve snímku), je možné využít druhou metodu, která má pouze jeden předpoklad – snímaný objekt musí zabírat většinu obrazu.

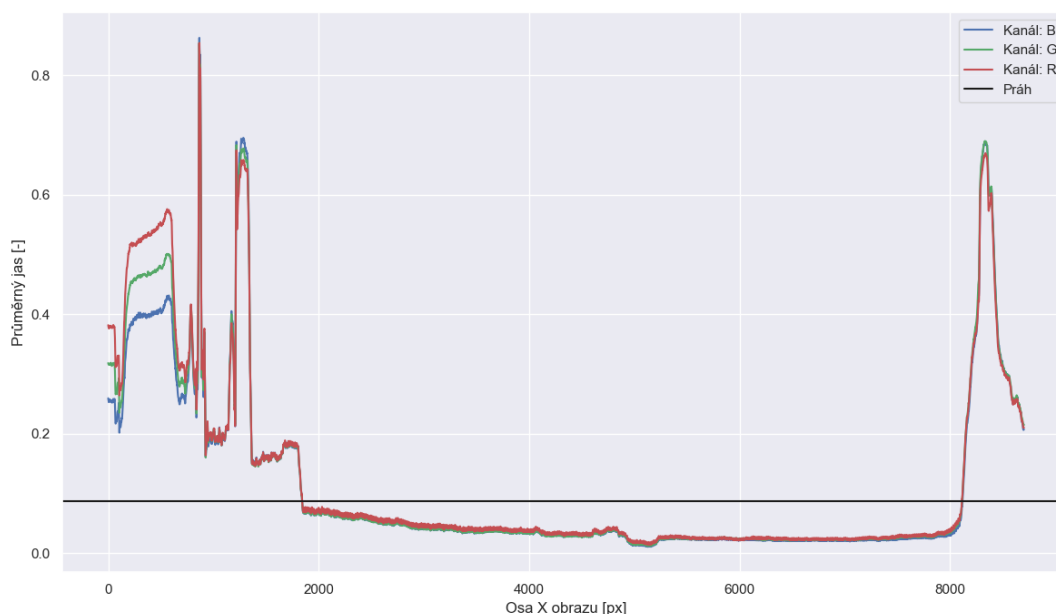
Algoritmus nejprve vypočítá průměrnou hodnotou jasu všech pixelů pro každý sloupec obrazu:

$$g(x) = \frac{1}{N} \sum_{y=1}^N f(x, y)$$

Rovnice 3.1: Průměrná hodnota dle sloupce obrázku

Poté Root Tracker spočítá medián všech průměrných hodnot, čímž získá jas pozadí. K této hodnotě přičte zadanou konstantu určující heterogenitu pozadí a tím určí prahovou hodnotu (v případě dokonale homogenního pozadí se konstanta rovná nule, v případě substrátového pozadí stačí použít při správné expozici konstantu například o hodnotě 1/20 bitové hloubky obrazu).

Z binarizované řady Root Tracker již jen extrahuje první a poslední sloupec, který má hodnotu nižší než práh.



Obrázek 3.4: Průměrný jas sloupců rhizotronu a určený práh

Stejnou operaci Root Tracker provede i pro všechny řádky a podle extrahovaných informací ořeže obraz vertikálně i horizontálně.



(a) Původní obraz

(b) Ořezaný obraz

Obrázek 3.5: Ukázka funkce algoritmu pro rhizotron

3.1.4 Další funkce

Automatická registrace

V případě, že Root Tracker zpracovává časovou sérii obrazů jednoho modulu (například s periodou dne po dobu jednoho týdne), automaticky si po ořezání provede registraci a sjednotí souřadnicovou soustavu všech obrazů oproti prvnímu snímku v časové sérii.

Registrace snímků je důležitá zejména pro zvýšení přesnosti až na jednotky pixelů a také v případě, že snímky nejsou fotografované pokaždé ze stejné pozice.

Z tohoto důvodu nejprve Root Tracker detekuje hrany ve všech obrazech pomocí Canny detektoru a následně pomocí konvoluce upravuje pozice obrazů tak, aby se co nejvíce hrany shodovaly, neboli měly stále stejnou pozici v čase.

Dodatečné odstranění okrajů

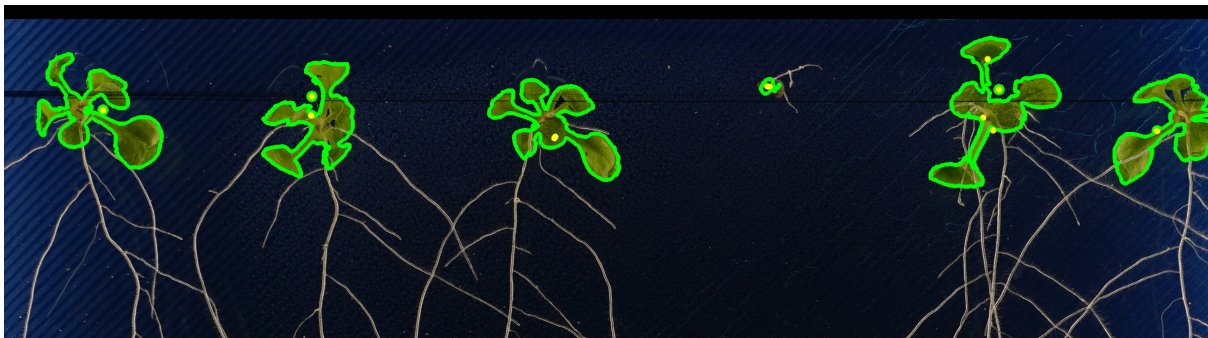
Po automatickém ořezání pozadí disponuje Root Tracker ještě možností provádějící poslední druh ořezání, jež slouží k manuálnímu odstřížení krajů snímaného objektu (nikoli celého obrazu jako v předchozích krocích).

3.2 Detekce počátků

Dle různé uživatelské konfigurace může snímaný model obsahovat odlišné počty rostlin zasazených v různých místech, proto nelze vždy předpokládat stejnou počáteční polohu, z které rostou kořeny. Root Tracker nabízí následující možnosti detekce počátků, ze kterých následně bude vyhledávat kořeny a vyhodnocovat statistiky:

3.2.1 Pomocí detekce středu zelené plochy

Tato metoda nejprve využije metody prahování na původním obrázku v HSV prostoru pro zelenou barvu (zelená barva je opět definovaná uživatelem). Následně vypočítá střed nejmenší možné kružnice opisující nejvzdálenější hranice daného objektu.



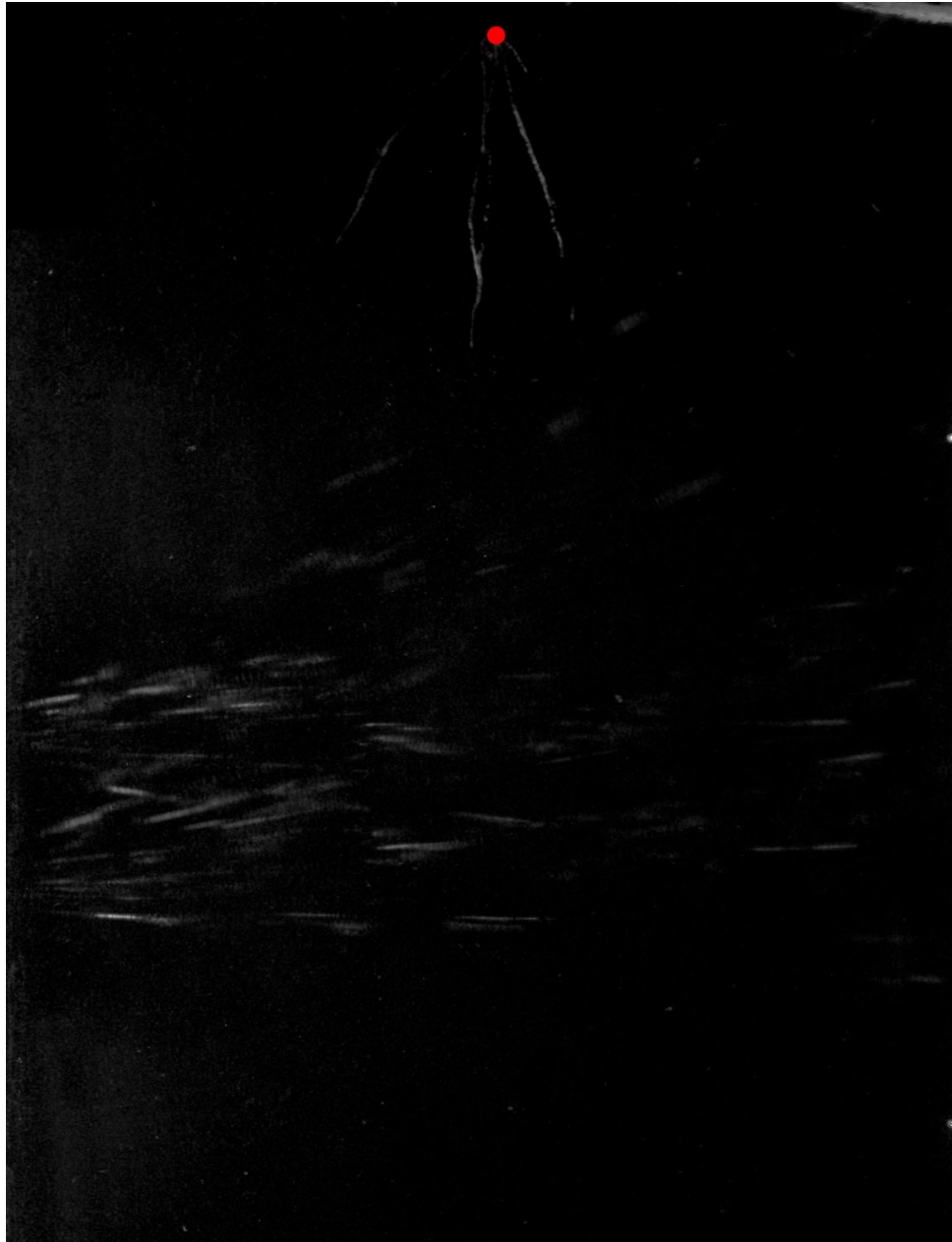
Obrázek 3.6: Ukázka detekce "nadzemní" části rostlin a výpočtu počátku

V některých případech se může stát, že algoritmus chybně detekuje konturu "nadzemní" části rostliny a rozdělí její geometrii například na dvě části. Proto jsem se rozhodl použít shlukovací metodu KMeans, jelikož počet rostlin v jednom modulu je vždy předem známý.

V případě, že program zpracovává časovou sérii snímků, pro zpřesnění výpočtu počátků použije klouzavý medián pro všechny počátky v čase.

3.2.2 Pomocí nejvyššího možného bodu

Počátek je možné získat i pomocí nejvyššího bodu z detekovaného kořenového systému (viz kapitola Detekce kořenů). Tuto metodu je vhodné použít jen v případě, že nadzemní zelená část rostliny není v obraze přítomná a tudíž její počátek nelze najít podle zelených listů.

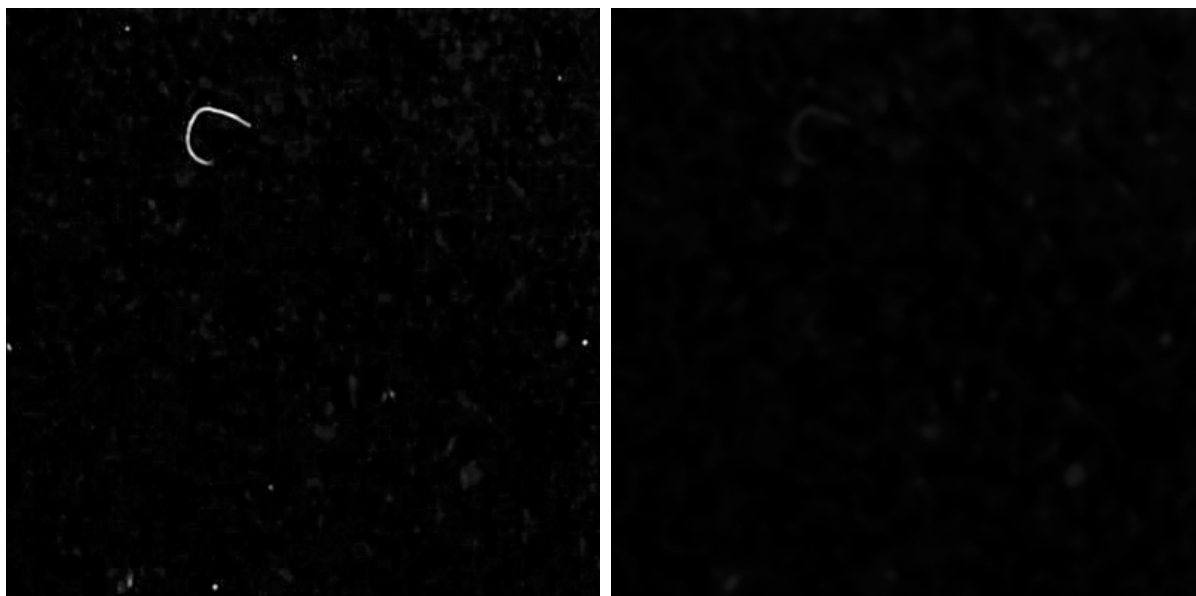


Obrázek 3.7: Ukázka detekce počátku pro rhizotron

3.3 Předzpracování obrazu

Pro kvalitní zpracování obrazu je vhodné obraz nejprve předzpracovat a odstranit tak jednoduše filtrovatelné druhy nečistot, případně odstranit některé deformace způsobené nedokonalým pořízením obrazu.

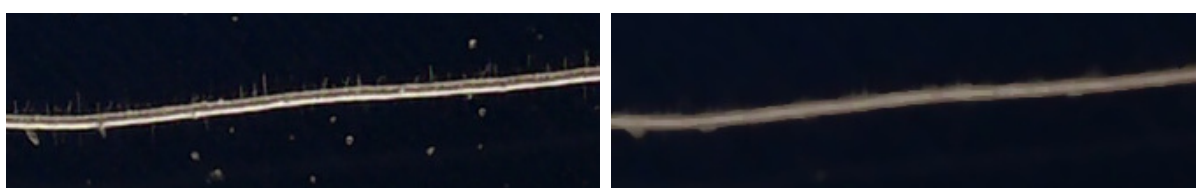
Pro tyto účely na obraz nejprve Root Tracker aplikuje mediánový filtr, jenž zajistí odstranění, nebo alespoň zmírnění některých nepříjemných jevů, jako například odlesky, třpytivé částice v substrátu, nebo rýhy ve skle/plexisklu.



(a) Původní obraz

(b) Předzpracovaný obraz

Obrázek 3.8: Ukázka předzpracování – filtrování nečistot z obrazu

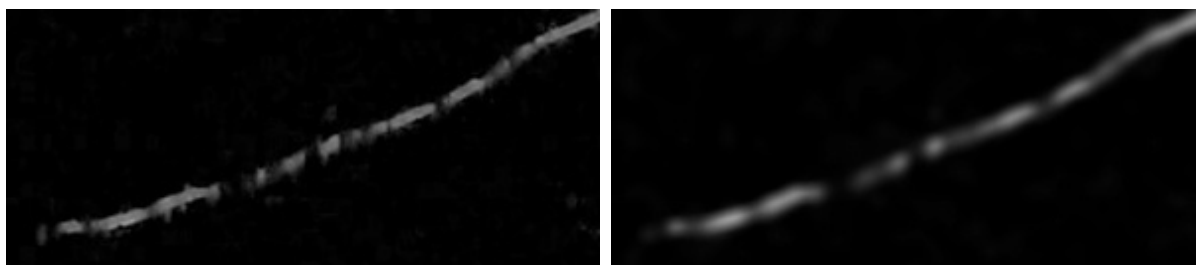


(a) Původní obraz

(b) Předzpracovaný obraz

Obrázek 3.9: Ukázka předzpracování – světelný defekt

Protože zejména u rostlin v substrátu jsou kořeny často skryty, pro zrychlení následného výpočtu pro každý pixel Root Tracker vypočítá hodnotu pixelu jako průměr 48 sousedních pixelů pomocí konvoluce, čímž výsledně spojí nejmenší mezery.

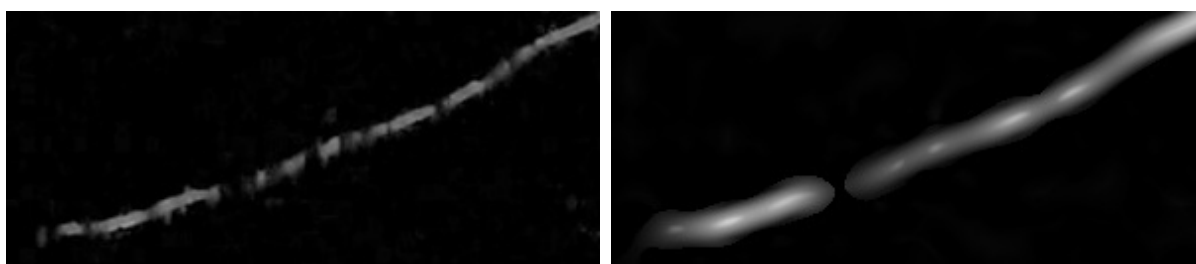


(a) Původní obraz

(b) Předzpracovaný obraz

Obrázek 3.10: Ukázka předzpracování – kořen

V případě nedostatečného kontrastu mezi kořenem a pozadím je možné ještě aplikovat Meijeringův filtr, který zvýrazní veškeré stejnoměrně široké a zároveň dlouhé struktury. [4]



(a) Původní obraz

(b) Po aplikaci filtru

Obrázek 3.11: Ukázka předzpracování – Meijeringův filtr

Výsledně Root Tracker vypočítá pozadí obrazu díky aplikování mediánu pomocí konvoluce s velkým konvolučním okénkem (pro standardní rozlišení například o velikosti 101x101 - 255x255). Tímto odstraní veškeré malé či jemné struktury obrazu a získá tak pouze gradient pozadí.



(a) Původní obraz

(b) Pozadí obrazu

Obrázek 3.12: Ukázka extrakce pozadí pro in vitro

Finálně od původního obrazu odečte detekované pozadí a odstraní již dříve extrahovanou zelenou plochu (viz Detekce počátků). Díky tomuto postupu zůstane v obraze jen systém kořenů.



Obrázek 3.13: Ukázka obrazu po aplikaci finálních úprav

3.4 Detekce kořenů

Root Tracker obsahuje dva algoritmy pro extrakci kořenové hierarchie:

3.4.1 Detekce pomocí prahování

Kdybychom použili samostatnou detekci pomocí prahování, zjistili bychom, že by tato metoda nebyla efektivní z důvodu častého výskytu nečistot v obraze. Proto jsem se inspiroval hranovým detektorem Canny a implementoval dvojité prahování – nejprve se provede prahování s vysokým prahem, které detekuje jen jasně určené kořeny a následně se provede prahování se sníženým prahem, ve kterém se mohou nacházet i jiné objekty než kořeny. V binarizovaném obraze Root Tracker nejprve detekuje veškeré kontury a následně pro každou konturu prověří, zda se nachází alespoň část v obraze, který byl binarizován za pomoci vysokého prahu. Pokud ano, je daná kontura zachována, pokud ne, je zahozena.



(a) Binarizovaný obraz s nízkým prahem

(b) Finální binarizovaný obraz

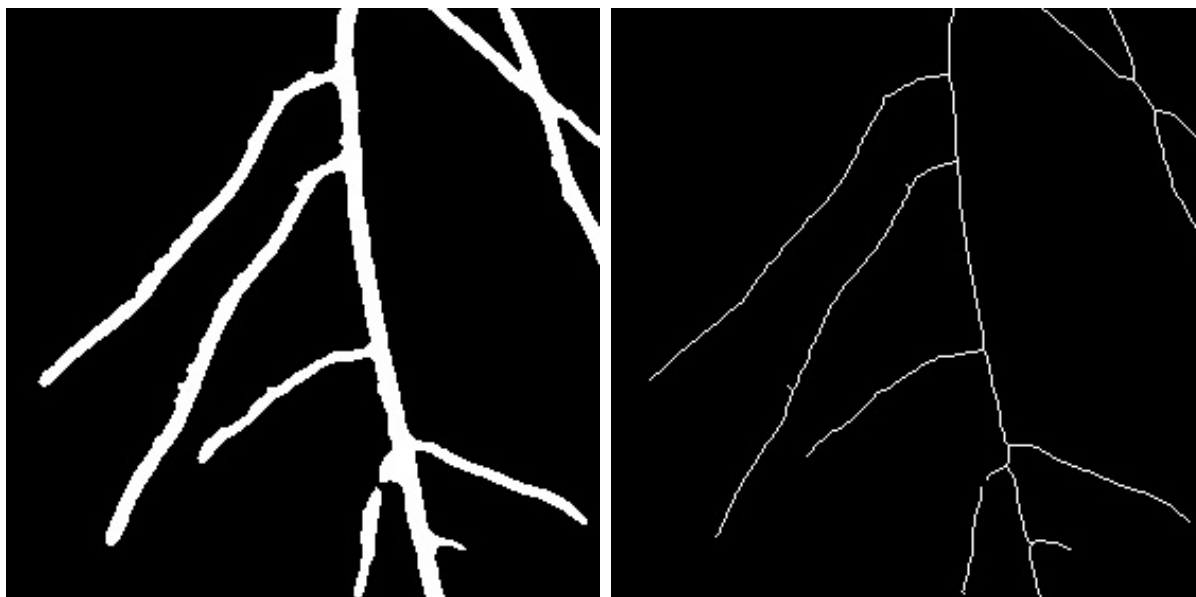
Obrázek 3.14: Ukázka aplikace filtru inspirovaného od Canny detektoru

Pokud i přes tento filtr zůstanou nečistoty s malou plochou a malou délkou v obraze, jsou automaticky zahozeny.

Po úspěšném prahování Root Tracker vytvoří skeleton z binarizovaného obrazu, aby mohl následná data jednoduše a rychle zpracovat.

Aby bylo možné rychle spočítat a najít konce kořenů, napadlo mě vytvořit si vlastní transformaci pomocí konvoluce následovně:

kde f je vstupní skeleton, g je obraz, kde hodnota každého pixelu odpovídá počtu sousedních bílých pixelů v jeho okolí 3×3 a Z je konstanta větší než > 8 . Maximum je použito z důvodu odstranění negativních hodnot po odečtení Z , které je implementováno z důvodu filtrace



(a) Binarizovaný obraz

(b) Skeleton binarizovaného obrazu

Obrázek 3.15: Ukázka skeletonizace

$$g(x, y) = \max(0, K * f(x, y) - Z),$$

Rovnice 3.2: Transformace pro výpočet počtu susedů

všech mimoskeletonových hodnot (pokud by Z nebylo použito jako středová hodnota konvolučního okénka a následně nebylo odečteno, konvoluce by způsobila jen rozmazání původního skeletonu). Konvoluční okénko K vypadá následovně:

$$K = \begin{bmatrix} 1 & 1 & 1 \\ 1 & Z & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

Rovnice 3.3: Konvoluční okénko K

Po této operaci Root Tracker rozpojí veškeré spojení v skeletonu (tedy umístí vyplněný kruh o hodnotě 0 do všech bodů, kde $g \geq 3$) z důvodu skeletonizovací funkce, která spojuje většinu křížení kořenů do jednoho svazku již několik desítek pixelů dopředu a bylo by tak výpočetně nepraktické hledat, který kořen pokračuje kterým směrem.

Místo toho Root Tracker vyhledá veškeré kontury v rozpojeném skeletonu a pro všechny konce, kde $g \neq 1$ (tedy takové konce, které byly vytvořeny až po odstranění křížení) si vypočítá průměrnou diferenci převedenou na úhel (směr konce) z několika posledních pixelů obou konců kontury a poté vypočítá průměrnou diferenci z průměrného úhlu opět z několika posledních pixelů pro zachycení zakřivování.

Po výpočtu těchto vlastností pro všechny rozpojené konce Root Tracker následně

kombinatoricky spočítá chybovou funkci pro každé dva konce, které je možné spojit, a vybere spojení s nejmenší chybou:

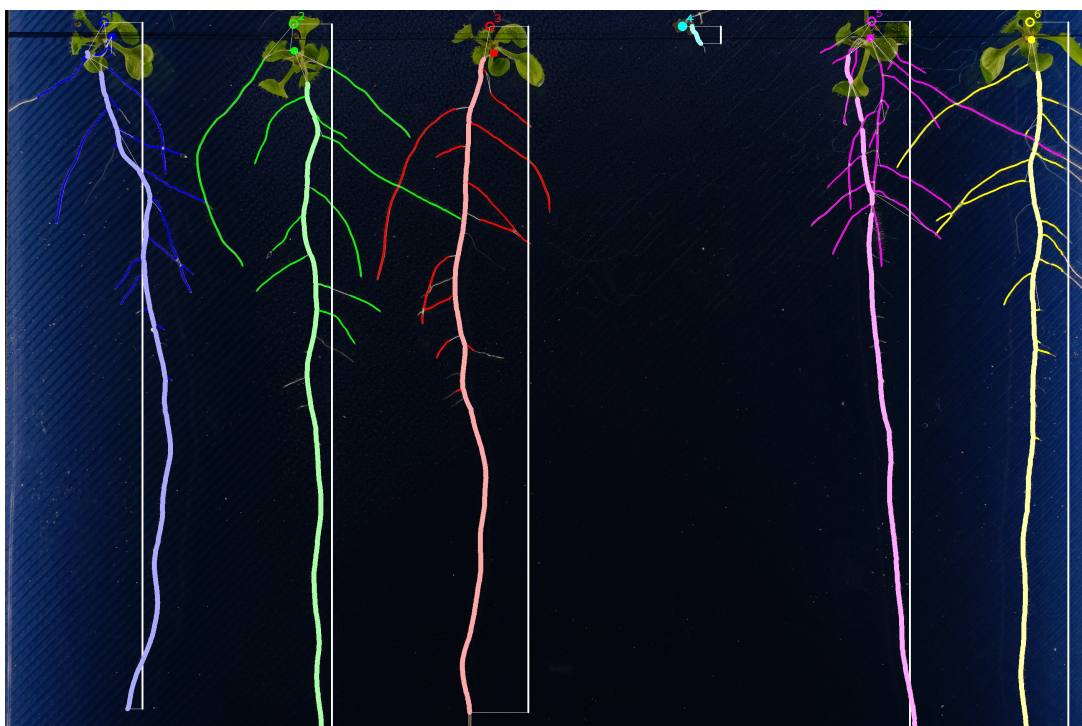
$$E(t_i, b_j) = \|t_i - b_j\|_2 + |a_{t_i} - a_{b_j}| + |d_{t_i} - d_{b_j}|,$$

Rovnice 3.4: Chybová funkce $E(t_i, b_j)$

kde a jsou první diference převedené na úhel, d jsou druhé diference z úhlu¹, $t_i[x, y]$ a $b_j[x, y]$ jsou dva různé konce ($i \neq j$) rozpojených kořenů, přičemž $t_i[x, y]$ je horní konec, který je z dvou konců stejného kořene blíže k počátku rostliny a $b_j[x, y]$ je dolní kořen nacházející z dvou konců stejného kořenu dále od počátku (neboli níže), v úvahu je proto možné brát jen taková spojení konců kořenů, kde b_j je výše než t_j .

Při kombinatorickém optimalizování spojování konců kořenů Root Tracker považuje jako jeden z konců i všechny počátky (viz kapitola Detekce počátků), díky tomuto postupu automaticky přiřadí i všechny počátky k nejpravděpodobnějším kořenovým systémům.

Ve finálním výstupním obrazu Root Tracker enumerativně označí kořenové struktury zleva doprava a každé přiřadí unikátní barvu. Dále tlustěji a světlejší barvou označí hlavní kořen a vedle něj nakreslí úsečku, jež signalizuje maximální dosaženou hloubku celého kořenového systému.



Obrázek 3.16: Ukázka finálního výstupu pro in vitro modul

¹Dané veličiny jsou nejprve otočené o 180°, aby byly porovnatelné, jelikož dva dané konce vždy musí mířit proti sobě, nikoli od sebe

3.4.2 Detekce pomocí SNR

V případě, že není možné využít metodu prahování (například pokud jsou v záběru silné odlesky), vyvinul jsem pro Root Tracker unikátní, jednoduchou a účinnou metodu, která detekuje kořenovou hierarchii rovnou při její binarizaci.

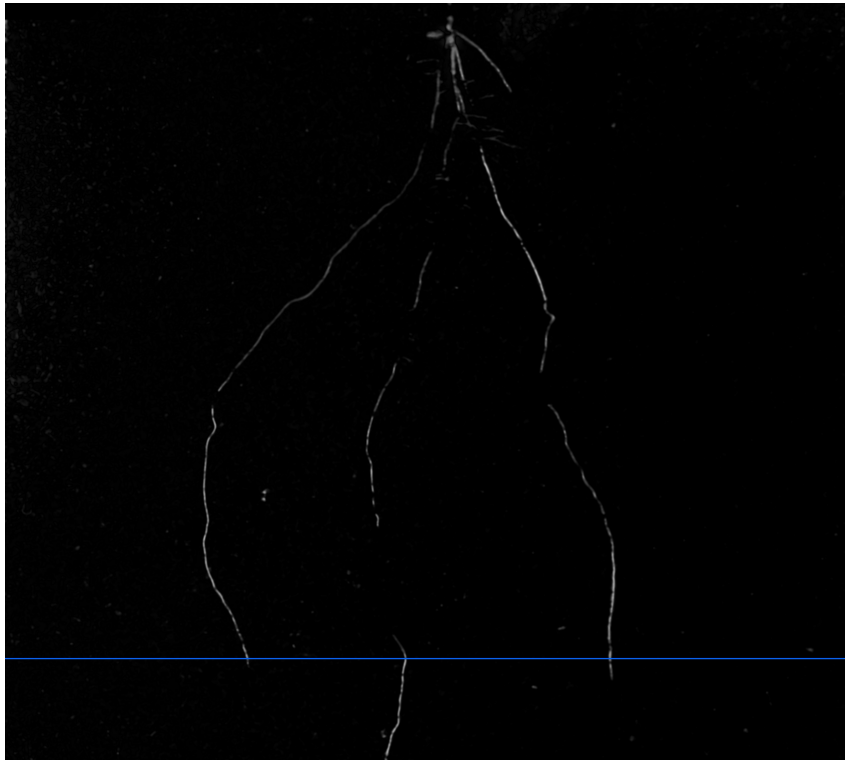
Root Tracker prochází každý řádek systémem od posledního po první a hledá vrcholy (při implementaci pomocí diference je vrchol takové místo, kde se mění diference z kladné na zápornou), které mají hodnotu vyšší než je 80% kvantil (vyloučíme tak většinu pozadí, avšak zachováme kořeny, které zdaleka nezabírají dohromady více než 20 % celého řádku a jsou jasnější než pozadí).

Root Tracker opět implementuje prahování chytrým způsobem jako Canny detektor – aby mohl být daný vrchol klasifikován jako kořen a dále sledován, je nejprve nutné, aby přesáhl vysoký práh. Jakmile ho přesáhne, je klasifikován jako kořen a pokud je při průchodu dalšího řádku nalezen vrchol na místě, které je blízké předchozímu místu, stačí již aby jen přesáhl úroveň nízkého prahu a nacházel se v podobném směru jako předchozí vrcholy. Následně se spočítá signal-to-noise ratio pro daný vrchol následujícím způsobem:

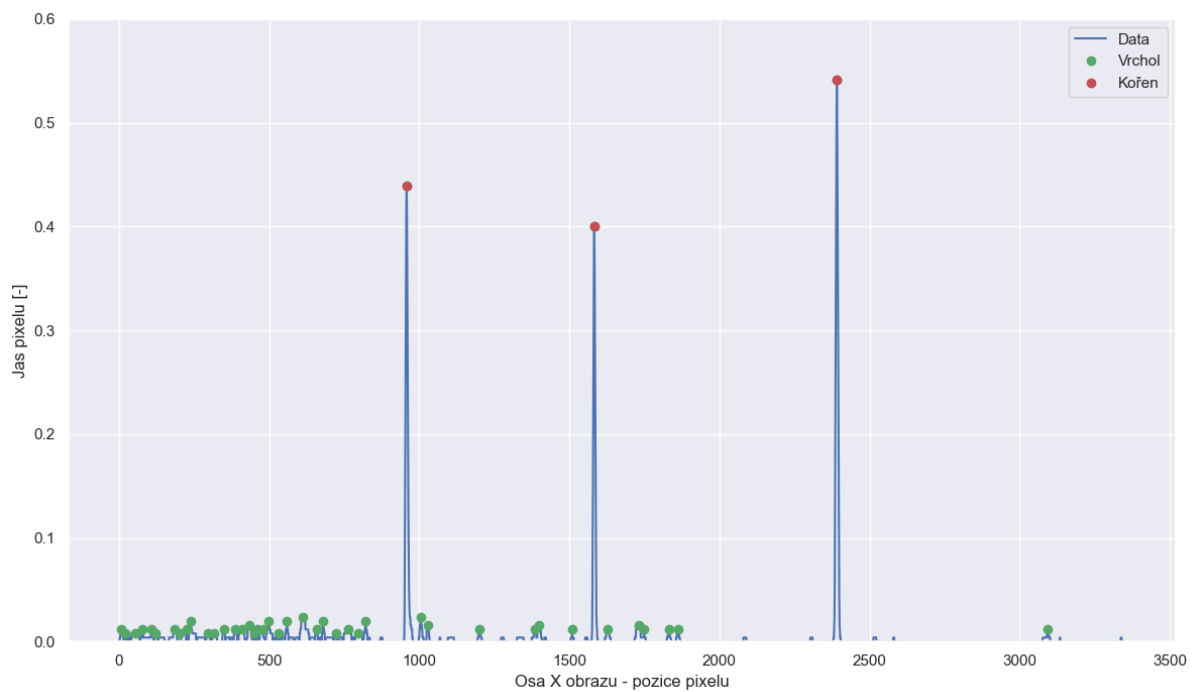
$$SNR(x) = \frac{x}{NOISE},$$

Rovnice 3.5: Výpočet *SNR*

kde *NOISE* je definováno opět jako 80% kvantil nejbližší okolí (desetina šířky obrazu) a *x* je jas daného vrcholu. Pokud tato hodnota překročí práh a splní předchozí podmínky, je pixel přiřazen k předchozímu vrcholu, od kterého se porovnával směr a vzdálenost. Tímto způsobem je vytvořena celá kořenová hierarchie.

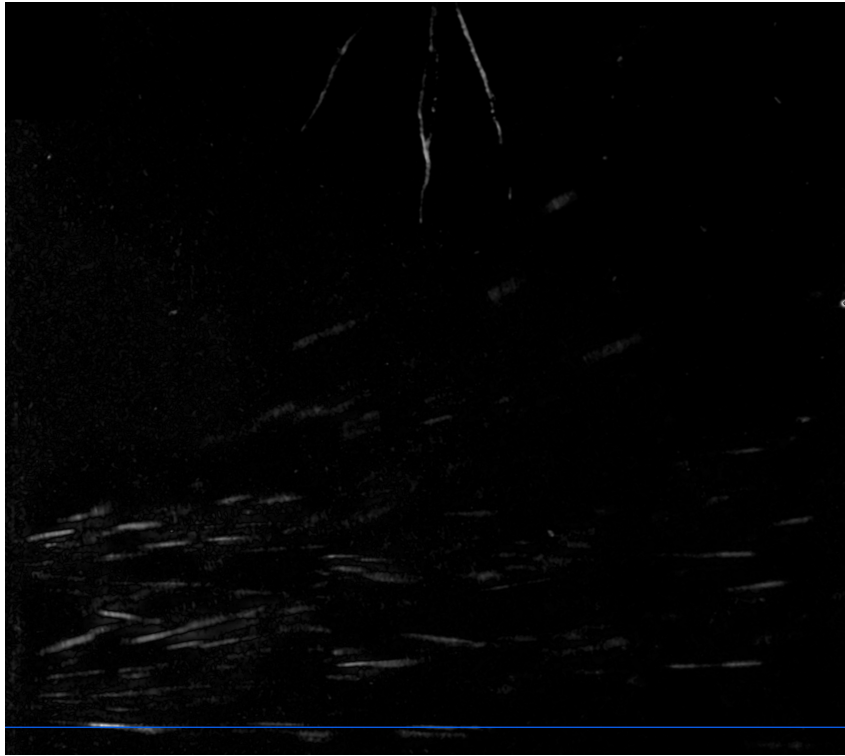


(a) Původní obraz

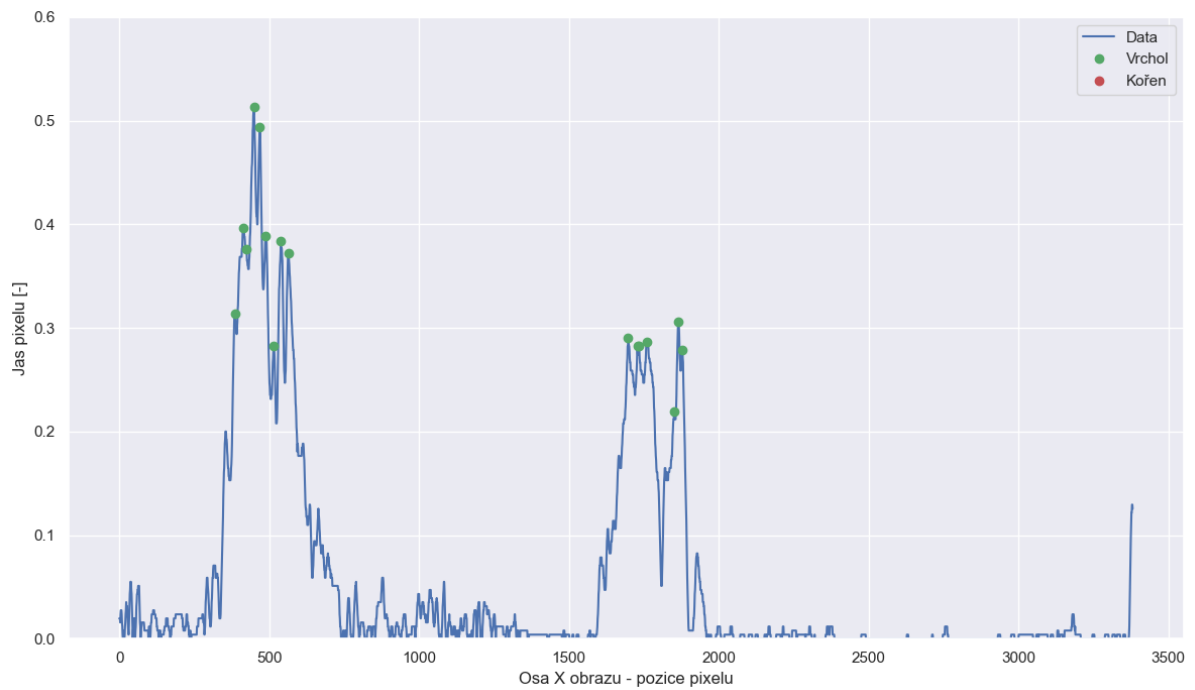


(b) Řez obrazem a detekce vrcholů

Obrázek 3.17: Ukázka jednoho z řádků obrazu, kde jsou přítomny kořeny



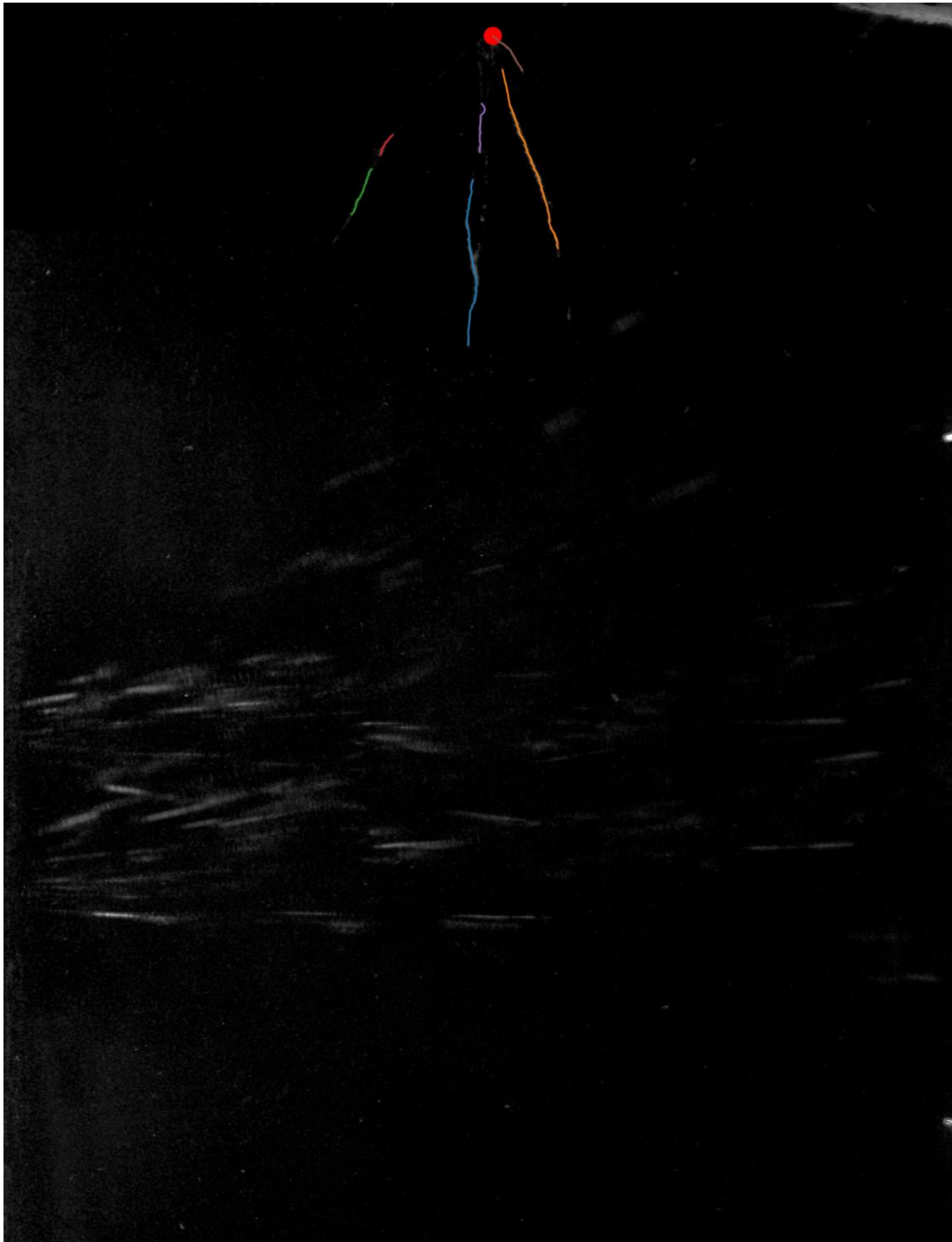
(a) Původní obraz



(b) Ukázka jednoho z řádků obrazu, kde jsou přítomny odlesky

Obrázek 3.18: Ukázka řezu v místě silného odlesku

Díky tomuto postupu je tato metoda velmi robustní a vůbec nedetekuje dlouhé horizontální odlesky vyskytující se často v obraze, i když odlesky dosahují často vyšších jasových hodnot než samotné kořeny.

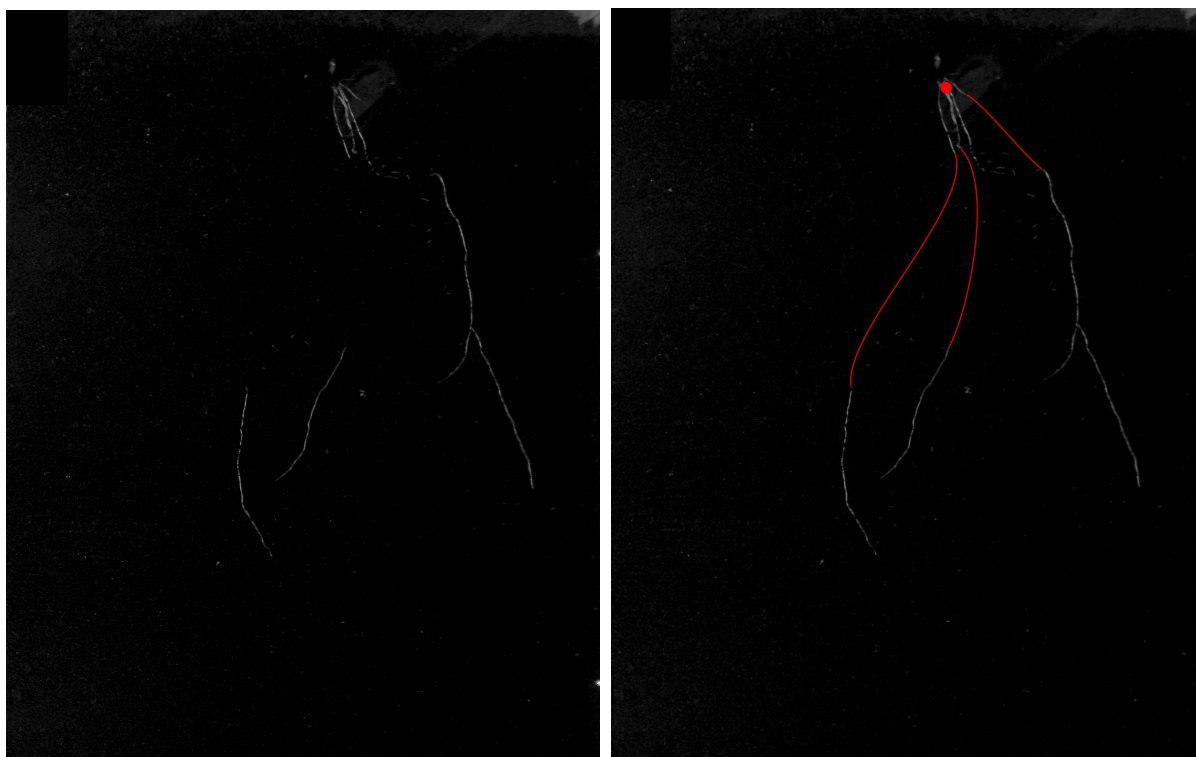


Obrázek 3.19: Ukázka funkčnosti se silnými světelnými odlesky

3.5 Predikce chybějících kořenů

V případě, že je tento modul aktivován, Root Tracker provede optimalizaci všech možných propojení mezi volnými konci kořenů spojených s kořenovým systémem a kořeny, které nebyly spojeny s dosud žádným kořenovým systémem v předchozím kroku. Poté vybere takové řešení, které má minimální hodnotu chybové funkce, která penalizuje počet volných konců, vzdálenost a směr konců kořenů.

Finálně propojí konce kořenů řešení s nejmenší chybovou hodnotou za pomoci [Bezierovy křivky](#). Jako počáteční a konečný bod křivky zvolí propojované konce kořenů, které jsou od sebe vzdálené se vzdáleností d_0 . Prostřední řídící body určí Root Tracker polárně jako body, které se nachází od konců kořenů ve vzdálenosti $d = \frac{1}{4}d_0$ a směru vypočítaném z průměrné derivace několika posledních bodů jednotlivých konců kořenů.



(a) Bez predikce chybějících kořenů

(b) S predikcí chybějících kořenů

Obrázek 3.20: Ukázka predikce chybějících kořenů

3.6 Další funkce

Root Tracker nabízí spoustu dalších funkcionalit, z významných například:

3.6.1 Časová informace

Jelikož Root Tracker udržuje automaticky stejnou souřadnicovou soustavu (tzn. každý pixel obrazu je v časové sérii možné nalézt vždy na stejných souřadnicích), tak v případě zpracování časové série automaticky pro každý obraz v čase t používá informaci i z obrazu v čase $t - 1$.

Toto chování je užitečné zejména při klasifikaci a extrakci informací ze složitějších a často se křížících kořenových systémů. Root Tracker tak automaticky klasifikuje kořeny na stejných pozicích v čase t stejně jako pro minulý čas $t - 1$. Je pravděpodobnější, že klasifikace bude přesnější v čase $t - 1$, když počty překřížení měli menší četnost, než v čase t , kdy je četnost překřížení vyšší.

3.6.2 Automatické čtení čárových kódů

Root Tracker dokáže automaticky přečíst čárový kód identifikující daný experiment. Pro čtení jsem využil již hotovou knihovnu [zbar](#), v Pythonu konkrétně [pyzbar](#), která obaluje tuto funkcionalitu.

3.6.3 Paralelní výpočet

Veškerý výpočet je implementován paralelně pomocí knihovny [multiprocessing](#) a jeho nadstavby s ukazatelem průběhu [tqdm](#). Root Tracker tedy využije maximálně celý výpočetní výkon, který je mu k dispozici, aby doručil výsledky co nejrychleji.

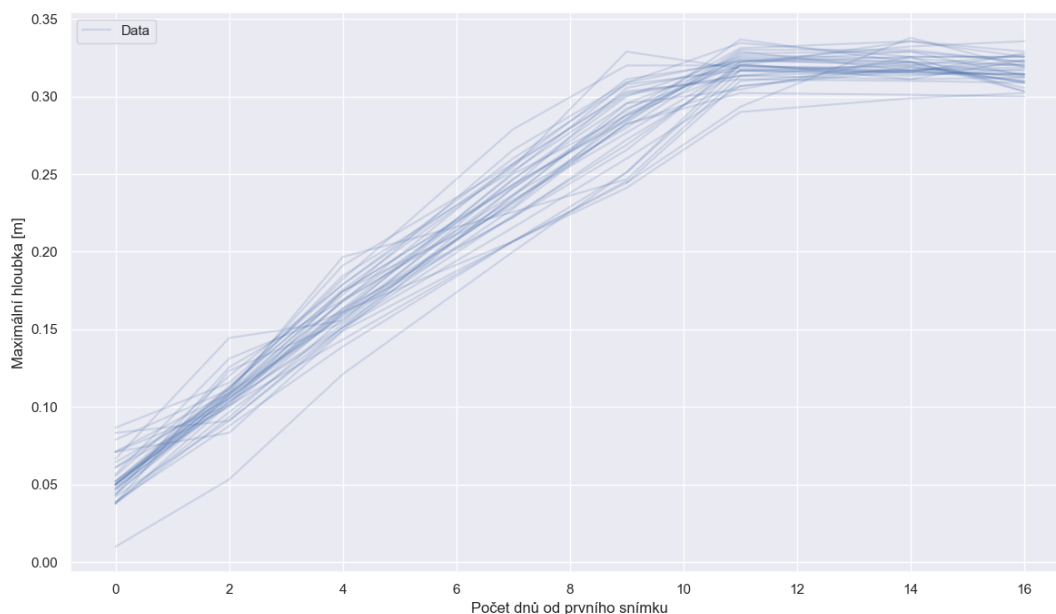
Kapitola 4

Vyhodnocení

4.1 Extrakce vlastností

Root Tracker má informace pro všechny kořeny individuálně, dokáže tedy vypočítat jakoukoli požadovanou metriku. Z často používaných se jedná například o délku a hloubku (vzdálenost nejnižšího bodu kořenové soustavy od počátku) hlavního kořene, počet bočních kořenů, nebo plochu kořenové soustavy.

Všechny vypočítané informace Root Tracker ukládá do výstupního CSV souboru, ze kterého následně dokáže vytvořit vizualizace vhodné pro porovnávání a vyhodnocení.



Obrázek 4.1: Vizualizace surových dat z CSV¹

¹Data byla za účelem dočasného utajení upravena

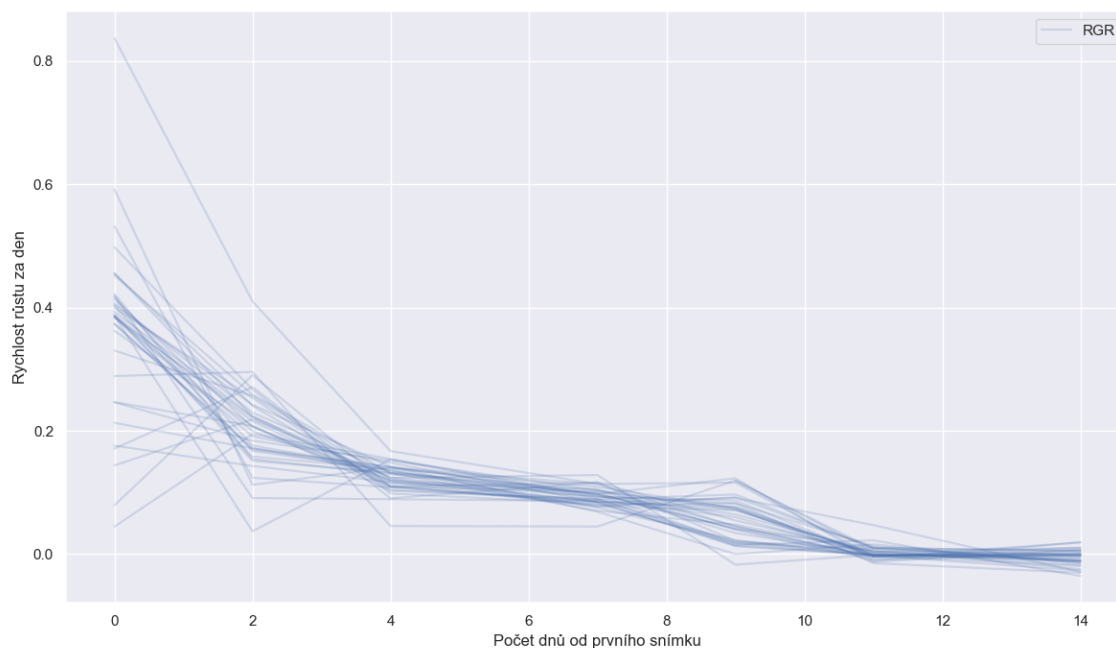
4.2 RGR

Jakmile má Root Tracker vyhodnocené zvolené metriky, dokáže pro jakoukoli veličinu (délka, hloubka, ...) vypočítat standardně využívanou biologickou metriku pro vyhodnocování – rychlost růstu:

$$RGR = \frac{\ln S_2 - \ln S_1}{t_2 - t_1},$$

Rovnice 4.1: Výpočet *RGR*

kde S_1 a S_2 je sledovaná veličina (například velikost, délka, plocha) zvolené části rostliny (například list, kořen) naměřená ve dvou časových bodech t_1 a t_2 .



Obrázek 4.2: Ukázka závislosti vypočtené RGR pro hloubku v čase²

²Data byla za účelem dočasného utajení upravena

4.3 Logistická křivka

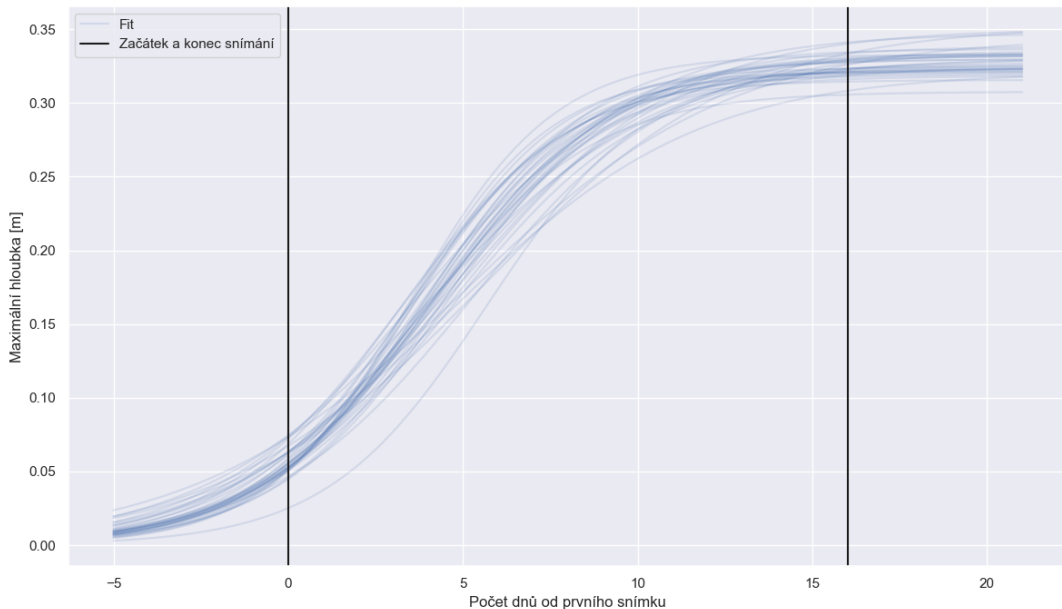
Jelikož kořenový systém rostlin při dlouhodobém sledování neroste exponenciálně ale logisticky, Root Tracker také přichází s inovativní metrikou pro vyhodnocování. Nejprve projde extrahovaná data a následně je proloží logistickou křivkou za pomoci metody nejmenších čtverců:

$$f(x) = \frac{A}{1 + e^{-B(x+C)}}$$

Rovnice 4.2: Rovnice logistické funkce

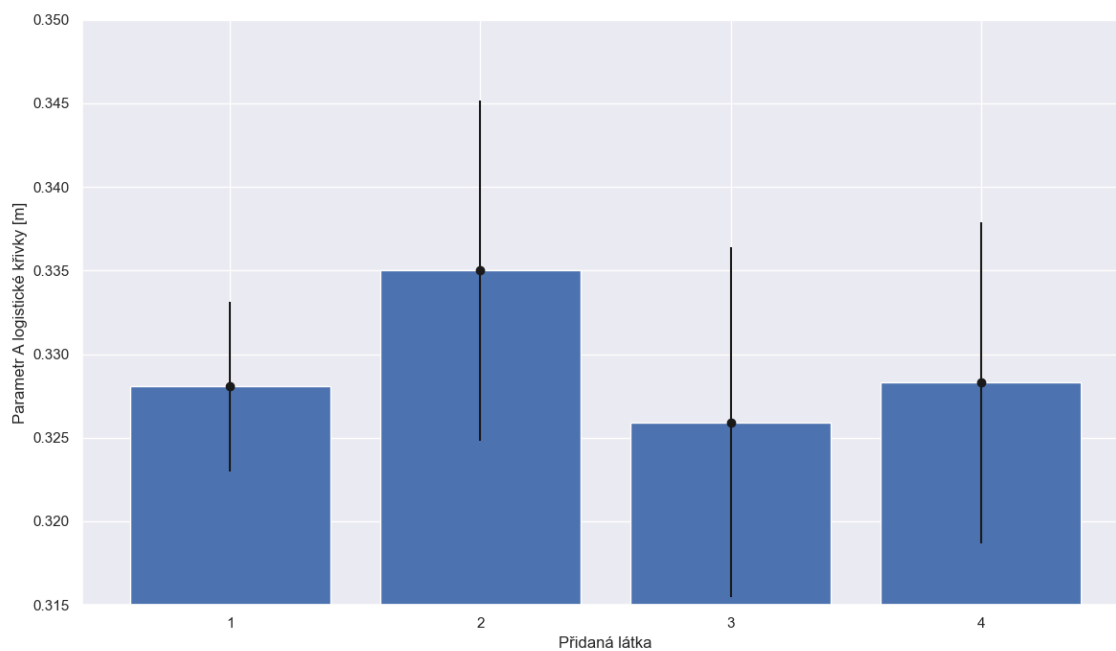
kde:

- parametr A reprezentuje maximální hodnotu sledované veličiny, kterou daná veličina dosáhla, nebo dosáhne jakmile přestane růst (v případě, že bylo snímání ukončeno dříve).
- parametr B symbolizuje rychlost růstu po celou dobu, nikoli jen v jednotlivých časových bodech, jako je to u RGR
- parametr C symbolizuje časový posun (zda daný kořenový systém rostl v čase dříve/později)

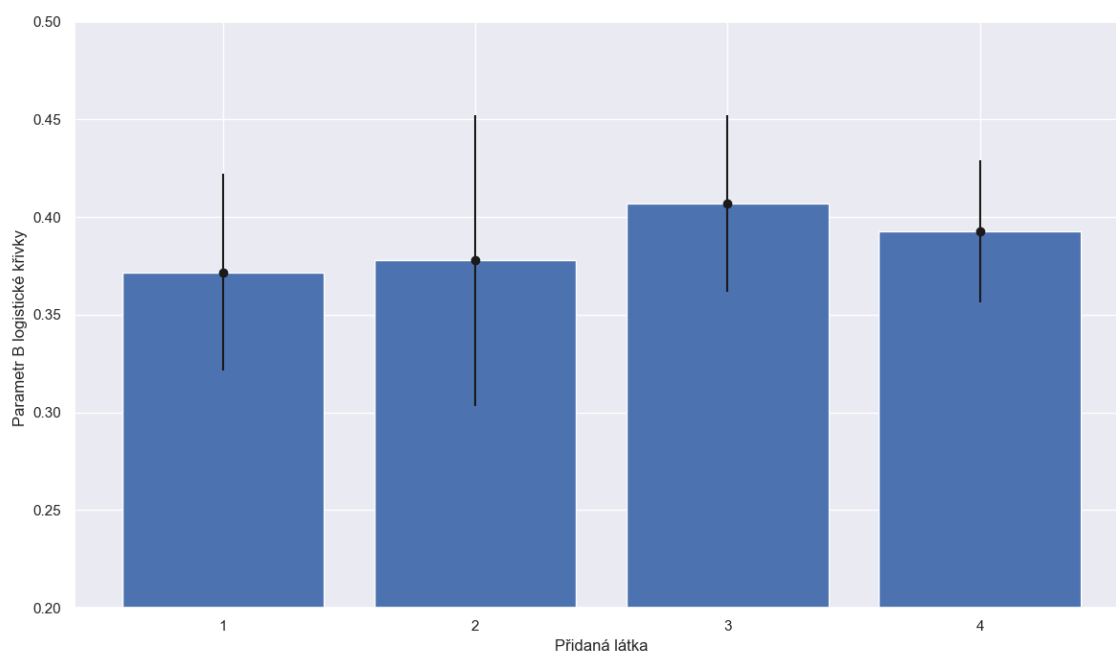


Obrázek 4.3: Ukázka proložení dat logistickou křivkou³

³Data byla za účelem dočasného utajení upravena



Obrázek 4.4: Ukázka vyhodnocení dle parametru A⁴



Obrázek 4.5: Ukázka vyhodnocení dle parametru B⁵

⁴Data byla za účelem dočasného utajení upravena

⁵Data byla za účelem dočasného utajení upravena

Závěr

Ve své práci jsem vytvořil metodiku pro zpracování, extrakci parametrů (jako například tloušťka, délka, počet vedlejších kořenů, atp.) a vyhodnocení dat pro snímky kořenových systémů i pro komplexních případy, kdy se složité kořenové systémy často kříží, nebo kdy jsou části kořenů skryté za substrátem a je nutné je pro správné vyhodnocení predikovat.

Tuto metodiku jsem poté modulárně implementoval v programovacím jazyce [Python](#) za pomoci knihoven [NumPy](#), [OpenCV](#) a [Scikit-Learn](#). Nastavení použitých algoritmů jsem umožnil uživateli upravovat pomocí konfiguračních souborů. Při vizualizacích jsem využil klasického způsobu vyhodnocení pomocí RGR, ale také i inovativního způsobu, který interpretuje parametry logistické křivky.

Využití a aplikaci Root Trackeru a jeho algoritmů jsem demonstroval na snímcích rostlin pěstovaných in vitro a poté na rostlinách zasazených v rhizotronech.

V dalších měsících čeká Root Tracker rozšíření o uživatelské rozhraní pro maximální zjednodušení konfigurace za pomoci moderních webových technologií.

Oficiální zdrojový kód bude zveřejněný na webové adrese <https://github.com/PalackyUniversity/root-tracker> během Q3/2023 při publikaci článku.

Seznam zdrojů

1. PYTHON. *Python* [online]. [cit. 2022-03-06]. Dostupné z: <https://python.cz/>.
2. PRATAP, Aditya; GUPTA, Sanjeev; NAIR, Ramakrishnan Madhavan; GUPTA, S. K.; SCHAFLEITNER, Roland; BASU, P. S.; SINGH, Chandra Mohan; PRAJAPATI, Umashanker; GUPTA, Ajeet Kumar; NAYYAR, Harsh; MISHRA, Awdhesh Kumar; BAEK, Kwang-Hyun. Using Plant Phenomics to Exploit the Gains of Genomics. *Agronomy*. 2019, roč. 9, č. 3. ISSN 2073-4395. Dostupné z DOI: [10 . 3390 / agronomy9030126](https://doi.org/10.3390/agronomy9030126).
3. R. FISHER S. Perkins, A. Walker; WOLFART., E. *Morphology - skeletonization* [online]. [cit. 2023-04-10]. Dostupné z: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm>.
4. MEIJERING, E.; JACOB, M.; SARRIA, J.-C.F.; STEINER, P.; HIRLING, H.; UNSER, M. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A*. 2004, roč. 58A, č. 2, s. 167–176. Dostupné z DOI: <https://doi.org/10.1002/cyto.a.20022>.

Seznam zkratek

Zkratka	Význam
DLSR	Digital Single-Lens Reflex
SW	Software
ROI	Region of Interest
LED	Light-Emitting Diode
SNR	Signal-to-Noise Ratio
RGR	Relative growth rate
PETG	Polyethylentereftalát

Seznam obrázků

1.1	Ukázka in vitro modulu	10
1.2	Ukázka rhizotronu	11
1.3	Příklad binarizace	12
1.4	Příklad skeletonizace	13
2.1	Ukázka sestavy pro fotografování in vitro	15
2.2	Ukázka sestavy pro fotografování kořenů in vitro	16
2.3	Ukázka sestavy pro fotografování in vitro bez krytu	17
2.4	Ukázka sestavy pro fotografování in vitro s krytem	17
3.1	Ukázkový originální vstupní obraz in vitro	18
3.2	Ukázka manuální ROI s rotací 180° a ořezem 0 %	19
3.3	Ukázka automatického ořezání pomocí prahování pro in vitro	20
3.4	Průměrný jas sloupců rhizotronu a určený práh	21
3.5	Ukázka funkce algoritmu pro rhizotron	22
3.6	Ukázka detekce "nadzemní"části rostlin a výpočtu počátku	23
3.7	Ukázka detekce počátku pro rhizotron	24
3.8	Ukázka předzpracování – filtrování nečistot z obrazu	25
3.9	Ukázka předzpracování – světelný defekt	25
3.10	Ukázka předzpracování – kořen	26
3.11	Ukázka předzpracování – Meijeringův filtr	26
3.12	Ukázka extrakce pozadí pro in vitro	27
3.13	Ukázka obrazu po aplikaci finálních úprav	27

3.14	Ukázka aplikace filtru inspirovaného od Canny detektoru	28
3.15	Ukázka skeletonizace	29
3.16	Ukázka finálního výstupu pro in vitro modul	30
3.17	Ukázka jednoho z řádků obrazu, kde jsou přítomny kořeny	32
3.18	Ukázka řezu v místě silného odlesku	33
3.19	Ukázka funkčnosti se silnými světelnými odlesky	34
3.20	Ukázka predikce chybějících kořenů	35
4.1	Vizualizace surových dat z CSV	37
4.2	Vizualizace závislosti vypočtené RGR pro hloubku v čase	38
4.3	Ukázka proložení dat logistickou křivkou	39
4.4	Ukázka vyhodnocení dle parametru A	40
4.5	Ukázka vyhodnocení dle parametru B	40

Seznam rovnic

3.1	Průměrná hodnota dle sloupce obrázku	21
3.2	Transformace pro výpočet počtu sousedů	29
3.3	Konvoluční okénko K	29
3.4	Chybová funkce $E(t_i, b_j)$	30
3.5	Výpočet SNR	31
4.1	Výpočet RGR	38
4.2	Rovnice logistické funkce	39

Seznam příloh

1. Výkres 3D modelu držáku pro in vitro moduly ve formátu PDF
2. Výkres 3D modelu krytu pro in vitro moduly ve formátu PDF
3. Výkres 3D modelu hřídele pro in vitro moduly ve formátu PDF
4. Výkres sestavy in vitro modulu
5. CAD model držáku pro in vitro moduly ve zdrojovém formátu SLDPRT
6. CAD model krytu pro in vitro moduly ve zdrojovém formátu SLDPRT
7. CAD model hřídele pro in vitro moduly ve zdrojovém formátu SLDPRT
8. CAD soubor sestavy ve zdrojovém formátu SLDASM
9. 3D model držáku pro in vitro moduly ve formátu STEP
10. 3D model krytu pro in vitro moduly ve formátu STEP
11. 3D model držáku ve formátu 3MF připravený pro multibarevný tisk
12. 3D model krytu ve formátu 3MF připravený pro multibarevný tisk
13. Seznam použitých knihoven pro Python a jejich verze